ACCELERATED COMPUTING FOR MOLECULAR DYNAMICS SIMULATION

by
FNU Samarjeet

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
July 2019

ABSTRACT

Molecular dynamics (MD) simulation serves as a computational microscope into the behavior of the biological and chemical macromolecules. At its core, MD models the interactions between atoms at various levels – force fields model the higher quantum level interactions using simpler physics-based models of interaction energies, while periodic boundary conditions model the bulk phase using lattice-based periodic copies of the simulation box. One limitation of the finite size of the simulation box seen during the simulation of membrane bilayers is the artifact of a chemical disequilibrium between the two layers as a drug molecule enters into the bilayer. We have tried to solve this problem by using a periodic boundary condition which has a half screw symmetry. Our results show that the method scales similar to the best-known method for the normal periodic boundary conditions.

We have migrated CHARMM to an efficient implementation on the GPUs. These architectures provide thousands of cores on the same chip but require different programming model in order to use the underlying architecture. Our results show that the new CHARMM CUDA engine is efficient in time and accurate in precision.

We have also participated in blind prediction challenges organized by SAMPL community to have a fair assessment of the computational chemistry tools. We developed a hybrid QM and MM technique to predict the pKa of drug-like molecules. It avoids the implicit solvent model used by quantum mechanical models and uses explicit solvent molecules. Since modeling explicit solvent molecules is difficult at QM level, they are modeled at the MM level instead. Thermodynamic

cycle couples the aqueous Gibbs free energy of deprotonation to simpler components which can be modeled with higher accuracy.

We also built a deep learning model to predict the logP of a set of drug-like molecules in a blind fashion. The generated model is robust over a large number of molecules, not just the ones that it was tested for in the SAMPL competition. We expect the method to be interesting for the drug design industry since lipophilicity of a molecule is important to be known even before it has been synthesized.

Thesis Advisers: Thomas W. Woolf, Phd (JHMI)

Bernard R. Brooks (NIH)

Thesis Reader: Albert Lau, Phd (JHMI)

To Maa and Papa.

ACKNOWLEDGEMENTS

I would like to thank my parents for ensuring that I grow up in an environment of learning and pursuance of excellence. They serve as my guiding light at all moments of despair. It's hard to be thousands of miles away on the other side of planet and during all this time they have supported in the best extent possible.

I would like to thank my mentor at JHMI, Thomas Woolf, who has always had my best interests in mind. I am blessed to have such unconditional support from a person for my progress. I will always be indebted to him for ensuring that I am able to pursue my dreams!

My mentor at NIH, Bernie Brooks, has been such a positive influence in my life. He has given me so much independence to explore my field of research and at the same time always being available to help me out when I get stuck. Rich Pastor has always had great insights into the properties of the lipids and I have learnt a lot from him over the years. Richard Venable has an eye on the details of the simulations and performing simulations has been so much easier due to his help.

I would like to thank my siblings Amarjeet, Keshavjeet and Kshitiz Anand for being great role models. You guys provided the most affectionate ambience at home and I have enjoyed growing up with you.

My lab members and friends have had a tremendous impact in my work both inside and outside the group. Andrew Simmonett has always given me the best advice and helped me figure out the

intricacies of reciprocal spaces. I have had great interactions with Frank Pickard, Gerhard Konig, Florentina Tofoleanu, Juyong Lee, Mohsen Pourmousa, Kyungreem Han, Michael Jones, Andreas Kramer and others in topics not just in science but life in general! Milind Gunjan, Pushkar Vartak and Ankit Munoth have been great sport outside lab. I got to learn so much from them.

And then there's my lovely girlfriend, Asma, who has stood by me for so long time. If it wasn't for her, I would have never been able to pen-down this thesis. I am amazed every single day that she chooses to be around me!

My thesis committee members, Albert Lau, Carolyn Machamer, and Mario Amzel, have been very supportive over these years. My interactions with them have helped me shape my thesis towards the form that it is in now.

I would like to thank the BCMB community and administrators for their support. I would especially like to thank, Carolyn Machamer, who ensured that I pass through the difficult phases of my graduate work. She stood by me and without her help I would have never been able to find a lab of my interest and shape my career in the track that I wanted to take when I joined the BCMB program. I would also like to thank, Arrhonda Gogos, who helped me in making sure that all the nitty-gritties are well taken care of.

Table of Contents

Table of Tables

Table of Figures

This page has intentionally been left blank

Chapter 1

Introduction

Molecular dynamics (MD) simulations play an important role in understanding the structure, function and interactions of systems at an atomic level of description. Since the first MD simulation carried out in 1977 by McCammon et. al., these simulations have evolved in complexity in terms of size of systems studied and time lengths of the simulations. The basic idea behind MD simulation is pretty simple: a particle-based model of the system is first generated in terms of the nature of interaction between the particles in the system. The system is then evolved in time based on certain propagation rules. Under the ergodic assumption that states sampled over a long period of time are similar to the entire set of accessible states, simulations can be used to study the thermodynamic properties of the system.

If a quantum mechanical level of description of the particles is chosen, electrons are explicitly accounted for in the model. Interaction energies are then calculated by solving the Schrodinger's equation under certain assumptions. However, it soon becomes intractable for a system beyond a hundred atoms. Instead, under the Born-Oppenheimer approximation that the motion of nuclei of an atom can be separated from the motion of the electrons, MD simulations typically use an atom level description of the system. A force-field is first designed which approximates the QM level interactions in terms of bonded and non-bonded interactions of a pair of atoms (higher level potentials using many-body interactions have also been studied, though their applications remain limited so far). With this level of description, thousands to hundreds of thousands of atoms can be studied.

Speed of simulation is critical when studying such large systems. In order to remove the artifacts emanating from the finite size of the simulation box, interactions from infinite images of the box

are considered. Non-bonded interactions are inherently $O(n^2)$ and this makes the energy and gradient calculations at each time step quite time-consuming. To address these challenges many hardware and algorithmic advancements have been made in the last few decades to increase the speed of the simulations.

DE Shaw's Anton supercomputers have been a major advancement in the field of molecular dynamics. The Application-specific Integrated Chip (ASIC) along with duplex network connected in a torus grid allow the very efficient scaling of the parallelization over all the nodes of the machine. This hardware architecture at its core uses the algorithmic principle of spatial decomposition of simulation box. These methods have been termed as Neutral Territory (NT) methods as the interaction between a pair of atoms is often calculated on nodes where neither of the atoms reside! This scheme, unintuitively, minimizes the import volume for each node and hence the overall simulation scales very well. However, these methods were developed for the normal periodic boundary conditions called P1. My work on Extended Eighth Shell (in chapter 2) builds upon the same principles and gives an equivalent parallelization scheme for more complicated periodic boundary conditions that involve a half-screw rotational symmetry.

While Anton provides a very specialized architecture for the parallelization of the computations in MD, manufacturing the ASIC is very expensive and is not available. The compute-intensive nature of the calculations has made Graphical Processing Units (GPUs) interesting processors to parallelize the simulation. They are affordable to individual labs and are in state of rapid improvements with each new version of the architecture. However, the programming model for these machines to extract the full performance is complicated. My work on implementation of

CHARMM on the GPUs (chapter 5) shows our redesign of the code to harness the best out of the GPUs.

Outline

In Chapter 2, I discuss the Extended Eighth Shell (EES) method that I developed for the simulation of lipid bilayers under P21 periodic boundary condition. One limitation observed during the simulation of insertion of drug molecules in membrane bilayers is the creation of chemical disequilibrium for the lipids. As the molecule enters one of the layers, the area per lipid in the two layers changes and lipids in the two layers are no longer in equilibrium with each other. EES method provides a scalable technique for carrying out the simulation such that the lipids, when they leave one of the layers, enter into the simulation box in the opposite layer. We show that the method is stable, scales efficiently over a large number of nodes and is able to reproduce data from physical experiments.

In Chapter 3, I report a novel hybrid QM and MM technique for the calculation of pKa of small drug-like molecules. This study was carried out in a blind prediction competition organized by the Drug, Design and Data Resources consortium under the SAMPL6 challenge.

In Chapter 4, the second part of the SAMPL6 challenge is discussed. This challenge involved the blind prediction of logP of a subset of the molecules in the previous pKa challenge. logP is measure of the lipophilicity of a molecule and is routinely used in drug design pipeline to assess the absorption, distribution, metabolism and excretion of the drug. For this challenge, I developed a novel deep learning approach for logP prediction.

In Chapter 5, I get back to the recent developments made in CHARMM to move it to the GPUs. The previous version of the code was designed for a heterogeneous CPU-GPU system where the

parallelization of the computation was aimed at multiple CPUs nodes with each connected to a GPU device. However, in the last few years, GPUs have had a remarkable improvement in performance while the transfer bandwidth between the CPU and GPU has not seen the same level of development. These trends are expected to continue even in the future. Hence, I have redesigned CHARMM CUDA to perform all the computations on the GPU device itself and the CPU is used only for the input/output. This redesign allows CHARMM to give throughputs in the similar range to other leading MD packages like OpenMM and Amber.

Chapter 2

An Extended Eighth-Shell Method for Periodic Boundary Conditions with Rotational Symmetry

(A version of this chapter has been submitted and is currently under review.)

Abstract

The Eighth shell method has previously been shown to be the most optimal in terms of parallelization of molecular dynamics simulation over large number of nodes. However, this method supports only the P1 periodic boundary condition (PBC) and cannot handle reflection and/or rotational symmetry. In this work we developed the Extended Eighth shell (EES) method that simulates only the asymmetric unit and communicates coordinates and images with images that correspond to P21 PBC. The P21 periodic boundary condition has application in lipid bilayer simulations as it can be used to allow the movement of lipids from one layer to the other, thus balancing the chemical potential difference between the two layers. Our results show that the EES scales similar to ES with the P21 symmetry.

1 Introduction

Molecular dynamics (MD) simulation serves as an important tool in several fields including computational chemistry, biophysics and material science. They provide a powerful model-based method to probe the microscopic and macroscopic properties of chemical systems. While the time scale of an individual step is of the order of femto-second (10^{-15}s), most physically relevant phenomenon occur at timescales of milliseconds (10^{-3}s) or higher. In order to cover the spatial and temporal scales of simulations, millions of interactions have to be calculated for billions of steps[1].

Hence, there is an acute need for fast algorithms to parallelize the MD code in order to utilize large number of processors in an efficient way.

D.E. Shaw group evaluated several parallelization strategies and introduced a set of zonal or neutral territory methods[2,34]. In these methods, interaction between a pair of particles is calculated not necessarily on the node where either of the two particles resides. This communication pattern among the nodes minimizes the inter-process communication bandwidth, which is the bottleneck for distributed memory parallel algorithms. Two of these zonal methods, Eighth Shell (ES) and Midpoint method[5], achieve the least amount of communication for regular setups of simulation, i.e. system size being not too small compared to cutoff of pairwise interaction. For general purpose architectures, communication in ES has lower latency than in Midpoint method. ES has subsequently been implemented in several MD software packages including GROMACS[6] and CHARMM[7].

MD simulations are routinely used for the study of insertion and/or rearrangement of peptides in lipid bilayers[89]. During the course of the simulation, area per lipid (APL) for lipids in the two layers changes. For example, consider the insertion of a peptide into the top layer. APL in the top layer would decrease. However, in the bottom layer it would remain the same or increase under a constant pressure or constant surface tension simulation. This leads to a state of chemical disequilibrium between the layers. In contrast to cell membranes, where lipids can move further to release the stress, lipids in simulations return back to the same layer. Flipping of lipids happens at a time scale not accessible to MD simulations and cannot be relied on for balancing the stress.

Our group earlier reported a method to allow the exchange of lipids between the two leaflets of the bilayer during the course of the simulation[10]. Rather than using the usual periodic boundary condition, called P1, this method uses the P21 periodic boundary condition. A lipid leaving the primary cell under the P21 PBC, enters the opposite layer along the orthogonal face. This method allows the equilibration of lipid chemical potential between the two layers through the course of simulation.

The symmetry operation in P21 PBC is a half screw symmetry that preserves the chirality. It involves reflections along two orthogonal axes and half a unit translation along the third axis. These two operations are equivalent to a rotation along the screw axis and half-unit translation along the same axis. However, this P21 PBC is incompatible with the basic version of ES method for scalable parallelization.

Lack of parallel scalability for P21 has deterred its wide-spread adoption[11–14]. In this work, we have designed the Extended Eighth Shell (EES) method and implemented it in CHARMM. EES minimizes the import volume for P21 PBC and scales similar to ES for P1 PBC. We present the computational details of the algorithm in Section 2. In Section 3, we present benchmark results and show its importance for lipid bilayer simulations.

2 Computational Details

Periodic boundary conditions are used in MD simulations to avoid the edge effect due to the finite size of the simulation systems. The most commonly used PBC is the P1 PBC where the

neighboring cells of the primary simulation box are simply its translated copies. This means that as a molecule leaves the primary box, its image enters back into the box along the opposite face.

The P21 symmetry operation is given by a 180° rotation around an axis and translation by half a unit length. In the present implementation, the axis chosen is chosen to be the X-axis. It facilitates the extension of the domain decomposition and the Particle Mesh Ewald (PME) calculations and minimized the import volume.

We first give an overview of the eighth shell (ES) method as implemented in the domain decomposition (domdec) package of CHARMM. We then follow this up the with the Extended Eighth Shell (EES) in the context of P21 periodic boundary condition and its implementation in CHARMM.

The Eighth Shell method is a class of spatial decomposition method called Neutral Territory (NT) method. It involves the splitting of the simulation cell of dimension (Lx,Ly,Lz) into smaller regions called boxes. A processor is associated with a single box and it is responsible for updating the positions of all the atoms in its box. Thus, if we have nx, ny, nz boxes along the X, Y and Z axes, we have nx*ny*nz total number of processors (for direct space component). Calculation of the energies and the forces however can happen in another box based on the relative positions of the atoms in the pair.

In order to avoid double counting of the pairwise interaction, each node calculates interactions in 8 zones - I, FZ, FY, EX, FX, EZ, EY, C. Here FZ, FY and FX are the face regions along the Z, Y and X regions respectively. EZ, EY and EX are the edge regions along the Z, Y and X regions respectively. According to the minimum image convention, we use the pair of images for each interaction which are closest to each other.

Here, I, is the homezone for the node and this node is responsible for updating the positions of the atoms in only the homezone. However, it calculates the interactions with other atoms in all the eighth zones as well.

2.1 Direct space calculations

Direct space calculations involve the calculation short-range component of non-bonded energy terms. Additional bonded terms like the bonds, angles, urey-bradley, improper dihedrals and dihedrals are calculated through their respective lists in the home-zone region.

2.1.1 Communication of coordinates

As the atoms involved in the pairwise interaction might not be local to the processor, coordinates of the atoms need to be communicated to the processor where the particular interaction will be calculated. A schematic figure of the communication is shown in Figure 2.2. The full communication is done in three steps:

1. Transfer of coordinates from Zone I (home zone) along the Z axis the receiving node stores these coordinates in the FZ region.
2. Transfer of coordinates from Zone I and FZ along the Y axis: The receiving node stores these coordinates in the FY and EX regions.
3. Transfer of coordinates from Zone I, FZ, FY, EX along the X axis: The receiving node stores these coordinates in the FX, EZ, EY and C regions.

As shown in Figure 2.1, P21 PBC involves a half screw symmetry - it has a 180◦rotation around the axis of symmetry and a half a unit cell length translation along the same axis. Without the loss of generality, we can choose the axis of rotation as the X-axis and have it pass along the center of the box. The images along the -X and +X are created by performing reflection operations along the Y and Z axis - i.e. rotation along the X axis. We modify the communication pattern of the nodes such that the boxes along the -X face which have an interaction with the boxes along the +X face, communicate to the rotated boxes. In order to send the correct set of atoms, we also make extra communication along the Z and Y axes for these nodes to prepare the extended import region (hence the name Extended Eighth Shell).

2.1.2 Communication of forces

Communication of the forces is an inverse operation of the communication of the coordinates. Forces accumulated on the image atoms are transferred back to the primary atoms at this stage. Forces are first communicated along the X-axis, then along the Y-axis and finally along the Z-axis.

Forces on images which are translated by the simulation box length along the X-axis are first rotated by 180 degrees before being transferred. These are the nodes which lie within cutoff region along the higher X axis. This is followed by communication along the Y-axis. For nodes which are within cutoff distance along the lower X axis, a second communication in the inverse direction is done as well. In the Z-communication, forces that were communicated along the X and Y axes are added to the local forces and communicated towards the higher Z axis. For nodes which are within the cutoff along the lower X axis, a second communication is performed towards the lower Z axis as well.

A final rotation of forces by 180 along the X-axis is done for atoms that are not located in the primary box dimension in the X-axis. As these atoms do not lie in the primary box, their homezone membership is decided on the basis of the corresponding image that lies in the primary box. Since the image is rotated along the X-axis, the corresponding forces are rotated back to generate the force on the original atoms. As the symmetry involves a 180 degrees rotation, we invert the direction of the forces for the interactions that happen due to the P21 symmetry.

2.2 Reciprocal space calculations

Reciprocal space nodes handle the long-range component of the nonbonded energy. Smooth Particle Mesh Ewald (SPME) method for the Ewald calculations occur in five stages:

1. spreading the charge on the k-space grid,
2. a backward 3D FFT,

3. calculation of energy by a scalar sum over the grid,
4. a forward 3D FFT, and
5. calculation of the forces on the atoms by a gradient sum over the grid.

In CHARMM, these calculations are performed using the column FFT method implemented in the colfft module. These calculations occur on separate set of reciprocal nodes. Each direct node communicates the coordinates of the atoms in its homebox to its peer reciprocal node. A reciprocal node might have more than one direct node. An all-to-all communication is performed among the reciprocal nodes such that each reciprocal node has all the "current" coordinates of all the atoms. Simulation box is then split into yz slabs and each node handles the calculation for the slab using the grids in the slab and the halo region from the neighboring slabs.

In order to use the colfft module for PME calculations, we use the internal symmetry in the unit cell. The unit cell is produced by applying the symmetry operation to the asymmetric unit. The full unit cell can then use the regular colfft module.

After the reciprocal nodes receive the coordinates from their peer direct nodes, they apply the 21-symmetry operation along the x-axis on all the atoms they received. These coordinates are then communicated among the reciprocal nodes through the all-to-all MPI communication. The k-space grid spans the full unit cell and not just the asymmetric unit.

Calculation of the forces happens on all the atoms - the asymmetric unit as well as the image. However, after the calculations and transfer of forces among the reciprocal nodes, only the forces on the asymmetric unit are transferred back to the direct nodes.

2.3 Bonded terms and Constraints

Coordinates for atoms involved in bonded (bonds, angles, improper dihedral and torsion) and SHAKE interactions might not be present on the same node. Each of these terms should be calculated only once on only one node. Similar to the non-P21 domdec, minimum of the x,y and z coordinates of the involved atoms is calculated first. Homezone for this coordinate is responsible for calculating the energy and force term for the interaction. Unlike the non-P21 domdec, if the coordinate lies beyond the X-boundary of the simulation box at the lower end, coordinate is first rotated by 180 degrees before being assigned to the node.

Constraints are also handled similar to the non-P21 conditions. Absolute harmonic restraints are calculated on the homezone for the atoms as they have the current coordinates of their atoms. Distance-matrix constraints on the other hand require coordinates of atoms located at arbitrarily different nodes. Communication for distance matrix constraints are hence similar to reciprocal nodes where an all-to-all broadcast is performed. Root node of the reciprocal nodes performs the constraints calculations and then communicates the forces back to the direct nodes.

2.4 Virial Calculation

Virial is used for the calculation of pressure and scaling of the simulation box during constant pressure calculation. It is defined as:

$$V = \sum_{ij} F_{ij} r_{ij}$$

For periodic systems with minimum image convention, this can be reformulated as:

$$V = \sum_{i} F_i r_i + S$$

Here, S, are accumulated forces along each of the 26 neighboring boxes and the primary box.

Both these set of calculations for virial is supported depending on whether the virial values are needed before or after the transfer of forces from the image to the primary cell atoms. Notably, since only orthorhombic simulation boxes are used, only the diagonal elements of the virial matrix are needed for scaling the box. Pressure calculation uses only the diagonal elements as well.

2.5 P21Tetragonal crystal type

Constant surface tension ensemble, which are often used in bilayer simulations, use the Tetragonal crystal in CHARMM. This crystal group has 2 degrees of freedom for the crystal: first linking the X- and Y- axes and the second for the Z-axis. In order to carry out constant surface tension simulations under the P21 periodic boundary conditions, we have added a corresponding crystal group (P21Tetragonal). As the asymmetric box length along the X-axis is only half the size of the unit cell (unlike the case of P1 PBC where the asymmetric unit length is same as the unit cell), the first degree of freedom scales the simulation box only half as much.

3 Results and Discussion

Our results demonstrate that chemical equilibrium between the lipid bilayers can be balanced by the use of periodic boundary conditions. Extended eighth shell method allows the rotational symmetry in the periodic boundary condition and thus offers an efficient way of performing lipid bilayer molecular dynamics simulations.

In order to check the correctness of the implementation, we first perform kappa-sweep tests of for energy and gradient. In these tests we varied the kappa and grid dimension. Results of these can be found in Table 2.1. For a system with 11,748 atoms, the sum of the reciprocal and direct space energies as well as the net root mean square of the gradient remains unchanged even with a range of kappa and coarseness of the grid used for the reciprocal space. Changing the kappa value

changes the component of non-bonded energy calculated in direct space vs reciprocal space. Changing the grid dimensions changes the value of charge on the grid points interpolated on the grids from the original charge distribution. While the individual components of direct and reciprocal space contributions change in the different tests, their sum remains the same.

Next, we look at the shadow Hamiltonian of the simulation. While the integrator does not sample the actual Hamiltonian, it does sample the shadow hamiltonian[15]. As shown in Figure 2.3, the mean of high frequency corrected total energy remains conserved over long time scale. For a system with 11,748 atoms, the energy drift per degree of freedom is less than 10^{-6} kcal/mol over 500 ps.

For constant pressure and/or constant surface tension simulations, lattice vectors are added as additional phase space coordinate[16]. Specifically, for constant surface tension simulations in CHARMM a tetragonal crystal is used. For this crystal type, there are two additional degrees of freedom: one for the X and Y and the other for Z. Since the X-dimension of the crystal is twice that of the asymmetric unit in P21 PBC, we added an additional P21 crystal type. As can be seen in Table 2.2, gradient of energy for these two degrees of freedom is very small.

The interaction neighborhood of a homebox can be divided into the following subregions:

1. Six face regions: cuboidal regions extending along each of the six faces of width r_{cutoff}
2. Twelve Edge regions: quarter cylinders of radius r_{cutoff} along each of the twelve edges
3. Eight corner regions: octants of radius r_{cutoff} at each of the eight corners

The import volume in ES for all nodes is[17]:

$$V_{in} = V_b[\frac{1}{6}\pi\alpha_r^3 + \frac{\pi}{4}\alpha_r^2(\alpha_x + \alpha_y + \alpha_z) + \alpha_r(\alpha_x^{-1} + \alpha_y^{-1} + \alpha_z^{-1})]$$

In the case of EES, all nodes except the ones with r_cutoff from the -X-edge will have the import region as the ES. Only the ones within the cutoff region, will have an additional cost of:

$$V_{in,extra} = V_b[\frac{\pi}{4}\alpha_r^2(\alpha_y + \alpha_z) + \alpha_r(\alpha_y^{-1} + \alpha_z^{-1})]$$

In these equations, V_b is the volume of the sub-box, $\alpha_z$is the reduced cutoff, α_x, α_y, α_z are the reduced length along the x, y and z axes respectively. The slight extra cost in communication is not in the critical region of the runtime (non-bonded force calculation) and hence it does not impact the speed of the simulation. Within the communication of coordinates, the extra load is only during the Z and Y communication of coordinates - which are also the smallest of the communications. As the extra load during the Z and Y axis communication are asynchronously coupled to the normal eighth communication, its cost remains hidden and does not impact the efficiency of the run. Communication cost is limited by the latency of inter process communication and not its bandwidth.

Lipids migrate from one layer to another in the P21 PBC. This is important during the simulations where peptides/small molecules insert into the bilayer and the area per lipid (APL) and surface

tension (ST) is different between the two layers. As a lipid moves out of the box along the X-axis, it enters back into the box according to the P21 PBC. According to the symmetry, the lipid enters into the second layer. This is not possible with P1 PBC as it has only translational symmetry and the lipid leaving the simulation box enters into the same layer.

The EES based P21 design is different from the earlier P21 in non-domdec CHARMM [10]. The initial setup of the crystal for non-domdec CHARMM involves an initial rotation around the Z-axis by 45 degrees and quarter of side-length translation along the diagonal. In the present implementation, the axis of symmetry passes through the center of the box as compared to the previous one where it passed through a point a quarter along the diagonal. In the older method, lipids enter back into the box into the other layer but along the orthogonal face of the box. In contrast, lipids enter the opposite back in the alternate layer in the new implementation. Hence, we should not expect the two simulations to give same trajectory.

The domain decomposition implementation is targeted only for orthorhombic while the original one can work for any monoclinic space group P21 preserves the chirality unlike Pc. We implement the space group P2111, i.e. the symmetry operation along the X-axis is 2_1 while just translational along the Y and Z-axis. In contrast, the previous method can possibly use any of the three axes as the screw-axis. We do not believe this to be a limitation in any way as the simulation box can be rotated to align the X-axis with the screw axis.

Even though we should not expect the trajectory obtained from the previous P21 and the EES based P21 to be similar, we would like to similar properties from the fact that both these

simulations allow the movement of lipids between the two layers. In Figure 2.6a, the number of lipids in the top layer of a DOPC bilayer simulation is compared in two simulations. The number of lipids fluctuate around 40 for both the simulations. Additionally, we checked the amount of time that the lipid spends in the two layers. Similar to the older P21, on an average lipid counts are 40 in each layer. Finally, we interpolated this data to area per lipid for the DOPC lipids. During the course of the simulation with constant pressure (i.e. changing box sizes), area per lipid remains close to it experimental value of 69 A^2.

The two layers should be considered as a torus such that the top layer is the bottom layer is the top layer. The two layers continue into each other. Generally, only the lipids near the X-faces migrate between the two layers. But this depends on the system being simulated. We are further investigating its application in the simulation of asymmetric layers.

In order to show the application of the EES method, we started a simulation with 40 and 32 lipids on the top and the bottom leaflets [Figure 2.5]. In a normal P1 simulation, no lipids will move from the top to the bottom layer or vice-versa. However, under the P21 PBC, 4 lipids move from the top to the bottom layer and system equilibrates. Based on the interactions during the course of the simulation, lipids continue being exchanged between the two layers.

As mentioned earlier, images around the simulation unit for P21 are different than the P1 PBC. A system prepared under P1 cannot be directly used for P21 simulation as it would lead to clashes between the images and make the system unstable. For the systems presented in this paper, we

generated the initial system via CHARMM-GUI and minimized it under P21 PBC. This is followed by equilibration under P21 before the final production run.

Using only the X axis as the axis of symmetry allows us to limit the import region for each node and hence the import volume. The system can always be rotated in order to adhere to the axis and does not pose as a limitation in its usage. Judicious use of asynchronous communication ensures that although the import volume is slightly higher, it does not show up in the cost as it can be performed while the node waits for the other communication (during Y and Z communications only).

4 Conclusion

This work solves a long-standing problem in the field of lipid molecular dynamics simulations of running efficient simulations with P21 periodic boundary conditions. Extended Eighth Shell (EES) method is adapted from De Shaws Eight Shell method to handle rotations in the symmetry of the periodic boundary conditions. Judicious use of the asynchronous communication pattern allows the simulation to run at the same speed as the ES counterpart.

There are several ways of further optimizing the implementation of EES. Load balancing should shrink the volume of the regions along X-axis boundaries so that each node performs similar amount of work. Calculating the distance of each group center from the face boundaries can be independently parallelized over a number of threads. It is especially suited for multi-GPU implementations in the future as the NVLink based inter-GPU communication bandwidth increases further. We are working on a GPU based implementation of the P21 PBC as well.

The EES implementation in domdec is available in CHARMM version c43a2 version and later. Examples and usage are described in the code.

5 Acknowledgement

The authors would like to thank Richard Pastor, Richard Venable and Mohsen Pourmousa for helpful discussion. Samarjeet Prasad would like to thank the BCMB program at JHMI for supporting the training. We would like to thank LoBos and Biowulf team for the computing resources.

6 Declaration of Interest

None

7 References

1. Kutzner, C.; Páll, S.; Fechner, M.; Esztermann, A.; De Groot, B. L.; Grubmüller, H. Best Bang for Your Buck: GPU Nodes for GROMACS Biomolecular Simulations. *Journal of Computational Chemistry*. **2015**.
2. Shaw, D. E. A Fast, Scalable Method for the Parallel Evaluation of Distance-Limited Pairwise Particle Interactions. *J Comput Chem* **2005**, *26*, 1318–1328.
3. Bowers, K. J.; Dror, R. O.; Shaw, D. E. Overview of Neutral Territory Methods for the Parallel Evaluation of Pairwise Particle Interactions.

4. David E. Shaw. A Fast, Scalable Method for the Parallel Evaluation of Distance-Limited Pairwise Particle Interactions,Journal of Computational Chemistry (2005)26(13)1318-1328. *J. Comput. Chem.* **2005**, *26* (16), 1803–1803.
5. Bowers, K. J.; Dror, R. O.; Shaw, D. E. The Midpoint Method for Parallelization of Particle Simulations. *J. Chem. Phys.* **2006**, *124* (18), 184109.
6. Hess, B.; Kutzner, C.; Van Der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation.
7. Hynninen, A.-P.; Crowley, M. F. New Faster CHARMM Molecular Dynamics Engine. *J. Comput. Chem.* **2014**, *35* (5), 406–413.
8. Perrin, B. S.; Pastor, R. W.; Pastor, R. W. Simulations of Membrane-Disrupting Peptides I: Alamethicin Pore Stability and Spontaneous Insertion. *Biophys. J.* **2016**, *111* (6), 1248–1257.
9. Perrin, B. S.; Fu, R.; Cotten, M. L.; Pastor, R. W.; Pastor, R. W. Simulations of Membrane-Disrupting Peptides II: AMP Piscidin 1 Favors Surface Defects over Pores. *Biophys. J.* **2016**, *111* (6), 1258–1266.
10. Dolan, E. A.; Venable, R. M.; Pastor, R. W.; Brooks, B. R. Simulations of Membranes and Other Interfacial Systems Using P21 and Pc Periodic Boundary Conditions. *Biophys. J.* **2002**, *82* (5), 2317–2325.
11. Park, S.; Beaven, A. H.; Klauda, J. B.; Im, W. How Tolerant Are Membrane Simulations with Mismatch in Area per Lipid between Leaflets? *J. Chem. Theory Comput.* **2015**, *11* (7), 3466–3477.
12. Rui, H.; Root, K. T.; Lee, J.; Glover, K. J.; Im, W. Probing the U-Shaped Conformation of Caveolin-1 in a Bilayer. *Biophys. J.* **2014**, *106* (6), 1371–1380.

13. Rui, H.; Im, W. Protegrin-1 Orientation and Physicochemical Properties in Membrane Bilayers Studied by Potential of Mean Force Calculations. *J. Comput. Chem.* **2010**, *31* (16), n/a-n/a.
14. Rui, H.; Kumar, R.; Im, W. Membrane Tension, Lipid Adaptation, Conformational Changes, and Energetics in MscL Gating. *Biophys. J.* **2011**, *101* (3), 671–679.
15. Merz, P. T.; Shirts, M. R. Testing for Physical Validity in Molecular Simulations. **2018**.
16. Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. Constant Pressure Molecular Dynamics Simulation: The Langevin Piston Method. *J. Chem. Phys.* **1995**, *103* (11), 4613–4621.
17. Bowers, K. J.; Dror, R. O.; Shaw, D. E. Zonal Methods for the Parallel Execution of Range-Limited N-Body Simulations. *J. Comput. Phys.* **2007**, *221* (1), 303–329.

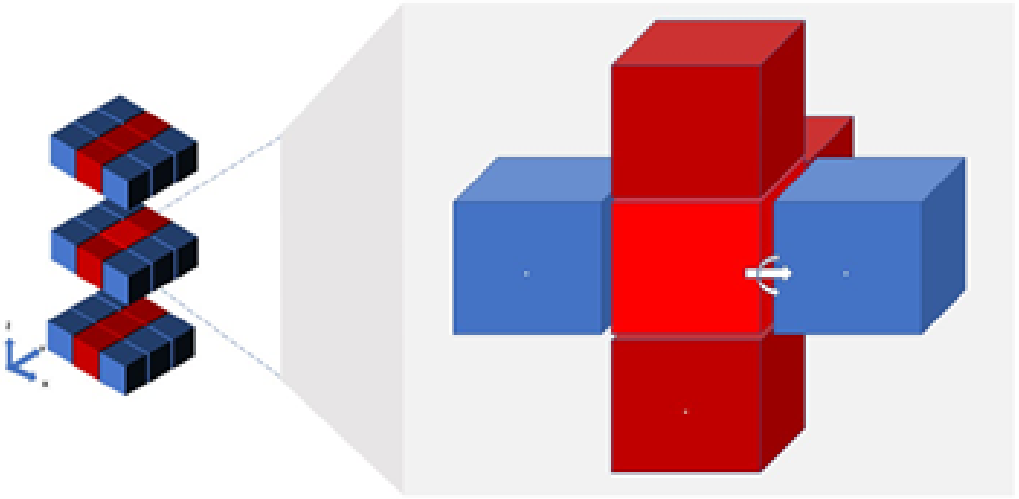


Figure 2.1: Neighboring cells of the primary simulation box. Blue colored cells are rotated images of the primary cell. Red cells are only translated with respect to the asymmetric unit.

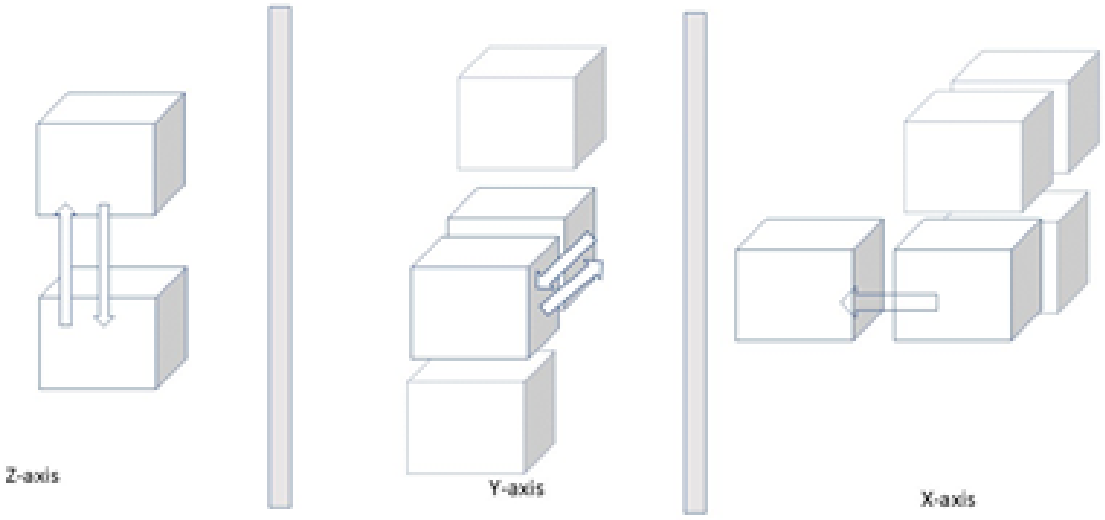


Figure 2.2: Communication of coordinates between nodes during in EES. During the communication along Z-axis, coordinates within the cutoff region from the region handled by the box are communicated in both the -Z and +Z directions. In the second set of communications along the Y-axis, the coordinates communicated from the previous step and from the present node, which are within the cutoff region are communicated along the -Y and +Y axes. In the final step, only communication along the X-axis in the -X direction is needed. The nodes which are within cutoff region from the -X border (i.e. the ones which interact with the image atoms, communicate with rotated version of the node configuration). Other nodes communicate to the nodes adjacent along the -X-axis.

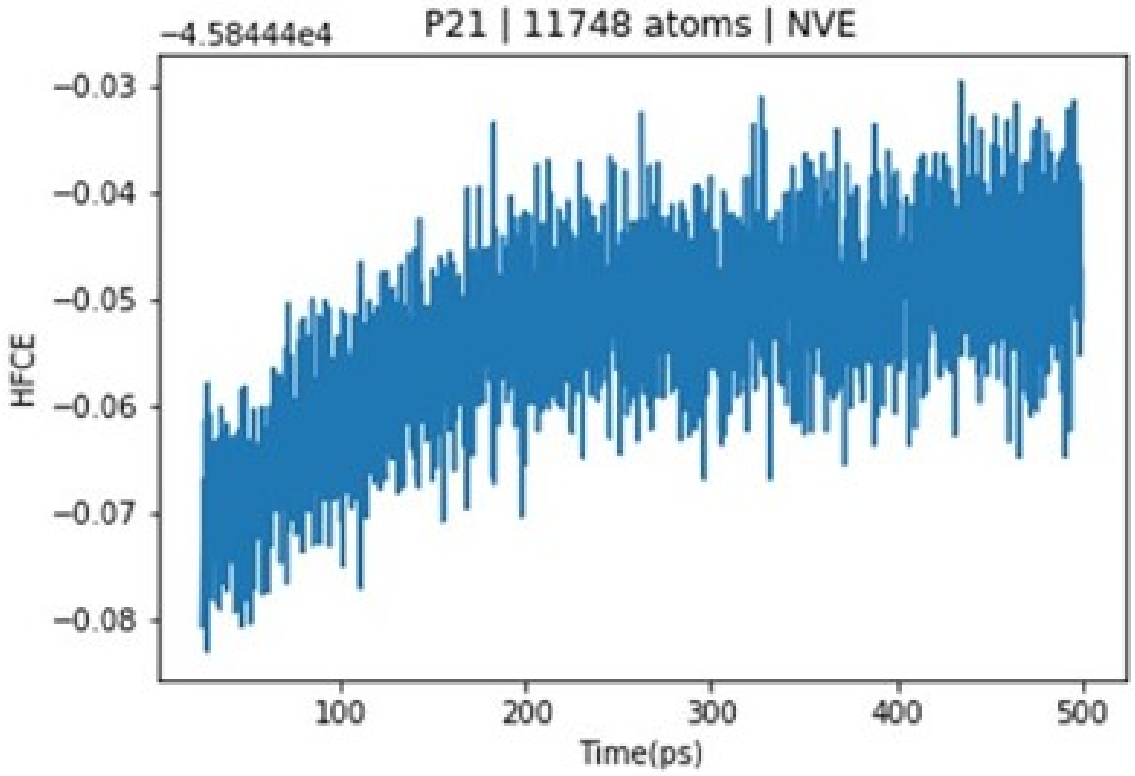


Figure 2.3: Shadow Hamiltonian is conserved in a micro-canonical simulation. For a system with 11,748 atoms, the energy drift per degree of freedom is less than 10-6 kcal/mol over 500 ps.

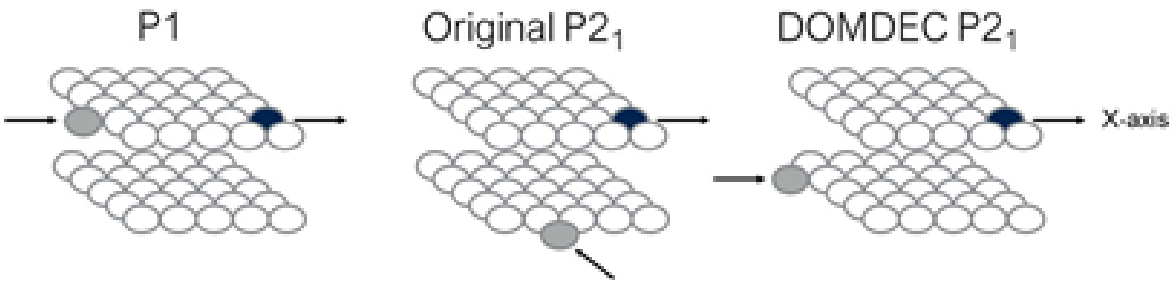


Figure 2.4: Comparison of P1, non-EES $P2_1$ and the new EES-$P2_1$. In a P1 PBC simulation, lipid leaving one layer enters back into the same layer. In the previous $P2_1$ PBC, lipids leaving one layer, enters into the other layer along the orthogonal face. In the new $P2_1$ PBC, based on the EES scheme, lipids leaving the simulation along the YZ face enters back in the YZ face but in the opposite layer.

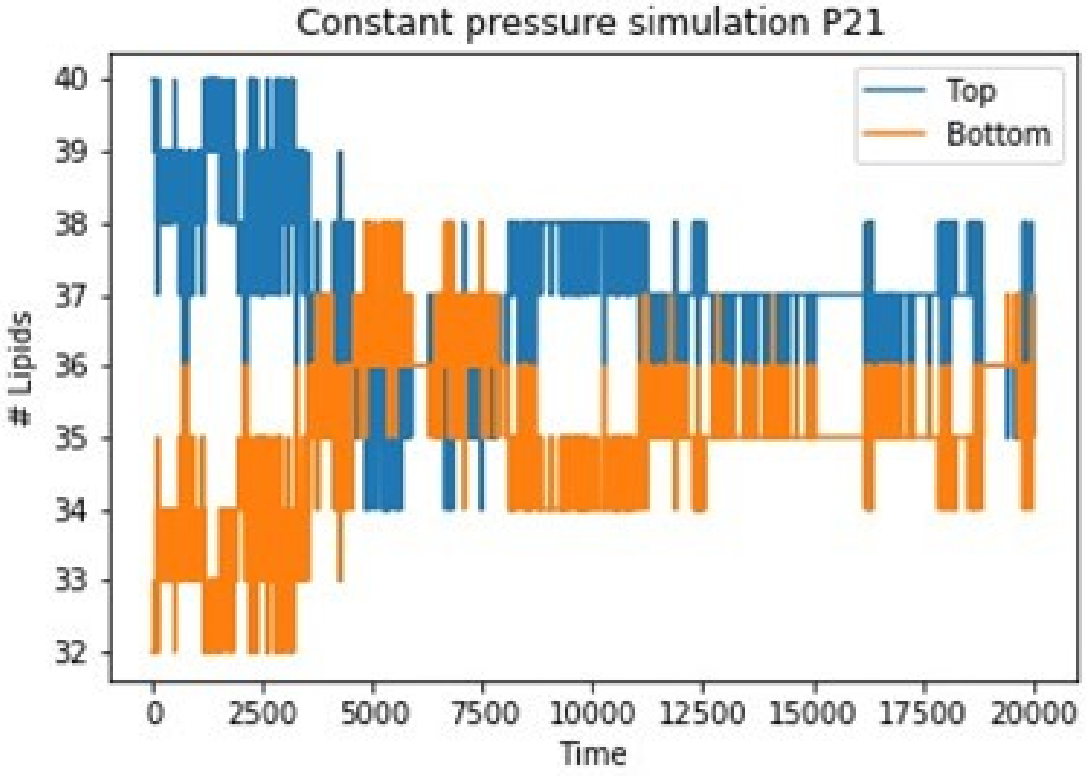


Figure 2.5: Number of lipids in the top and bottom layers during with the EES scheme. Simulation was started with 40 lipids in the top layer and 32 lipids in the bottom layer. Lipids from the top layer move to the bottom later within 500ns and then remain in equilibrium.

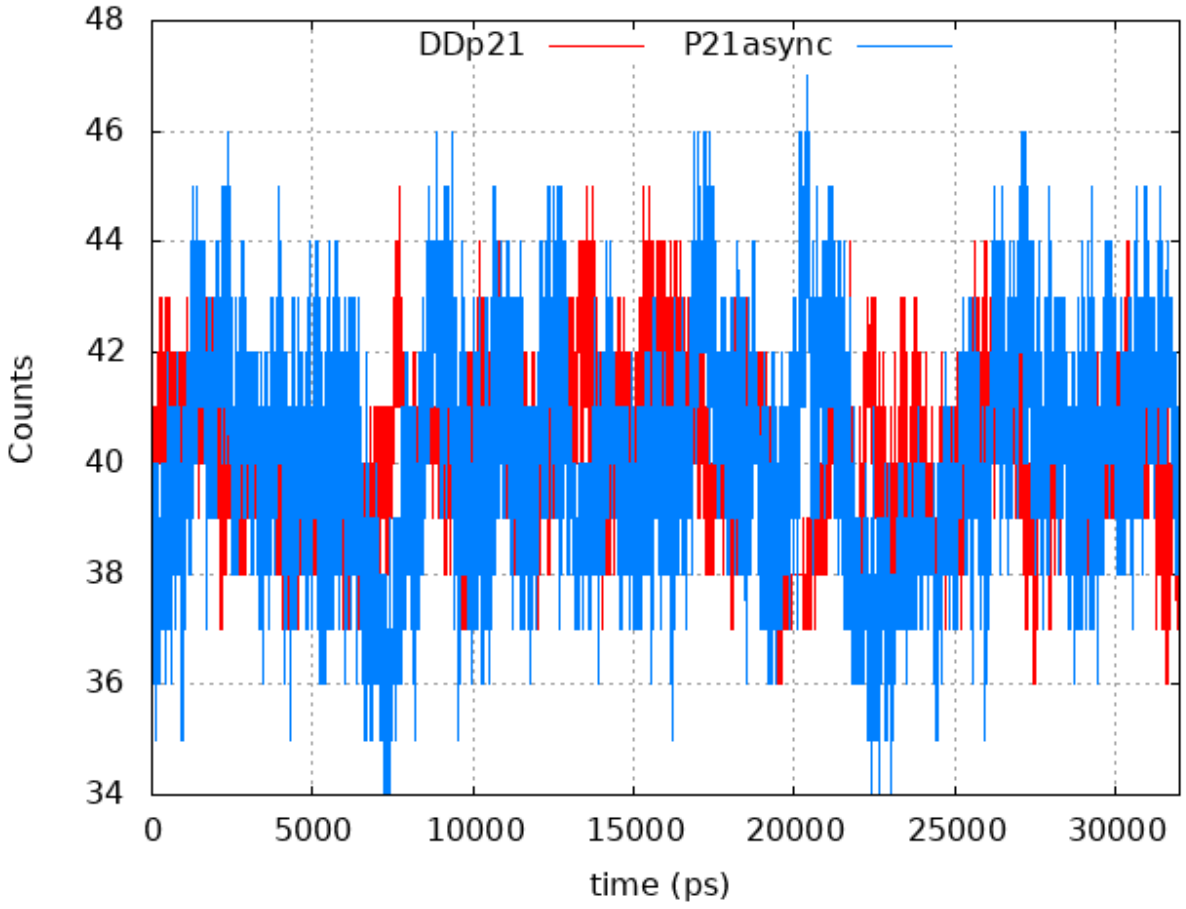
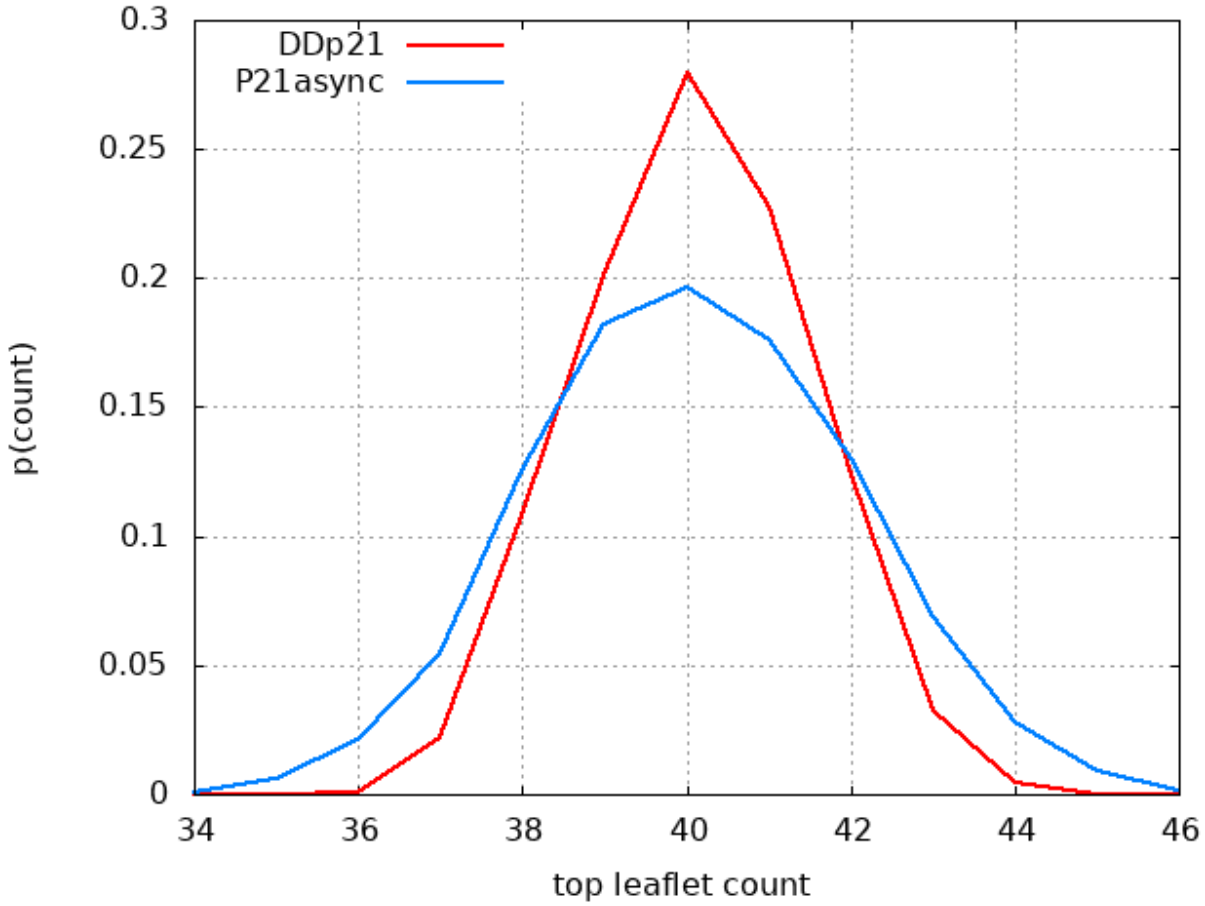
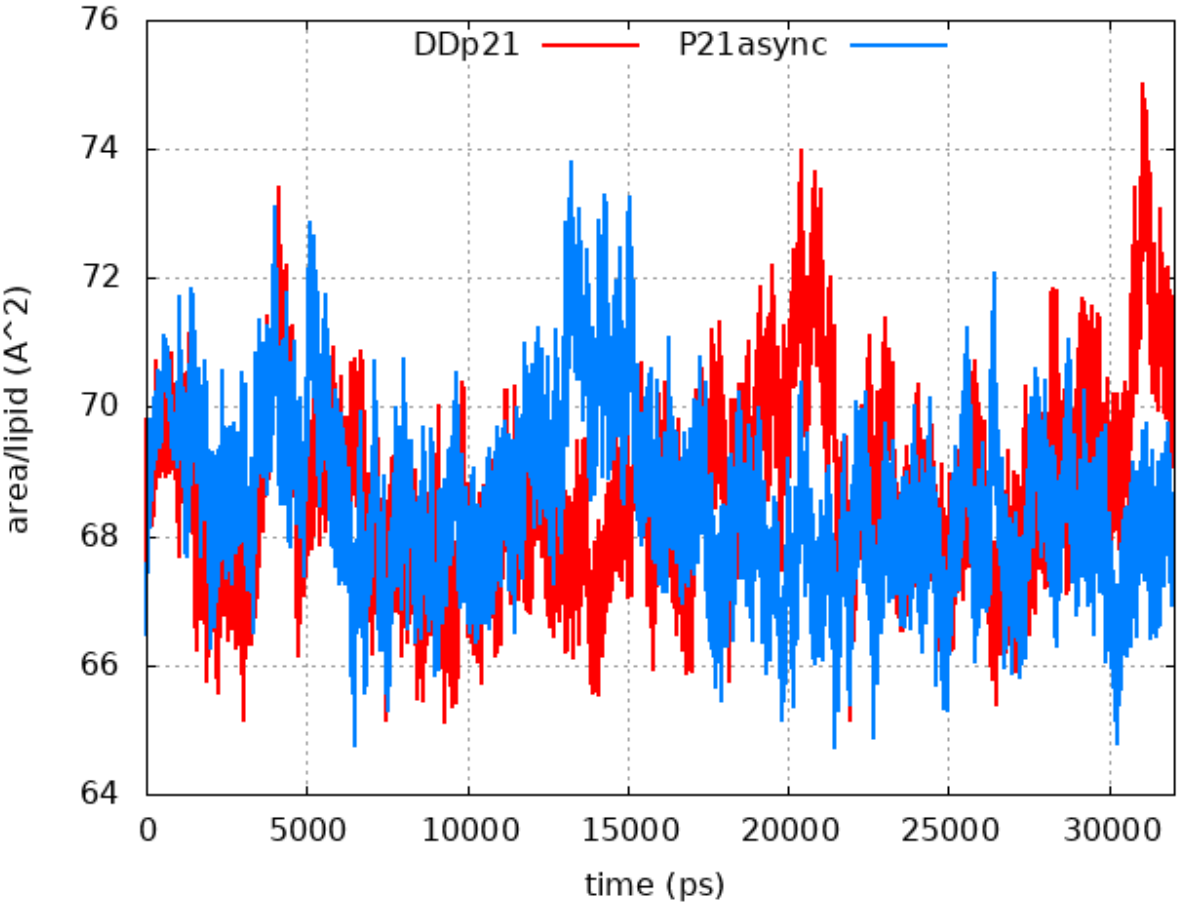


Figure 2.6:

a. Comparison of EES based P21 vs original P21 in CHARMM (referred here as P21async). Both simulations start with 40 lipids on the two layers. Lipids exchange between the two layers and that can be seen in the fluctuation in the number of lipids.



b. Comparison of EES based p21 vs original P21 is CHARMM (referred here as P21async). Counts vs number of lipids in the top layer. For most of the simulation, 40 lipids remain in the top layer although it can move back and forth between the layers.



c. Comparison of EES based P21 vs original P21 is CHARMM (referred here as P21async). Area per lipid for DOPC remains in the experimentally observed 69A2 on an average.

kappa	grid_dim	direct space	reciprocal space	Potential energy	grms
0.34	128	-54784.20381	398.56020	-48788.03453	2.31140
0.40	128	-51576.03953	646.63444	-48788.03817	2.31140
0.34	256	-54784.20381	398.56040	-48788.03433	2.31140
0.28	256	-57038.79833	252.26083	-48788.04996	2.31140

Table 2.1: Kappa sweep for validation check. For a system with 11,748 atoms, the sum of the reciprocal and direct space energies as well as the net root mean square of the gradient remains unchanged even with a range of kappa and coarseness of the grid used for the reciprocal space. Changing the kappa value changes the component of non-bonded energy calculated in direct space vs reciprocal space. Changing the grid dimensions changes the value of charge on the grid points interpolated on the grids from the original charge distribution.

Step size	dof1	dof2
0.1	0.20195728	0.01248617
0.01	0.00232570	0.00014901
0.001	0.00001016	0.00000048
0.0001	0.00000065	0.00000103

Table 2.2: Finite difference tests for crystal degree of freedom for P21Tetragonal crystal type in CHARMM. There are two additional degrees of freedom, the first linking X and Y and second for the Z-axis. For small step sizes along these vectors, the change in energy is very small.

Chapter 3

An explicit-solvent hybrid QM and MM approach for predicting pKa of small molecules in SAMPL6 challenge

(A version of this chapter appeared in the Journal of Computer Aided Drug Design October 2018 special issue.)

Abstract

In this work we have developed a hybrid QM and MM approach to predict pKa of small drug-like molecules in explicit solvent. The gas phase free energy of deprotonation is calculated using the M06-2X density functional theory level with Pople basis sets. The solvation free energy difference of the acid and its conjugate base is calculated at MD level using thermodynamic integration. We applied this method to the 24 drug-like molecules in the SAMPL6 blind pKa prediction challenge. We achieved an overall RMSE of 2.4 pKa units in our prediction. Our results show that further optimization of the protocol needs to be done before this method can be used as an alternative approach to the well-established approaches of a full quantum level or empirical pKa prediction methods.

1 Introduction

Computational prediction of pKa values is of considerable interest for a number of fields including pharmaceutical and material sciences[1, 2, 3]. Even though several methods have been developed to predict this value, the problem still remains a challenge[4, 5, 6]. Most prediction methods can be divided into two broad categories - empirical and ab initio ones. The first set of methods use a cheminformatics-based approach[7, 8, 9]. In this approach the compound is represented as a vector of molecular descriptors including constitutional, topological, electrostatic and quantum

descriptors[10]. Machine learning models for specific functional groups are trained based 11 on these descriptors[10]. Notably, these methods ignore the three-dimensional conformation of the compound explicitly[11]. Although training the models might be expensive in terms of curating experimental pKa data for generating appropriate models, subsequent pKa prediction using trained models can be very fast and inexpensive.

Ab initio methods use a thermodynamic cycle combining with quantum mechanics (QM) calculations to compute the solvent-phase pKa[12, 13, 14, 18 15, 16, 17, 18, 19, 20]. It consists of the calculation of dissociation free energy in gas phase[21] along with solvation free energy of the acid and the conjugate base using dielectric continuum solvation models (DCSMs)[22, 12, 23, 24, 25]. These methods have been very successful in calculating pKa. However, DC22 SMs cannot model the hydrogen bonding between solute and water, which can be important in the protonation or deprotonation process[26]. Their ac24 curacy in describing the short-range electrostatics of polar solutes and ions is also limited[12]. Moreover, typically only one conformation is used for the estimation of free energy although an ensemble of conformations is required for a complete statistical mechanics treatment of the free energy[27]. Even if multiple low-lying conformations are included in the calculation, the entropic variations associated with the deprotonation process still cannot be completely accounted for without explicitly considering the solvent dynamics and extensively exploring the potential energy landscape of the solute-solvent systems.

Calculation of solvation free energy during pKa estimation remains one of the bottlenecks in getting accurate values. An alternative way of calculating solvation free energy is to use molecular dynamics simulations with empirical force field[28, 29, 30]. Shirts et. al. were able to do a very precise

measurement of solvation free energy with 0.85 kcal/mol RMSE[31]. Gilson et. al. used double decoupling method and achieved 1.3 kcal/mol RMSE. K¨onig et. al.[29] used the annihilation approach and obtained accuracy on par with the quantum calculations. Mobley et. al. have created the FreeSolv[30] database to catalog molecules with known experimental solvation free energy and assist in the development of new methods from these resources.

Given the large number of diverse methods available for predicting pKa, the Statistical Assessment of the Modeling of Proteins and Ligands (SAMPL)[32] blind prediction challenge was organized to assess the methods on a common set of small drug-like molecules. Previous iterations of the SAMPL competitions have focused on assessing methods for solvation free energy calculations[33], distribution coefficient and other challenges[34, 35, 36, 37]. We note that in the SAMPL5 distribution coefficient competition, Pickard and coworkers have calculated pKa values with QM methods, and used computed pKa to further correct their prediction of distribution coefficients[34].

In this work we have presented a new method to computationally predict the pKa of small drug-like molecules in explicit solvent. This is a hybrid QM and MM approach that allows ab initio prediction of absolute pKa values and supports any chemistry. Since calculation of pKa requires relative solvation free energy between the acid (protonated species) and the conjugate base (de protonated species), our method calculates this quantity directly rather than computing the absolute solvation free energies of both by employing two thermodynamic cycles.

This paper is organized as follows. In Section 2, we describe the theory behind the prediction of the microscopic and macroscopic pKa values. Section 3 covers the details of the description of the

QM and MM methods that we used to carry out calculations. Next in Section 4, we present our results that submitted to the SAMPL6 competition and analyze the accuracy of the results. Finally, in Section 5, a brief conclusion is provided.

2 Theory

SAMPL6 pKa challenge involved blind computational prediction of pKa of 24 small drug-like molecules (Figure 3.1). These molecules were similar to kinase inhibitors and were chosen for experimental tractability. All the molecules were polyprotic in nature i.e. there were multiple sites on each molecule where the molecule could lose a proton. For further details, please refer to Isik et. al[38] 71 where the organizers have described the rationale for choosing the molecules as well as the methods used for experimental pKa prediction.

In order to compare the computational and experimental pKa predictions, it is important to understand the difference between the microscopic and macroscopic pKa of a molecule. The chemical environment around a functional group (in this case, the protonation state of other titrable moieties) affect the propensity of the group to lose its proton. This is referred to as the microscopic pKa, i.e. pKa for deprotonation at a site at a fixed protonation state of all other titratable sites in the molecule. This differs from the macroscopic pKa which is related to the dissociation constant of losing a proton from the molecule as a whole and can be experimentally measured. Converting microscopic pKas to macroscopic pKas or vice versa is complicated due 83 to the large number of equilibrium processes involved[8, 39]. If, for a specific charge transition, the microscopic pKas are fairly well separated (ex. more than one pKa unit), the smallest pKa can be considered as the

macroscopic pKa. However, if they are close, the macroscopic pKa is shifted as multiple microscopic transitions contribute to the macroscopic value. Several studies[40, 41] discuss this in greater detail. In our method, we calculate microscopic pKa value for each acid-base pair of microscopic states. We then assign one dominant microscopic pKa as the macroscopic pKa for each titration process, which can be directly compared with the experimental observables.

To calculate the microscopic pKa of a particular acid-base pair, let us consider the dissociation of acid HA:

$$HA_{aq} \leftrightarrow H^{+}_{aq} + A^{-}_{aq}$$

Here the subscripts "aq" indicate that the species are solvated in water. The dissociation constant and pKa value for this dissociation are given by the following relations,

$$K_a = \frac{[H]^{+}_{aq}[A]^{-}_{aq}}{[HA]_{aq}}$$

where,

$$\Delta G^{*}_{aq} = G^{*}\left(H^{+}_{aq}\right) + G^{*}\left(A^{-}_{aq}\right) - G^{*}(HA_{aq})$$

Here, G refers to the absolute Gibbs free energy of the solvated species. The superscript * implies that the standard state of one mole per liter and 298.15 K have been used. R and T are the gas constant and the absolute temperature respectively. Thus, to calculate pKa we need to calculate aqueous phase deprotonation free energy ΔGaq.

Rather than calculating the absolute free energies in the aqueous phase directly, the aqueous phase calculations are coupled with gas phase calculation using the following thermodynamic cycle (Figure 3.2a). The two vertical lines in the figure refer to the solvation of the species into aqueous phase. Thus, the ΔGaq can be calculated as:

$$\Delta G^*_{aq} = G^*\left(H^+_{aq}\right) + G^*\left(A^-_{aq}\right) - G^*(HA_{aq})$$

The absolute free energy for proton H+ in the gas phase at standard temperature and pressure is calculated by Sackur-Tetrode equation and has been previously calculated as -6.28 kcal/mol[42]. Solvation free energy of proton (-264.5 kcal/mol) has been taken from Tissandier et. al.[43]. The gas phase calculations are done at standard gas conditions i.e. one atmosphere of pressure. Converting them to 1 mole/liter further involves a standard state correction 115 of -1.89 kcal/mol.

The above equation involves the calculation of solvation free energies of the deprotonated ΔG*solv(A−) and of the protonated species ΔG∗solv(HA), respectively. Most ab initio pKa prediction methods compute them in implicit solvent using quantum chemistry and continuum solvent approaches. We note that, however, the only relevant quantity for pKa prediction is the difference of solvation free energies

In the present work, we directly compute this solvation free energy difference in explicit solvent. The calculation is done at the force field level in order to be computationally tractable. Furthermore, we consider a second thermodynamic cycle (Figure 3.2b) that alchemically change HA into A− in the gas and the aqueous phases. As we are interested in only the free energy

difference between the two species HA and A− and free energy is a state function so that its sum over a thermodynamic cycle equals zero, we can rewrite $\Delta\Delta G^*_{solv}$ as,

$$\Delta\Delta G^*_{solv} = \Delta G^*_{solv}(A^-) - \Delta G^*_{solv}(HA) = \Delta G^*_{deprot,aq}(HA) - \Delta G^*_{deprot,aq}(A^-)$$

where, $\Delta G^*_{deprot,aq}(HA)$ can be calculated using free energy perturbation (FEP) methods such as the thermodynamics integration (TI) method. By introducing a number intermediate λ states that alchemically connecting two states 0 and 1, the free energy difference between the two-end state is computed by TI as:

$$\Delta G = \int_0^1 < \frac{\partial U}{\partial \lambda} >_\lambda d\lambda$$

It's worth noting that for each acid-base pair only one relative free energy in the aqueous phase is computed, rather than two absolute solvation free energies. It has previously been shown by Jorgensen et. al[44] that this allows the cancellation of errors in MM calculations such as inaccuracy of force field parameters and inadequate conformational samplings. In their work they calculated the relative solvation free energy of methanol and ethane using alchemical transformation of methanol to ethane and vice versa and got results close to experimental relative solvation free energy value. The major advantage of using such a secondary thermodynamic cycle (Fig. 3.2b) is that the alchemical FEP only involves changing HA into A− in the gas and the aqueous phase, instead of annihilating whole molecules in the aqueous phase. This greatly improves the efficiency, accuracy and the throughput of our calculations.

In summary, we calculate the ΔG^*_{aq} by the following equation,

$$\Delta G^*_{aq} = \Delta G^*_g + \Delta G^*(H^+) + \Delta G^*_{deprot,aq}(HA) - \Delta G^*_{deprot,aq}(A^-)$$

where, ΔG^*_{aq} is calculated in the gas phase at the QM level, $\Delta G^*(H^+)$ is obtained from experimental value reported in literature,) $\Delta G^*_{deprot,aq}(HA)$is calculated using FEP in condensed phase at the MM level and $\Delta G^*_{deprot,aq}(A^-)$ in gas phase at the MM level.

3 Methods

The workflow for the complete method is shown in Figure 3.3. First the geometry of each microstate was optimized in gas phase. Then for each acid (protonated) - base (deprotonated) pair, ΔG for deprotonation in gas phase was calculated at the QM level. To carry out the MM simulations, force field parameters were generated for each of the microstates. Next, the gas phase and aqueous phase alchemical free energy difference between each acid-base pair were computed using FEP and MD simulations. All the QM calculations were performed with Gaussian16[45], while all the MD simulations were done with CHARMM[46, 47].

3.1 Geometry optimization and gas phase QM calculation

SAMPL6 pKa challenge had 24 molecules, each with different number of microstates. SMILES[48] string of the microstates were converted to PDB files using OpenBabel[49]. Geometry optimization and gas phase deprotonation energy was calculated with the M06-2X density functional theory[50]

and 6-31G* basis set for neutral-cationic microstate pairs and 6-31+G* for 166 neutral-anionic microstate pairs. Ultrafine grid and Tight convergence criteria were used in all calculations.

We would like to point out that as the computed pKa are directly related to the calculated electronic energies, higher-level methods such as MP2 and larger basis sets such as cc-pVTZ would improve calculation results. These, however, have not been pursued in this study. We also did not test other functionals, which might potentially lead to better pKa prediction results.

3.2 Parameterization of microstates

In order to carry out molecular dynamics simulations, we first generated force field parameters for the microstates based on the fixed-charge molecular mechanics potential energy functions used in CHARMM[51]. The potential energy is given by a sum of bonded and non-bonded components,

$$U = U_{bonded} + U_{non-bonded}$$

where,

$$U_{bonded} = \sum_{bonds} K_b(r_{ij} - r_0)^2 + \sum_{angles} K_\theta(\theta_i - \theta_0)^2 + \sum_{dihedrals} K_X(1 + \cos(n_x - \delta)) + \sum_{impropers} K_{imp}(\phi - \phi_0)^2$$

and,

$$U_{non-bonded} = \sum_{ij} \frac{q_i q_j}{4\pi \in_0 r_{ij}} + \in_{ij} \left[\left(\frac{R_{min}}{r_{ij}}\right)^{12} - \left(\frac{R_{min}}{r_{ij}}\right)^{6} \right]$$

Here, Kb and r0 are bond force constant and equilibrium bond-length for each atom type pair. Kθ and θ0 are angle force constant and equilibrium angle for each angle type triplet. Kimp and φ0 are improper angle force constant and equilibrium improper angle for each improper angle. Kχ, n, and δ are the force constant, periodicity, and phase for each torsional degree of freedom. The nonbonded potential energy terms involve Coulombic interactions between partial charge qi and qj, and the van der Waals (VdW) interactions modeled by the and Rmin parameters.

We used Antechamber to generate GAFF parameters. Single point calculation was done on the optimized geometry mentioned above using Gaussian16 at MP2 level of theory with 6-31G* basis set. RESP charges were calculated using the protocol mentioned in Jakalian et.al.[52]. Electrostatic potential was written in a data file using the option IOp (6/33=2) in Gaussian, and the RESP charges were fitted. Other parameters - bonded (bond, angle and torsion) and non-bonded (van der Waals) were assigned as per the General Amber Force Field (GAFF)[53] using the Antechamber[52] program in the AmberTools16 software. CHARMM formatted parameter and topology files were produced. These parameters were modified by in-house scripts to make the formats compatible with CHARMM molecular dynamics package. If the residues did not have an integer charge in the generated topology file (typically off by ±0.0 − 0.003), an ad-hoc fix was done by adjusting the charge on a random non-hydrogen atom to round up the total charge of residue.

3.3 Free energy simulations

All molecular dynamics simulations were carried out with CHARMM[47] and parameter sets mentioned in the previous subsection. Thermodynamic Integration calculations were carried out using the PERT module of CHARMM. 12 λ windows were used (0.0. 0.075, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.00) for transforming the partial charges of the acid into those of the conjugate base, with the charge on the dissociating proton transforming to zero. Each λ window was equilibrated for 1 ps followed by 10 ps MD simulations for sampling.

MD simulations in the gas phase were carried out with Langevin dynamics at a temperature of 298 K and using a time step of 2 fs with a friction coefficient of 5 ps−1 on all the atoms. No cutoffs were used in calculation of nonbonded interactions for gas phase simulations. For aqueous phase simulations, we used 2022 water molecules to solvate the solute molecule, constituting a 38 ˚A cubic water box to start with. 50 ps NPT simulations were run at 298 K and 1 atm, after which NVT simulations at 298 K were carried out for TI calculations. A Nose-Hoover thermostat[54] was used to maintain the microcanonical ensemble. Particle mesh ewald[55] was used to calculate the long-range electrostatic interactions with a direct space cutoff of 10 ˚A. Charge was spread on a grid of 48×48×48 for reciprocal space calculation using 6th order B-spline interpolation method[56]. A cutoff of 12 ˚A was applied for van der Waals interactions, and the integration time step is 1 fs.

4 Results and Discussion

The results discussed in this report are the ones that we submitted for the SAMPL6 competition [submission id: 0wfzo]. We submitted only the microscopic pKas for all acid-base pairs of all the

24 molecules. These results were compared to macroscopic pKas using two different approaches - closest and Hungarian. This analysis was done with the assumption that experimentally observed pKas with only one observed pKa or fairly-distant pKas (separated by more than 3 units) are equal to the microscopic pKa of the corresponding microscopic pKa. Only two molecules - SM14 and SM18 - did not satisfy this criterion and hence they were excluded from this analysis. Detailed analysis of the results can be found at https://github.com/MobleyLab/SAMPL6/tree/master/physical_properties/pKa/analysis/analysis_of_typeI_predictions.

In the closest analysis approach, the experimentally observed pKa is matched with the microscopic pKa which minimizes the absolute error i.e. the one that is closest to the observed pKa. We achieved a root mean squared error (RMSE) of 2.42 pKa units with respect to the experimental values. The mean absolute error (MAE) was 1.61 pKa units. The corresponding R2 for regression fit was 0.53 and the slope of line was 1.08.

In the hungarian approach[57], an optimum global match between experimentally observed pKa and predicted set of pKas is found by minimizing the linear sum of squared errors of the paired match. We achieved a root mean squared error (RMSE) of 2.89 pKa units with respect to the experimental values. The mean absolute error (MAE) was 1.88 pKa units. The corresponding R2 for regression fit was 0.48 and the slope of line was 0.99.

Out of the 22 molecules whose results were compared to experimental results, 3 of the molecules (SM06, SM15 and SM22) had 2 macroscopic pKas in the 2-12 pKa range while the other molecules had just 1 pKa in this range. Among these 25 comparisons, only 5 predictions were

more than 2 pKa units away from the experimental values. The most erroneous one concerns SM15, of which the first predicted pKa underestimated the experimental measurement by 8.86 pKa units, and the second pKa overestimated by 3.52 pKa units.

In general, our results compare less favorably to some of the more-established methods of pKa prediction, as used by other submissions in the SAMPL6 challenge. By carefully examining our calculations after the submission, a few mistakes were spotted, which are further analyzed and discussed here.

One major error is that the standard state correction was missed in our submission. The QM level gas phase calculations are done at standard state of gas while the aqueous phase species are at 1M concentration. This standard state correction needs to be applied while calculation of the overall free energy difference. This contribution is equal to -1.89 kcal/mol, i.e. 1.4 pKa units.

Another source of error comes from the inconsistency with GAFF protocol. Standard AMBER and GAFF force fields scale the electrostatic interaction between third-neighbors (1-4 interactions) by 0.833, while CHARMM force fields on the other hand do not scale the electrostatic 1-4 interactions. In the CHARMM program, an option e14fac (electrostatic 1-4 interaction scaling factor) should be set to 0.833 to use GAFF force fields, however its default value of 1.0 was used in our simulations by mistake. Furthermore, the CHARMM modified TIP3P parameter were used for water molecules which place a small value on the water hydrogen atom. These deviations to the standard GAFF practice render the force field parameters used in this work less optimal.

Other methods to generate more CHARMM-like force field parameters for the microstates have been attempted. The Paramchem server[58], which generates CGENFF force field parameters, was found to report error messages when parametrizing several charged species. The ffTK (force field ToolKit)[59], which is a plugin in VMD that generates CHARMM parameters, was found to be difficult in automatically generating parameters for all the microstates. Since we needed a method that could parameterize all the microstates in a high throughput fashion, we instead opted for using for Antechamber from AmberTools package.

From the absolute error analysis in Table 3.2 we can assume that SM15 parameters are not optimal as the errors for both pKa are very high for this molecule. Force field parameterization for small molecules is indeed difficult due to the very large chemical space of these molecules as compared to the amino acids[60]. The latter have seen several decades of work for a very limited number of species. The general strategy of optimization of parameters of molecules involves the use experimental hydration free energy data[61].

Optimization with this parameter would also be helpful as we indeed need to predict the solvation free energy difference. However, many of microstates of these molecules are charged species and getting high accuracy experimental hydration free energy data would be difficult. Even Self-Consistent Reaction Field (SCRF) based implicit solvent model (SMD) calculations have one order of magnitude higher error as compared to neutral species[23, 62]. One way to study the SM15 errors would be to generate parameters with a different force field and compare their relative performance. While Antechamber generates GAFF-based parameters, ffTK can be used to generate CHARMM300 based parameters.

Our simulation runs also suffered from inadequate sampling of the phase space in the aqueous phase simulation. For the calculation of hydration free energy in SAMPL4 competition with similar system sizes, Gilson et. al.[28] had simulated each λ point for 5 ns. Konig et. al.[29] for the same set of molecules had used a 0.5-1 ns simulation for each λ state in aqueous phase. In principle, much less sampling time would be required in our FEP calculations as relative free energies instead of absolute solvation free energies were being computed. However, the MD simulation time used in this study was still too short (10 ps per λ state), not allowing full water reorganization upon solute deprotonation. The number of simulations that we were performing was much larger (~650 in SAMPL6 vs 24 in SAMPL4) and hence we performed only 0.12 ns simulations for each acid-base pair. Achieving proper sampling is an area of active research in the molecular dynamics field. Indeed, one of the competitions in the SAMPL6 challenge focused on benchmarking this quantity especially in a blind setup. The results from that study would be able to set community-wide guidelines for benchmarking. A heuristic that we should have used to reduce the number of microstate pairs should have been to exclude all microstates that had charges more than 1 or less than -1 i.e. consider only neutral and singly-charged microstates. Some of the other submissions, have used this strategy to limit the number of microstate pairs that needs to be considered without loss in accuracy.

The FEP scheme we used for alchemical transformation included only the transformation of charges on all atoms from the protonated acid to its deprotonated conjugate base. This was similar in principle to the strategy used by Juyong et. al. in their enveloping distribution sampling (EDS) based constant-pH simulations[63], where each state differed from the reference state in only the charges on the residue of deprotonation. The changes in the parameters for VdW interactions as

well as the internal degrees of freedom during the solute deprotonation process will also contribute to the free energy difference, which is not captured in our FEP calculations. We note that it's feasible to include these effects by interpolating all force field parameters, although the bonded interactions might need to be carefully handled[64].

Another possible source of error comes from the value of $\Delta G*(H+)$. Solvation free energy of proton is a contentious value and a range of values from -259 to -264 kcal/mol are available in the literature. This can lead to large errors in the absolute prediction of pKa as just a difference of 1.36 kcal/mol is equivalent to 1 pKa unit. One way to handle this error is to use isodesmic reactions with another acid with known experimental pKa and couple two thermodynamic cycles together such that the solvation free energy of proton cancels out. The second acid chosen should also be similar to the original acid that we are interested in. Essentially, the pKa shift is calculated with respect to a simpler model compound with known experimental pKa values, as being done in most constant pH simulation methods[65, 66, 63]. Our approach instead aims at predicting the absolute pKa, and a fixed value of -264.5 kcal/mol is used for $\Delta G*(H+)$ as derived from cluster-ion solvation data by Tissandier et al[43]. An alternative way to handle this issue, as well as other systematic errors in absolute pKa calculations, is to perform a linear free energy regression against molecules with known experimental pKa, i.e., to consider $\Delta G*(H+)$ as a variable whose value is fitted to best reproduce a set of known pKa values. The empirical correction has been shown to improve the results although the slope of the regression still remains a debatable issue[12]. We have also used the assumption that only one microscopic pKa contributes to the macroscopic pKa if the former is fairly well separated. However, this is an approximation as for a given charge transition, multiple protonated-deprotonated pairs of microstates contribute to the macroscopic pKa[41].

In our approach the is computed using QM calculations at the M06-2X level using 6-31G* basis set (6-31+G* for microstate pairs involving anionic species). Higher level of ab initio methods, larger basis set, and including counterpoise correction should improve our results. Although our method allows the sampling of the phase space during the calculation of the solvation free energy difference, only one conformation (the energy minimized one) is considered for the calculation of by QM in the gas phase. This is again an approximation as previous work by Bochevarov et.al.[11] have shown that multiple low-lying conformations do contribute to the deprotonation free energy. There can be a couple of different strategies to handle this phenomenon. Multiple low-lying conformations can be sampled and the deprotonation energy of each important conformation can be calculated separately and combined together in a Boltzmann weighted sum. Another solution for this problem is to use reweighting as used by Tao et.al.[67]. Free energy of constraining the geometry to the ones used the calculation of gas-phase QM step, can be calculated separately and will have to be added for the protonated microstate and subtracted for the deprotonated microstate.

One of the key physics behind the free energy of deprotonation and hence pKa is the water reorganization when the solute is protonated or deprotonated, which involves water response to the sudden changes of charge distributions. In this case, polarizable force fields should in principle provide higher accuracy in our approach as fixed charge force-fields are limited in their ability to account for the change in charges during the course of the simulation. A theoretically promising method to handle this effect is to use polarizable force fields such as AMOEBA[68, 69], Drude[70] or a recently formulated multipole and induced dipole (MPID) model[71].

Any of these polarizable models should improve the pKa prediction results of our method, given high quality polarizable force field parameters for general drug-like molecules are available.

5 Conclusion

This work reports our submission for the SAMPL6 pKa prediction challenge, where we have attempted to calculate pKa of small drug-like molecules in explicit solvent using a hybrid QM and MM approach. While including multiple solvation shells is difficult in pure ab initio (QM) methods, modeling the dissociation of a proton is difficult at the MM level using conventional force fields. The novel contribution of this work is devising a method to allow the calculation of ΔG in explicit solvent while limiting the cost of the calculations. This is important for a high throughput prediction where a large number of microstates need to be considered.

However, traditional limitations in molecular dynamics simulation approaches limits its competitiveness as compared to a machine learning approach or a full-quantum level implicit solvent approach. At the same time, we committed a few avoidable mistakes in carrying out the simulations. Due to these results from the present version of our method did not do very well in the SAMPL6 pKa challenge. More work needs to be done to optimize and automate the protocols.

We are currently working on improving the method. We need to improve force field parameters for the small molecules, ensure proper sampling of the intermediate lambda points during free energy calculations and utilize a higher level of theory for the gas phase QM calculations. Our new version of the method is an open source tool where we can use test the method easily for each of

these factors. It will allow the method to be used for not just pKa calculation of small molecules but for larger proteins of interest as well. The open source tool, currently in development, is available at https://github.com/samarjeet/hpka.

6 Acknowledgements

The work is supported by the Intramural Research Program of the National Heart, Lung and Blood Institute Z01 HL001051. The authors would like to acknowledge Xiongwu Wu, Kyungreem Han, Philip Hudson, Michael Jones, Ana Damjanovic, Gerhard Konig, Frank Pickard, Florentina Tofoleanu, Reuben Meanapa for helpful discussion. This work utilized the computational resources of the NIH HPC Biowulf cluster. http://hpc. nih.gov and the Laboratory of Computational Biology cluster. SP would like to acknowledge Biochemistry, Cellular and Molecular Biology (BCMB) graduate program at JHMI.

7 References

1. James T. Muckerman, Jonathan H. Skone, Ming Ning, and Yuko Wasada Tsutsui. Toward the accurate calculation of pka values in water and acetonitrile. Biochimica et Biophysica Acta (BBA) - Bioenergetics, 1827(8-9):882891, 2013. doi: 10.1016/j.bbabio.2013.03.011.

2. Paul G. Seybold and George C. Shields. Computational estimation of pka values. Wiley Interdisciplinary Reviews: Computational Molecular Science, 5(3):290297, 2015. doi: 10.1002/wcms.1218.

3. Yulan Wang, Jing Xing, Yuan Xu, Nannan Zhou, Jianlong Peng, Zhaoping Xiong, Xian Liu, Xiaomin Luo, Cheng Luo, Kaixian Chen, and et al. In silico adme/t modelling for rational drug design. Quarterly Reviews of Biophysics, 48(4):488515, 2015. doi: 10.1017/S0033583515000190.

4. Eric Hajjar, Annick Dejaegere, and Nathalie Reuter. Challenges in pkapredictions for proteins: The case of asp213 in human proteinase 3. The Journal of Physical Chemistry A, 113(43):1178311792, 2009. Doi: 10.1021/jp902930u.

5. Adam C. Lee and Gordon M. Crippen. Predicting pka. Journal of Chemical Information and Modeling, 49(9):2013–2033, 2009. doi: 10.1021/ci900209w. URL https://doi.org/10.1021/ci900209w. PMID:19702243.

6. Yu. E. Zevatskii and D. V. Samoilov. Modern methods for estimation of ionization constants of organic compounds in solution. Russian Journal of Organic Chemistry, 47(10):14451467, 2011. doi: 10.1134/ s1070428011100010.

7. Jeremy R. Greenwood, David Calkins, Arron P. Sullivan, and John C. Shelley. Towards the comprehensive, rapid, and accurate prediction of the favorable tautomeric states of drug-like molecules in aqueous solution. Journal of Computer-Aided Molecular Design, 24(6-7):591604, 2010. doi: 10.1007/s10822-010-9349-1.

8. Robert Fraczkiewicz, Mario Lobell, Andreas H. Gller, Ursula Krenz, Rolf Schoenneis, Robert D. Clark, and Alexander Hillisch. Best of both worlds: Combining pharma data and state of the art modeling technology to improve in silico pka prediction. Journal of Chemical Information and Modeling, 55(2):389397, 2014. doi: 10.1021/ci500585w.

9. John C. Shelley, Anuradha Cholleti, Leah L. Frye, Jeremy R. Greenwood, Mathew R. Timlin, and Makoto Uchimaya. Epik: a software program for pk a prediction and protonation state generation for drug-like molecules. Journal of Computer-Aided Molecular Design, 21(12):681691, 2007. doi: 10.1007/s10822-007-9133-z.

10. Mengshan Li, Huaijing Zhang, Bingsheng Chen, Yan Wu, and Lixin Guan. Prediction of pka values for neutral and basic drugs based on hybrid artificial intelligence methods. Scientific Reports, 8(1), May 2018. doi: 10.1038/s41598-018-22332-7.

11. Art D. Bochevarov, Mark A. Watson, Jeremy R. Greenwood, and Dean M. Philipp. Multiconformation, density functional theory-based pka prediction in application to large, flexible organic molecules with diverse functional groups. Journal of Chemical Theory and Computation, 12(12): 466 60016019, 2016. doi: 10.1021/acs.jctc.6b00805.

12. Andreas Klamt, Frank Eckert, Michael Diedenhofen, and Michael E. Beck. First principles calculations of aqueous pka values for organic and inorganic acids using cosmors reveal an inconsistency in the slope of the pka scale. The Journal of Physical Chemistry A, 107(44):93809386, 2003. doi: 10. 471 1021/jp034688o.

13. Jasna J. Klici, Richard A. Friesner, Shi-Yi Liu, and Wayne C. Guida. Accurate prediction of acidity constants in aqueous solution via density functional theory and self-consistent reaction field methods. The Journal of Physical Chemistry A, 106(7):1327–1335, 2002. doi: 10.1021/jp012533f. URL https://doi.org/10.1021/jp012533f.

14. Bishnu Thapa and H. Bernhard Schlegel. Improved pka prediction of substituted alcohols, phenols, and hydroperoxides in aqueous medium using density functional theory and a cluster-continuum solvation model. The Journal of Physical Chemistry A, 121(24):46984706, Aug 2017. doi: 10.1021/acs.jpca.7b03907.

15. Junming Ho. Are thermodynamic cycles necessary for continuum solvent calculation of pkas and reduction potentials? Physical Chemistry Chemical Physics, 17(4):28592868, 2015. doi: 10.1039/c4cp04538f.

16. Peng Lian, Ryne C. Johnston, Jerry M. Parks, and Jeremy C. Smith. Quantum chemical calculation of pkas of environmentally relevant functional groups: Carboxylic acids, amines, and thiols in aqueous solution. The Journal of Physical Chemistry A, 122(17):43664374, Oct 2018. doi: 10.1021/acs.jpca.8b01751.

17. Amanda G. Riojas and Angela K. Wilson. Solv-ccca: Implicit solvation and the correlation consistent composite approach for the determination of pka. Journal of Chemical Theory and Computation, 10(4):15001510, Dec 2014. doi: 10.1021/ct400908z.

18. Matthew D. Liptak and George C. Shields. Accurate pka calculations for carboxylic acids using complete basis set and gaussian-n models combined with cpcm continuum solvation methods. Journal of the American Chemical Society, 123(30):7314–7319, 2001. doi: 10.1021/ja010534f. URL https://doi.org/10.1021/ja010534f. PMID: 11472159.

19. Matthew D. Liptak and George C. Shields. Experimentation with different thermodynamic cycles used for pka calculations on carboxylic acids using complete basis set and gaussian-n models combined with cpcm continuum solvation methods. International Journal of Quantum Chemistry, 85(6):727741, 2001. doi: 10.1002/qua.1703.

20. Benjamin G. Tehan, Edward J. Lloyd, Margaret G. Wong, Will R. Pitt, John G. Montana, David T. Manallack, and Emanuela Gancia. Estimation of pka using semiempirical molecular orbital methods. part 1: Application 507 to phenols and carboxylic acids. Quantitative Structure-Activity Relationships, 21(5):457472, 2002. doi: 10.1002/1521-3838(200211)21:5h457::aid-qsar457i3.0.co;2-5.

21. R. Peverati and D. G. Truhlar. Quest for a universal density functional: the accuracy of density functionals across a broad spectrum of databases in chemistry and physics. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 372(2011):2012047620120476, Oct 2014. doi: 10.1098/rsta.2012.0476.

22. A. Klamt and G. Schrmann. Cosmo: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. J. Chem. Soc., Perkin Trans. 2, (5):799805, 1993. doi: 10.1039/p29930000799.

23. Aleksandr V. Marenich, Christopher J. Cramer, and Donald G. Truhlar. Universal solvation model based on solute electron density and on a continuum model of the solvent defined by the bulk dielectric constant and atomic surface tensions. The Journal of Physical Chemistry B, 113(18):63786396, Jul 2009. doi: 10.1021/jp810292n.

24. Junming Ho and Mehmed Z. Ertem. Calculating free energy changes in continuum solvation models. The Journal of Physical Chemistry B, 120(7):13191329, 2016. doi: 10.1021/acs.jpcb.6b00164.

25. Vincenzo Barone and Maurizio Cossi. Quantum calculation of molecular energies and energy gradients in solution by a conductor solvent model. The Journal of Physical Chemistry A, 102(11):19952001, 1998. doi: 10.1021/jp9716997.

26. Junming Ho. Predicting pka in implicit solvents: Current status and future directions. Australian Journal of Chemistry, 67(10):1441, 2014. doi: 10.1071/ch14040.

27. Rodrigo Casasnovas, Joaquin Ortega-Castro, Juan Frau, Josefa Donoso, and Francisco Muoz. Theoretical pka calculations with continuum model solvents, alternative protocols to

thermodynamic cycles. International Journal of Quantum Chemistry, 114(20):13501363, Dec 2014. doi: 10.1002/qua.24699.

28. Hari S. Muddana, Neil V. Sapra, Andrew T. Fenley, and Michael K. Gilson. The sampl4 hydration challenge: evaluation of partial charge sets with explicit-water molecular dynamics simulations. Journal of Computer-Aided Molecular Design, 28(3):277287, 2014. doi: 10.1007/s10822-014-9714-6.

29. Gerhard K¨onig, Frank C. Pickard, Ye Mei, and Bernard R. Brooks. Predicting hydration free energies with a hybrid qm/mm approach: an evaluation of implicit and explicit solvation models in sampl4. Journal of Computer-Aided Molecular Design, 28(3):245257, Jul 2014. doi: 10.1007/s10822-014-9708-4.

30. David L. Mobley, Christopher I. Bayly, Matthew D. Cooper, Michael R. Shirts, and Ken A. Dill. Correction to small molecule hydration free energies in explicit solvent: An extensive test of fixed-charge atomistic simulations. Journal of Chemical Theory and Computation, 11(3):13471347, 2015. doi: 10.1021/acs.jctc.5b00154.

31. Michael R. Shirts, Jed W. Pitera, William C. Swope, and Vijay S. Pande. Extremely precise free energy calculations of amino acid side chain analogs: Comparison of common molecular mechanics force fields for proteins. The Journal of Chemical Physics, 119(11):57405761, 2003. doi: 10.1063/1.1587119.

32. J. Peter Guthrie. A blind challenge for computational solvation free energies: Introduction and overview. The Journal of Physical Chemistry B, 113(14):4501–4507, 2009. doi: 10.1021/jp806724u. URL https://doi.org/10.1021/jp806724u. PMID: 19338360.

33. Hari S. Muddana, Andrew T. Fenley, David L. Mobley, and Michael K. Gilson. The sampl4 host guest blind prediction challenge: an overview. Journal of Computer-Aided Molecular Design, 28(4):305317, 2014. doi: 10.1007/s10822-014-9735-1.

34. Frank C. Pickard, Gerhard Knig, Florentina Tofoleanu, Juyong Lee, Andrew C. Simmonett, Yihan Shao, Jay W. Ponder, and Bernard R. Brooks. Blind prediction of distribution in the sampl5 challenge with qm based protomer and pka corrections. Journal of Computer-Aided Molecular Design, 30(11):10871100, 2016. doi: 10.1007/s10822-016-9955-7.

35. Jian Yin, Niel M. Henriksen, David R. Slochower, Michael R. Shirts, Michael W. Chiu, David L. Mobley, and Michael K. Gilson. Overview of the sampl5 hostguest challenge: Are we doing better? Journal of Computer-Aided Molecular Design, 31(1):119, 2016. doi: 10.1007/s10822-016-9974-4.

36. Matthew T. Geballe and J. Peter Guthrie. The sampl3 blind prediction challenge: transfer energy overview. Journal of Computer-Aided Molecular Design, 26(5):489496, Mar 2012. doi: 10.1007/s10822-012-9568-8.

37. Arin S. Rustenburg, Justin Dancer, Baiwei Lin, Jianwen A. Feng, Daniel F. Ortwine, David L. Mobley, and John D. Chodera. Measuring experimental cyclohexane-water distribution coefficients for the sampl5 challenge. Journal of Computer-Aided Molecular Design, 30(11):945958, Jul 2016. doi: 10.1007/s10822-016-9971-7.

38. Mehtap Isik et al. pka measurements for the sampl6 prediction challenge for a set of kinase inhibitor-like fragments. Journal of Computer-Aided Molecular Design, submitted.

39. Zolt´an Szak´acs and B´ela Nosz´al. Protonation microequilibrium treatment of polybasic compounds with any possible symmetry. Journal of Mathematical Chemistry, 26(1):139, Oct 1999.

40. Dean M. Philipp, Mark A. Watson, Haoyu S. Yu, Thomas B. Steinbrecher, and Art D. Bochevarov. Quantum chemical prediction for complex organic molecules. International Journal of Quantum Chemistry, 118(12):e25561. doi: 10.1002/qua.25561. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.25561.

41. Ivan G. Darvey. The assignment of pka values to functional groups in amino acids. Biochemical Education, 23(2):8082, 1995. doi: 10.1016/ 598 0307-4412(94)00150-n.

42. Donald A. McQuarrie. Statistical Mechanics. University Science Books, 2000.

43. Michael D. Tissandier, Kenneth A. Cowen, Wan Yong Feng, Ellen Gundlach, Michael H. Cohen, Alan D. Earhart, James V. Coe, and Thomas R. Tuttle. The proton's absolute aqueous enthalpy and gibbs free energy of solvation from cluster-ion solvation data. The Journal of Physical Chemistry A, 102(40):7787–7794, 1998. doi: 10.1021/jp982638r. URL https://doi.org/10.1021/jp982638r.

44. William L. Jorgensen and C. Ravimohan. Monte carlo simulation of differences in free energies of hydration. The Journal of Chemical Physics, 83(6):30503054, 1985. doi: 10.1063/1.449208.

45. M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian16 Revision B.01, 2016. Gaussian Inc. Wallingford CT.

46. Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. 628 Journal of Computational Chemistry, 4(2):187–217, 1983. doi: 10.1002/jcc.540040211. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540040211.

47. B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, and et al. Charmm: The biomolecular simulation program. Journal of Computational Chem634 istry, 30(10):15451614, 2009. doi: 10.1002/jcc.21287.

48. Eric Anderson, Gilman D. Veith, and David Weininger. SMILES, a line notation and computerized interpreter for chemical structures. U.S. Environmental Protection Agency, Environmental Research Laboratory, 1987.

49. Paolo Mazzatorta, Lien-Anh Tran, Benoit Schilter, and Martin Grigorov. Integration of structure activity relationship and artificial intelligence systems to improve in silico prediction of ames test mutagenicity. ChemInform, 38(15), Oct 2007. doi: 10.1002/chin.200715211.

50. Yan Zhao and Donald G. Truhlar. The m06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four m06-class functionals and 12 other functionals. Theoretical Chemistry Accounts, 120(1-3):215241, Dec 2007. doi: 10.1007/s00214-007-0310-x.

51. A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wirkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. The Journal of Physical Chemistry B, 102(18):3586–3616, 1998. doi: 10.1021/jp973084f. URL https://doi.org/10.1021/jp973084f. PMID: 24889800.

52. Araz Jakalian, David B. Jack, and Christopher I. Bayly. Fast, efficient generation of high-quality atomic charges. am1-bcc model: Ii. parame terization and validation. Journal of Computational Chemistry, 23(16):16231641, 2002. doi: 10.1002/jcc.10128.

53. Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. Journal of Computational Chemistry, 25(9):11571174, 2004. doi:10.1002/jcc.20035.

54. D. J. Evans and B. L. Holian. The nosehoover thermostat. The Journal of Chemical Physics, 83(8):4069–4074, 1985. doi: 10.1063/1.449071. URL https://doi.org/10.1063/1.449071.

55. Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An nlog(n) method for ewald sums in large systems. The Journal of Chemical Physics, 98(12):1008910092, 1993. doi: 10.1063/1.464397.

56. Ulrich Essmann, Lalith Perera, Max L. Berkowitz, Tom Darden, Hsing Lee, and Lee G. Pedersen. A smooth particle mesh ewald method. The Journal of Chemical Physics, 103(19):8577–8593, 1995. doi: 10.1063/1.470117. URL https://doi.org/10.1063/1.470117.

57. H. W. Kuhn. The hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(12):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

58. K. Vanommeslaeghe and A. D. Mackerell. Automation of the charmm general force field (cgenff) i: Bond perception and atom typing. Journal of Chemical Information and Modeling, 52(12):31443154, 2012. doi: 10.1021/ci300363c.

59. Christopher G. Mayne, James C. Gumbart, and Emad Tajkhorshid. The force field toolkit: Software for the parameterization of small molecules from first principles. Biophysical Journal, 104(2), 2013. doi: 10.1016/j.bpj.2012.11.209.

60. Lei Huang and Benot Roux. Automated force field parameterization for nonpolarizable and polarizable atomic models based on ab initio target data. Journal of Chemical Theory and Computation, 9(8):35433556, 2013. doi: 10.1021/ct4003477.

61. Chris Oostenbrink, Alessandra Villa, Alan E. Mark, and Wilfred F. Van Gunsteren. A biomolecular force field based on the free enthalpy of hydration and solvation: The gromos force-

field parameter sets 53a5 and 53a6. Journal of Computational Chemistry, 25(13):16561676, 2004. doi:10.1002/jcc.20090.

62. Elizabeth L. M. Miguel, Calink I. L. Santos, Carlos M. Silva, and Josefredo R. Pliego Jr. How accurate is the smd model for predicting free energy barriers for nucleophilic substitution reactions in polar protic and dipolar aprotic solvents? Journal of the Brazilian Chemical Society, 2016. doi: 700 10.5935/0103-5053.20160095.

63. Juyong Lee, Benjamin T. Miller, and Bernard R. Brooks. Computational scheme for ph-dependent binding free energy calculation with explicit solvent. Protein Science, 25(1):231243, 2015. doi: 10.1002/pro.2755.

64. Gerhard Knig and Bernard R. Brooks. Correcting for the free energy costs of bond or angle constraints in molecular dynamics simulations. Biochimica et Biophysica Acta (BBA) - General Subjects, 1850(5):932–943, 2015. ISSN 0304-4165. doi: https://doi.org/10.1016/j.bbagen.2014.09.001. URL http://www.sciencedirect.com/science/article/pii/ S030441651400302X. Recent developments of molecular dynamics.

65. Jana Khandogin and Charles L. Brooks. Constant ph molecular dynamics with proton tautomerism. 89:141–157, 2005.

66. Serena Donnini, Florian Tegeler, Gerrit Groenhof, and Helmut Grub muller. Constant ph molecular dynamics in explicit solvent with λ dynamics. 7:1962–1978, 2011.

67. Peng Tao, Alexander J. Sodt, Yihan Shao, Gerhard Knig, and Bernard R. Brooks. Computing the free energy along a reaction coordinate using rigid body dynamics. Journal of Chemical Theory and Computation, 10(10):41984207, Sep 2014. doi: 10.1021/ct500342h.

68. Jay W. Ponder, Chuanjie Wu, Pengyu Ren, Vijay S. Pande, John D. Chodera, Michael J. Schnieders, Imran Haque, David L. Mobley, Daniel S. Lambrecht, Robert A. DiStasio, Martin Head-Gordon, Gary N. I. Clark, Margaret E. Johnson, and Teresa Head-Gordon. Current status of the amoeba polarizable force field. The Journal of Physical Chemistry B, 114(8):2549–2564, 2010. doi: 10.1021/jp910674d. URL https://doi.org/10.1021/jp910674d. PMID: 20136072.

69. Richard T. Bradshaw and Jonathan W. Essex. Evaluating parametrization protocols for hydration free energy calculations with the amoeba polarizable force field. Journal of Chemical Theory and Computation, 12(8): 38713883, 2016. doi: 10.1021/acs.jctc.6b00276.

70. Christopher M. Baker, Pedro E. M. Lopes, Xiao Zhu, Roux Benoit, and Alexander D. Mackerell. Accurate calculation of hydration free energies using pair-specific lennard-jones parameters in the charmm drude polarizable force field. Journal of Chemical Theory and Computation, 6(4):11811198, 2010. doi: 10.1021/ct9005773.

71. Jing Huang, Andrew C. Simmonett, Frank C. Pickard, Alexander D. Mackerell, and Bernard R. Brooks. Mapping the drude polarizable force field onto a multipole and induced dipole model. The Journal of Chemical Physics, 147(16):161702, 2017. doi: 10.1063/1.4984113.

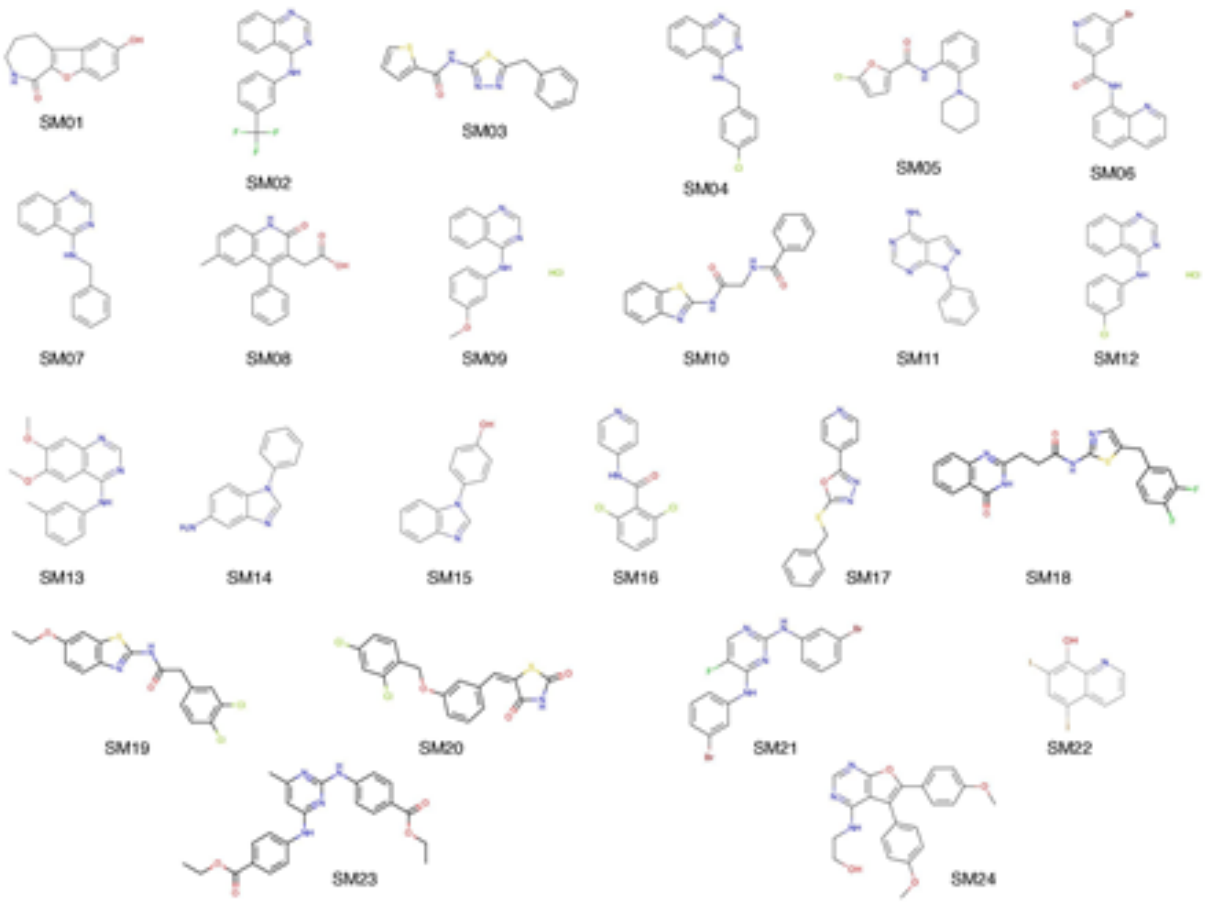


Figure 3.1: Molecules in the SAMPL6 prediction challenge.

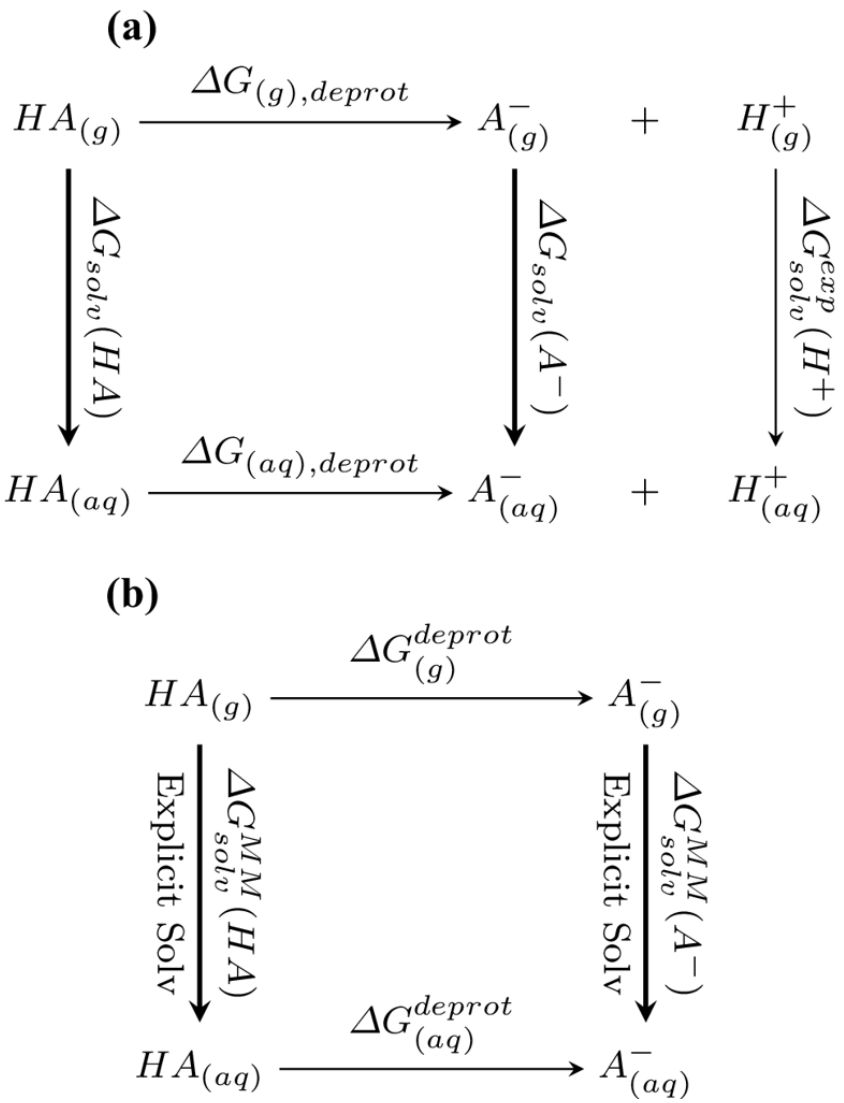


Figure 3.2: Thermodynamic cycles used in the pKa calculations a) chemical reaction of acid dissociation. This relates the free energy of dissociation in the aqueous phase as with the gas phase free energy of dissociation and solvation free energies of the acid, base and proton. b) Alchemical cycle for deprotonation. This cycle relates the solvation free energy difference of the HA and A^- with difference in free energy for deprotonation in the aqueous and gas phases.

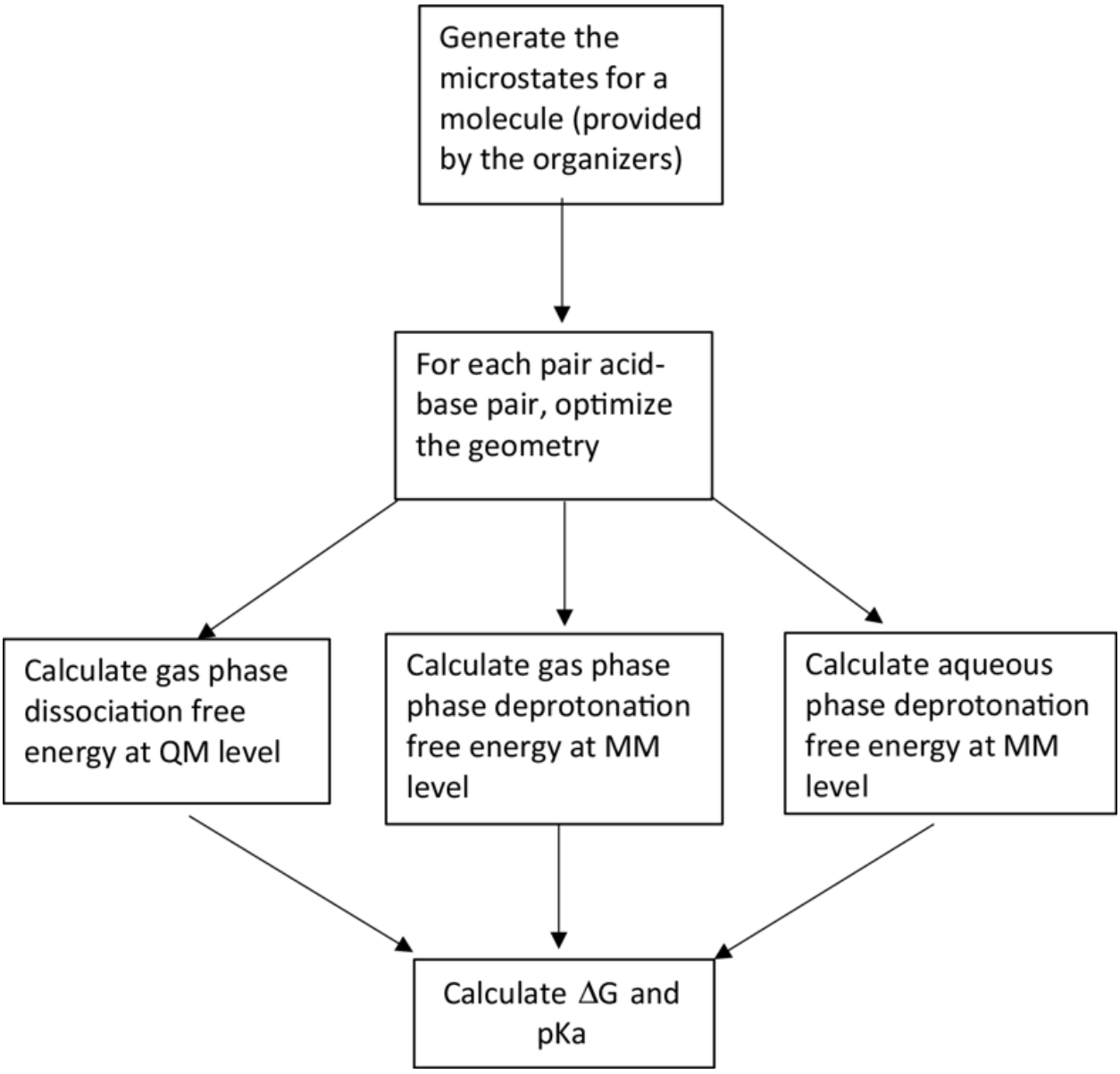


Figure 3.3: Workflow for the hybrid QM and MM pKa prediction approach

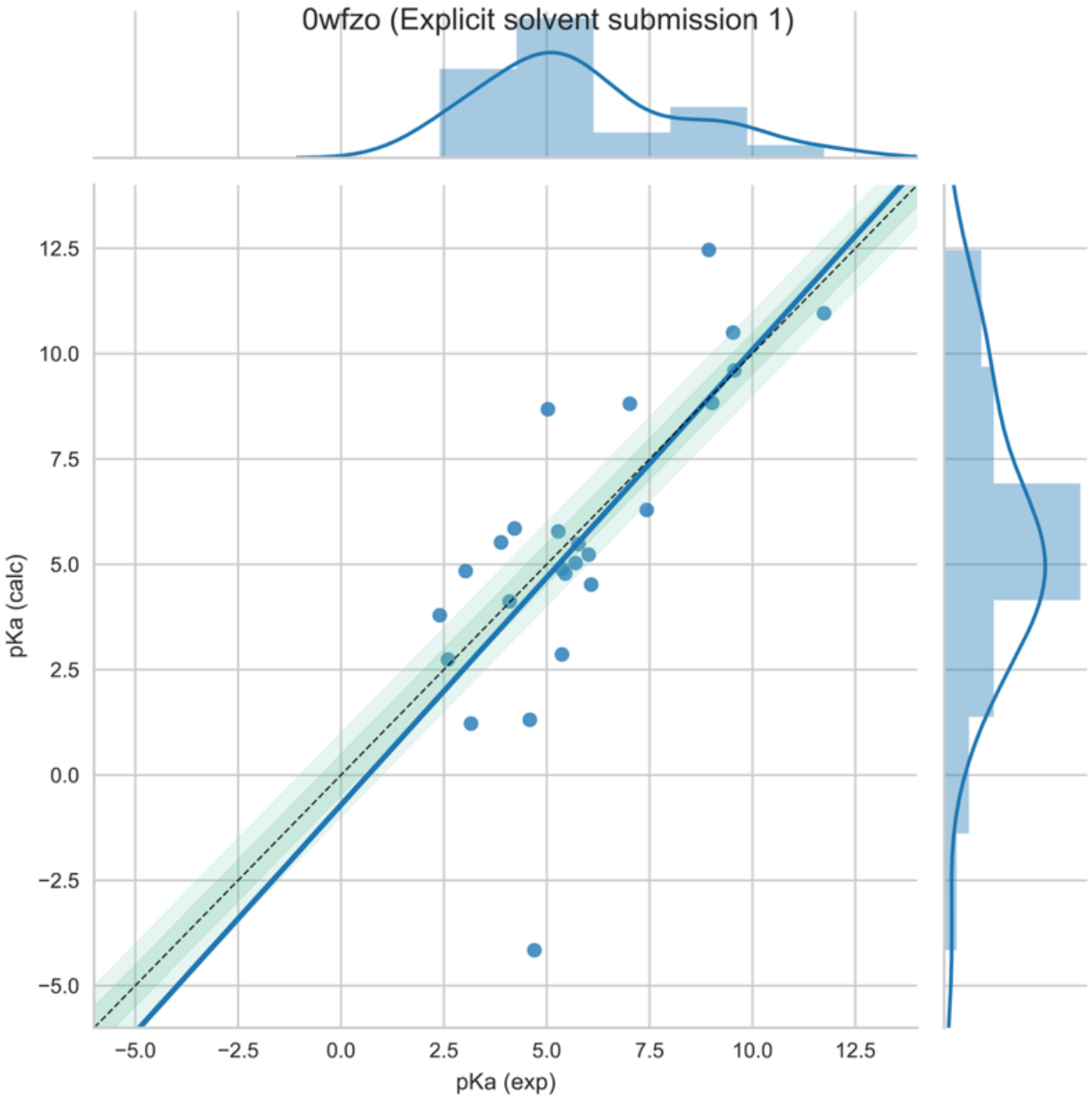


Figure 3.4: Plot of the closest analysis scheme and experimental pKa values. Plot courtesy of the organizers https://github.com/MobleyLab/SAMPL6/blob/master/physical_properties/pKa/analysis/analysis_of_typeI_predictions/analysis_outputs_closest/pKaCorrelationPlots/0wfzo.pdf

Table 1

Statistics of the performance of the method using Hungarian and closest schemes

Evaluation scheme	RMSE	MAE	r^2	m
Hungarian	2.89	1.88	0.48	0.99
Closest	2.42	1.61	0.53	1.08

RMSE root mean square error, *MAE* maximum absolute error, r^2 correlation coefficient of determination, m slope of the linear regression line

Table 3.1

Table 2
Comparison of experimental and calculated values using the closest scheme

Molecule	Exp. value	Calculated value
SM01	9.53 ± 0.01	10.5
SM02	5.03 ± 0.01	8.68
SM03	7.02 ± 0.01	8.81
SM04	6.02 ± 0.01	5.23
SM05	4.59 ± 0.01	1.31
SM06(1)	3.03 ± 0.04	4.84
SM06(2)	11.74 ± 0.01	10.96
SM07	6.08 ± 0.01	4.52
SM08	4.22 ± 0.01	5.85
SM09	5.37 ± 0.01	4.89
SM10	9.02 ± 0.01	8.83
SM11	3.89 ± 0.01	5.52
SM12	5.28 ± 0.01	5.78
SM13	5.77 ± 0.01	5.48
SM15(1)	4.7 ± 0.01	− 4.16
SM15(2)	8.94 ± 0.01	12.46
SM16	5.37 ± 0.01	2.86
SM17	3.16 ± 0.01	1.22
SM19	9.56 ± 0.01	9.6
SM20	5.7 ± 0.03	5.03
SM21	4.1 ± 0.01	4.12
SM22(1)	2.4 ± 0.02	3.79
SM22(2)	7.43 ± 0.01	6.29
SM23	5.45 ± 0.01	4.78
SM24	2.6 ± 0.01	2.74

Table 3.2 Comparision of experimental and calculated values using the closest scheme

Chapter 4

A deep learning approach for the blind logP prediction in SAMPL6 challenge

(A version of this chapter has been submitted to Journal of Computer Aided Drug Design and is expected to appear in the September 2019 special issue.)

Abstract

Water-octanol partition coefficient serves as a measure for the lipophilicity of a molecule and is important in the field of drug discovery. A novel method for computational prediction of logarithm of partition coefficient (logP) has been developed using molecular fingerprints and a deep neural network. The machine learning model was trained on a dataset of more than 12,000 molecules and tested on more than 200 molecules. In this article, we present our results for the blind prediction of logP for the SAMPL6 challenge. While the best submission achieved a RMSE of 0.41 logP units, our submission had a RMSE of 0.61 logP units. Overall, we ranked in the top quarter out of the 92 submissions that were made. Our results show that the deep learning model can be used as a fast, accurate and robust method for high throughput prediction of logP of small molecules.

1 Introduction

Computational prediction of logP values of molecules is important in several fields including drug design, agriculture, environment, consumer-chemicals etc. as it serves as a measure of lipophilicity (or hydrophobicity) of the molecule[1]. In the field of drug design, lipophilicity of a drug molecule is directly related to the absorption and membrane penetration, solubility, partitioning into tissues and the final excretion. It is considered as one of the most important properties of a drug and is a part of the Lipinski's rule of five[2]. A drug molecule has to be soluble enough in lipid to be absorbed

in the tissues and organs. However, it should not be too soluble to prohibit its excretion [3]. In agriculture and environment science, it is related to the toxicity of the fertilizers and pesticides used. In the field of cosmetics and skin care products, it measures the propensity of the product being absorbed by the skin.

LogP is related to the partition coefficient of a molecule between water and the lipid phase. Typically, the lipid phase is n-octanol and logP is given by:

$$logP = \log(\frac{[solute]oct}{[solute]water})$$

Given the importance of logP prediction, the Drug Design and Data Resources (D3R) consortium organized the sixth iteration of Statistical Assessment of the Modeling of Proteins and Ligands (SAMPL) competition to compare different methods in this field. Previously, SAMPL competitions involved solvation free energy[4], logD[5] and pKa[6] prediction. Specifically, in the first iteration of the SAMPL6 competition, pKa prediction challenge involved charged microstates of a set of 24 drug-like molecules. A subset of these molecules, where the neutral species was the most abundant microstate, were used in part II of the SAMPL6 competition for logP prediction.

Many different approaches were employed by a large number of groups. These can be seen in the other submissions to the SAMPL6 competition available in this special issue. These methods span both physical and empirical approaches. The quantum level approaches calculate the solvation free energy of the molecule separately in two solvent phases - water and octanol and use it to estimate the partition coefficient. Specifically, Andreas Klamt's group at COSMOlogic [to be

published in the same edition] generated relevant conformations of the molecule and for each conformation calculated the chemical potential difference between the two phases. Among the empirical approaches, Vzgyt used QSAR model based on molecular descriptor based on properties and trained a random forest model to predict pKa values [to be published in the same edition].

In this work, we have developed a novel deep learning model for the computational prediction of logP values. We have used fingerprinting to generate features for the molecule. A large database of more than 14,000 molecules was used to train and test the model. Our goal for the project was to develop a model which can utilize this large database and generalize over a large test set. Deep neural networks are an excellent choice for this training. They have recently been used in a number of fields including QSAR studies for IC_{50} prediction[7]. We explored the usage of deep neural networks with two different models with five and three hidden layers respectively.

The paper is organized as follows. In Section 2, we describe the computational details of the method, including the description of the data set used and the architecture of the neural networks that were designed. Section 3 covers our major results, comparison to other methods and a discussion on prospects for further improvement on the work. Finally, in Section 4, a brief conclusion of the study is provided.

2 Computational Details

We carried out this study to perform a blind prediction in the SAMPL6 logP challenge. A schematic representation of the approach is given in Figure 4.1. SAMPL6 organizers provided a

set of 11 drug-like molecules in the SMILES string format. In order to use a machine learning approach for prediction, we first made a vector-space representation of the molecule. We trained a number of deep neural network models on a previous dataset of logP predictions. The models which provided the least error in our validation and test sets were used further to make the final prediction for the challenge molecules. In the next subsections, we provide a detailed description of the approach used including the vector space representation, training and testing of our models.

2.1 SAMPL6 logP prediction challenge molecules

The logP prediction challenge consisted of making blind prediction of the octanol-water partition coefficients of 11 small molecules that are similar to small molecule protein kinase inhibitors. Figure 4.2 shows the 2-dimensional structure of the molecules. An ASCII formatted notation for the molecules, named Simplified Molecular-Input Line-Entry System (SMILES), is used for the initial representation of the molecule. SMILES string provides a unique way for naming a molecule. Atoms are represented by their symbol with the option of including the charge if any. Bonds are represented by symbols: single (-), double (=), triple (#) and aromatic (:). Branches are depicted with brackets. Cycles are broken at one bond and labels are attached on the atoms in the broken bond. More details about the representation scheme for SMILES can be obtained from the Daylight manual for SMILES.

2.2 Extended-Connectivity Fingerprinting

We used the approach formulated by Rogers and Hahn[8] to make a vector space representation of the molecule. This method, termed Extended-Connectivity Finger Printing (ECFP), has been used extensively in the QSAR field for structure-function modeling. It models the atoms and its bonded neighborhood iteratively at longer bond distances.

In the first iteration, seven features of each non-hydrogen atom in the molecule are calculated:

1. number of heavy neighbors
2. valence of the atom subtracted by the number of hydrogen atoms attached
3. atomic number
4. Atomic charge
5. Atomic mass
6. Number of attached hydrogens
7. Whether the atom is contained in a ring

These identifiers are hashed into a 32-bit integer. At the end of the first stage of iteration, we have an array of 32-bit integers, one for each heavy atom.

In the next set of iterations, we try to model the bonded environment of each atom. The identifier array is appended by a tuple for each bond. The first entry for the tuple is bond order and the second entry are the identifier calculated at the first stage. The full array for each atom is again

hashed to create a new 32-bit integer. Thus, at the end of this iteration, we have an array of integers - one for each atom and one for each bond centered at the respective atom. Any duplicate entries, if present, are removed from the array.

The same process is repeated for another iteration to get the identifiers for atoms separated by 2 bonds. These iterations can be seen as adding features representing atom-centered substructures of larger radii. In this study we have used ECFPs up to the fourth order. The array of integers at the last stage is hashed to create a 1024-bit vector. This vector serves as the final vector space representation for a molecule. We used rdkit python library to create the fingerprint for each molecule.

2.3 Training and testing data set

Training data was obtained from PHYSPROP database (www.srcinc.com). It contains a set of 14,176 data points with SMILES string as the molecule and the corresponding logP values. Of these, a randomly selected 12,000 data points were used as the training set and the rest as the training set.

2.4 Architecture of neural network

We trained a number of different models which varied in the size and number of hidden layers in their architecture. Here, we report of two of those architectures that we submitted in the SAMPL6 challenge. The first neural network has 5 hidden layers: 3 layers of 512 units and 2 layers of 256

units. The second neural network is much simpler and has 3 hidden layers with 512, 256 and 128 units in the hidden layers. All the models have one output layer. A total of 150 epochs of training was done on the dataset with 5-fold cross validation within the training data.

3 Results and Discussion

Our results show very good agreement with the experimental data. Results are presented in Table 4.2. With the 5 hidden layers model, we obtained a root mean squared error (RMSE) of 0.62 logP units while the Mean absolute error (MAE) was 0.51 logP units. Correlation with the experimental data was 0.66 and the slope of the regression line was 1.21. With the 3 hidden layers model, we obtained a RMSE of 0.85 logP units. This corresponded with MAE of 0.72 logP units. Correlation coefficient for this model was 0.52 and slope of the regression line was 1.18.

As expected, our 5 hidden layers model performs better than the one with 3 hidden layers. As seen in table 4.3, the number of tunable parameters in this model is more than 1.2 million. This model is able to approximate an arbitrary function much better than a model with lesser number of parameters. Since we have a large dataset of more than 12k training set representing a wide variety of chemical moieties (substructures), the bias in the model is expected to be low.

We also tested the model of a larger dataset of 2000 molecules and the results are shown in Figure 4.6. MAE in this set was 0.68 logP units. This shows that the model is robust over a larger number of molecules with a variety of substructures and is expected to perform well in other studies as well with non-kinase targeting drug-like fragments.

Radius of ECPC affects the feature space representation of the molecule. A larger radius creates identifiers which correspond to bigger interaction regions in the molecule. This also leads to a dramatic increase in the size of the features space and would give a sparser distribution over the bits of the feature space. As noted by Liu et.al., ECPC_2 serve as a good compromise for feature representation and performs well for database searching and QSPR studies.

One limitation of informatics-based approach for property prediction should be realized. The machine learning model learns the distribution of the data that it is used for its training. If the test data is drawn from a different distribution, the model is not expected to be robust enough to make the correct predictions. In other words, machine learning models are good at interpolating within the distribution but not reliable for extrapolation. In terms of logP prediction, if the test molecules contain substructures that are absent in the training data, the trained model will give high errors. However, the SAMPL6 competition involved prediction for kinase-fragment molecules which are very well represented in our training set. Hence the errors in our predictions are within 1.5 logP units for all the molecules.

A clear advantage of the present approach is the speed of calculation. Although collection of training data and training the model can take appreciable time, inference is very fast. In our tests performed on a Volta Nvidia GPU, each molecule takes less than a second for prediction. This makes the approach amenable to deployment for high-throughput prediction in the industrial setup where a large number of molecules need to be tested. Physical approaches, based on QM and/or MM approaches take several hours in contrast for prediction for one molecule.

There are several avenues to build up on the work presented in this article. One criticism of the machine learning approaches is the requirement of large amount of data needed to train a model. Small sample size gives high bias and the model is not expected to perform well. However, large training size might not be available for different physical properties. Transfer learning can be used to handle this issue. For example, a related problem to transfer free energy prediction between water and octanol is that of prediction of transfer free energy prediction between water and cyclohexane. The architecture of the present model, trained on a large water-octanol logP, can be modified at the outer layer to make a prediction for water-cyclohexane logP and training it further on a smaller set of data for the second property. These approaches have been used in the field of computer vision.

Our results show that deep neural networks can be used to predict logP values. The features space representation is easy to build and the model trains very fast on the modern GPUs even with a large number of tunable parameters. Results on over 2000 molecules show that the model is robust over a large variety of substructures.

4 Conclusion

We have developed a novel method for prediction of logP for the SAMPL6 physical properties challenge organized by Drug, Design, Data and Research Consortium. The method uses structure-based fingerprints to represent a molecule in a vector space. Several deep neural architectures were trained on a dataset of ~14,000 known logP values. The submitted models gave excellent results

on a blind set of 11 kinase-inhibitors drug like molecules. The method is fast, accurate and robust over a variety of molecules.

5 References

(1) Plante, J.; Werner, S. JPlogP: An Improved LogP Predictor Trained Using Predicted Data. *J. Cheminform.* **2018**, *10* (1), 61.

(2) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Deliv. Rev.* **2001**, *46* (1–3), 3–26.

(3) Wang, R.; Ozhgibesov, M.; Hirao, H. Analytical Hessian Fitting Schemes for Efficient Determination of Force-Constant Parameters in Molecular Mechanics. *J. Comput. Chem.* **2018**, *39* (6), 307–318.

(4) König, G.; Pickard, F. C.; @bullet, I.; Mei, Y.; Brooks, B. R. Predicting Hydration Free Energies with a Hybrid QM/MM Approach: An Evaluation of Implicit and Explicit Solvation Models in SAMPL4.

(5) König, G.; Frank Pickard IV, B. C.; Jing Huang, B.; Andrew Simmonett, B. C.; Tofoleanu, F.; Juyong Lee, B.; Pavlo Dral, B. O.; Samarjeet Prasad, B.; Jones, M.; Yihan Shao, B.; et al. Calculating Distribution Coefficients Based on Multi-Scale Free Energy Simulations: An Evaluation of MM and QM/MM Explicit Solvent Simulations of Water-Cyclohexane Transfer in the SAMPL5 Challenge.

(6) Prasad, S.; Huang, J.; Zeng, Q.; Brooks, B. R. An Explicit-Solvent Hybrid QM and MM Approach for Predicting PKa of Small Molecules in SAMPL6 Challenge. *J. Comput. Aided. Mol. Des.* **2018**,

32 (10), 1191–1201.

(7) Ghasemi, F.; Mehridehnavi, A.; Fassihi, A.; Pérez-Sánchez, H. Deep Neural Network in QSAR Studies Using Deep Belief Network. *Appl. Soft Comput.* **2018**, *62*, 251–258.

(8) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754.

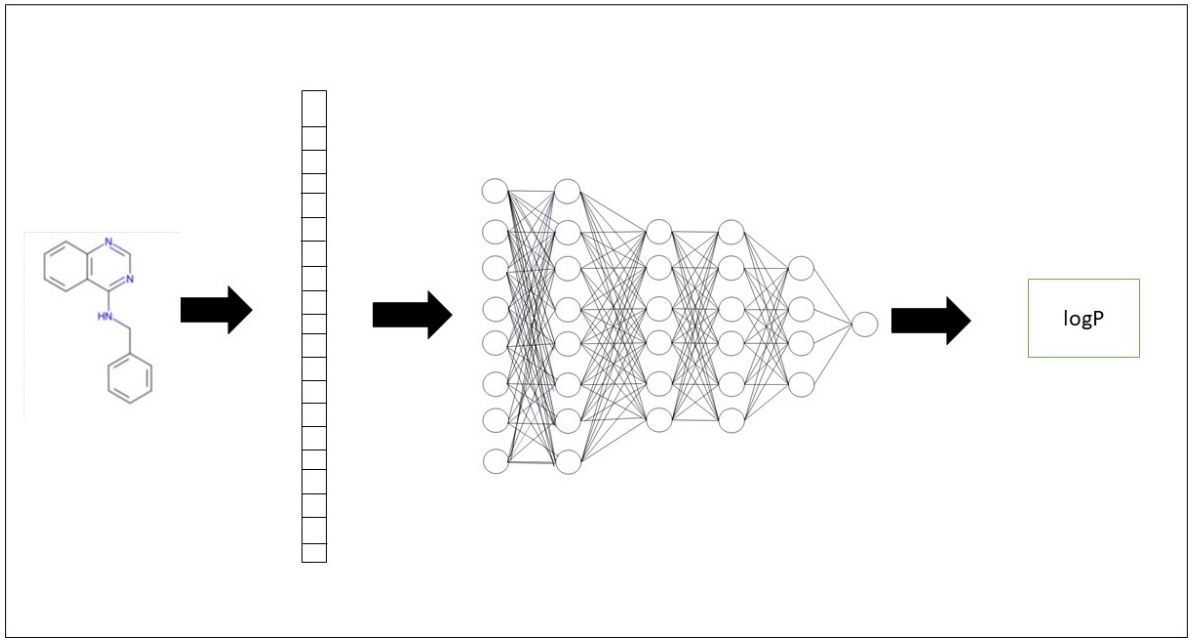


Figure 4.1: Schematic representation of the deep learning approach. In the first stage, molecule is transformed to its feature (vector) space representation using Extended Connectivity Fingerprinting. This serves as the input to the neural network. Neural network is trained on a large set of such molecules and corresponding logP. At the inference stage, output of the neural network is the predicted logP.

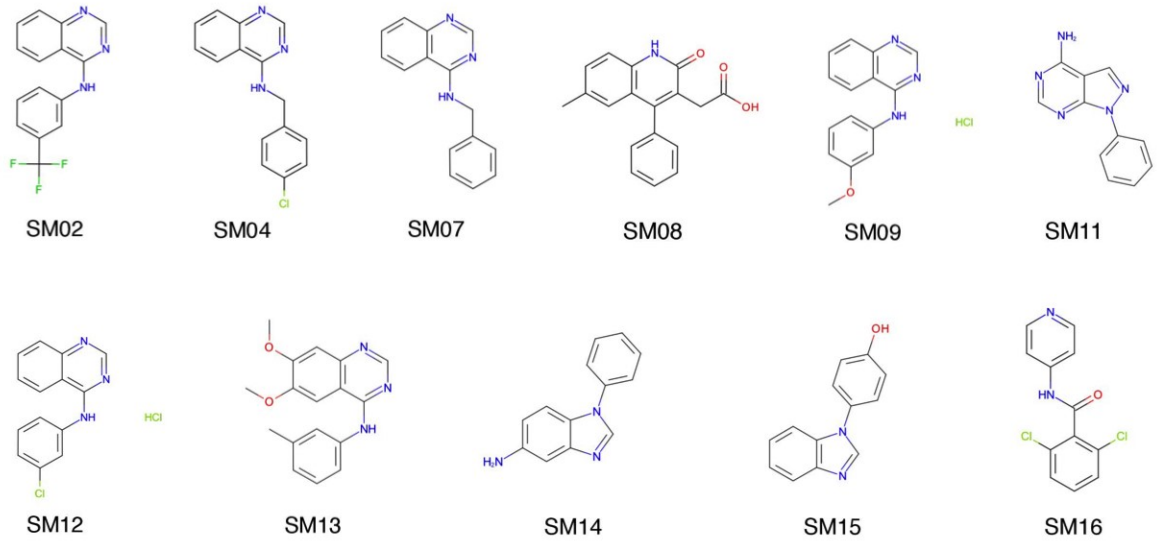


Figure 4.2: Two-dimensional structure of the SAMPLE6 logP challenge molecules. All the 11 molecules in this challenge were a subset of the previous pKa challenge.

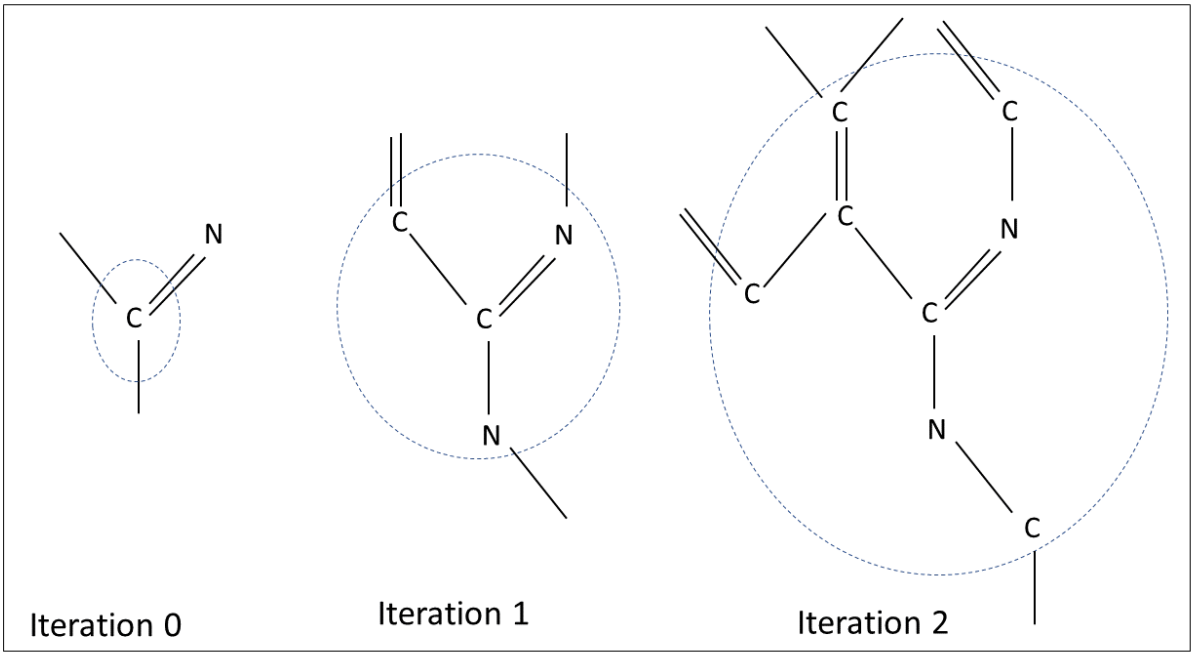


Figure 4.3: Schematic representation of the process of ECPC for three iterations for the identifier associated with the atom C represented above. After the zeroth iteration, the identifier associated with C is only about the atom and its bonds in the molecule. After the first iteration, the identifier also contains information about the atoms which are one bond away from atom C. The identifiers calculated after zeroth iteration for the neighboring atoms are used for creating the identifier for C at this iteration. After the second iteration, atoms within two bond distances from the center atom are included in the identifier. This iteration continues until a user specified iteration threshold is reached. In the present study four iterations are used.

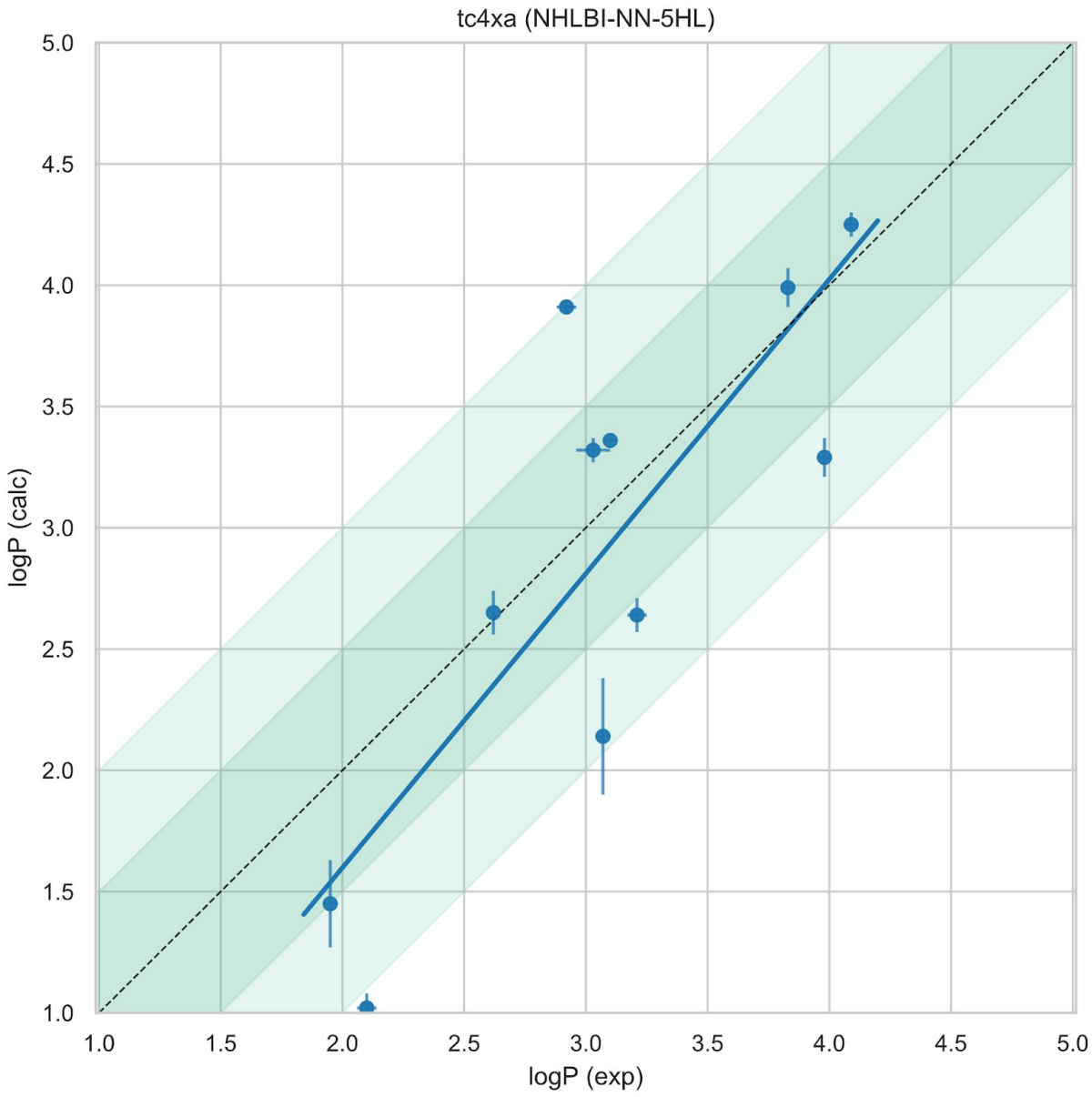
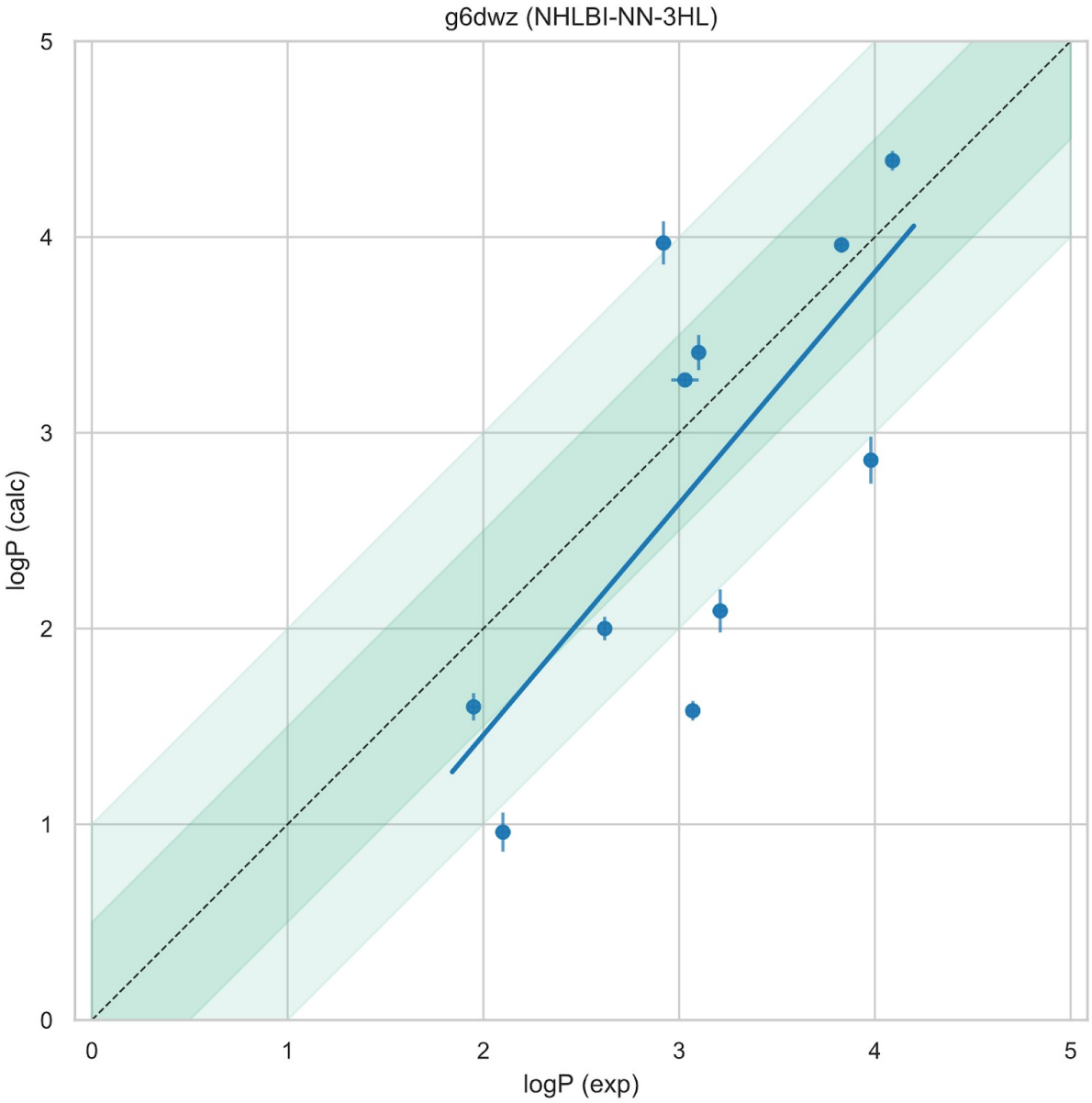
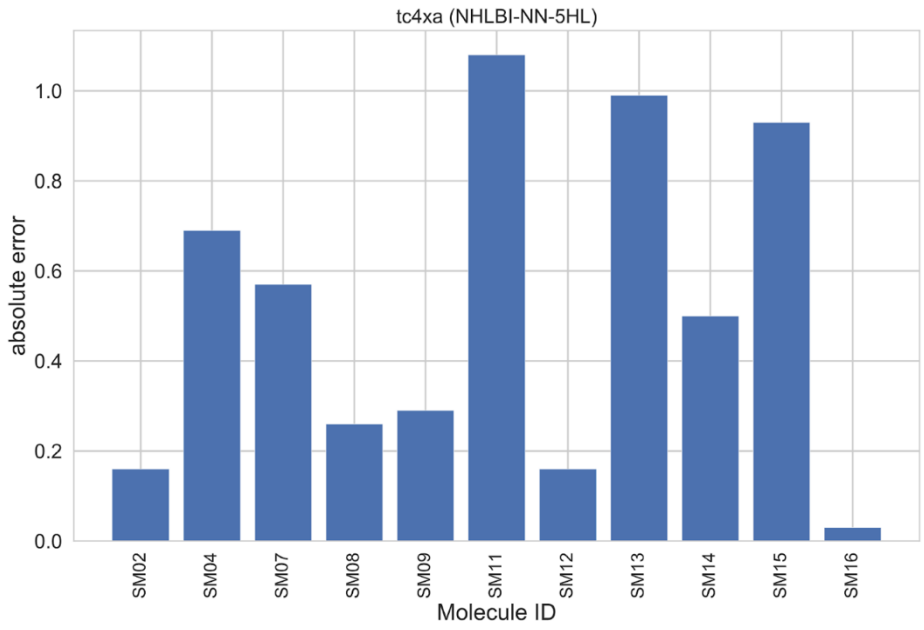


Figure 4.4

a) Experimental vs. prediction for the 5 hidden layers model. The darker shaded region is a 0.5 logP units while the lighter shaded region is the 1 logP units.



b) Experimental vs. prediction for the 3 hidden layer model. The darker shaded region is a 0.5 logP units while the lighter shaded region is the 1 logP units.



a.

b.

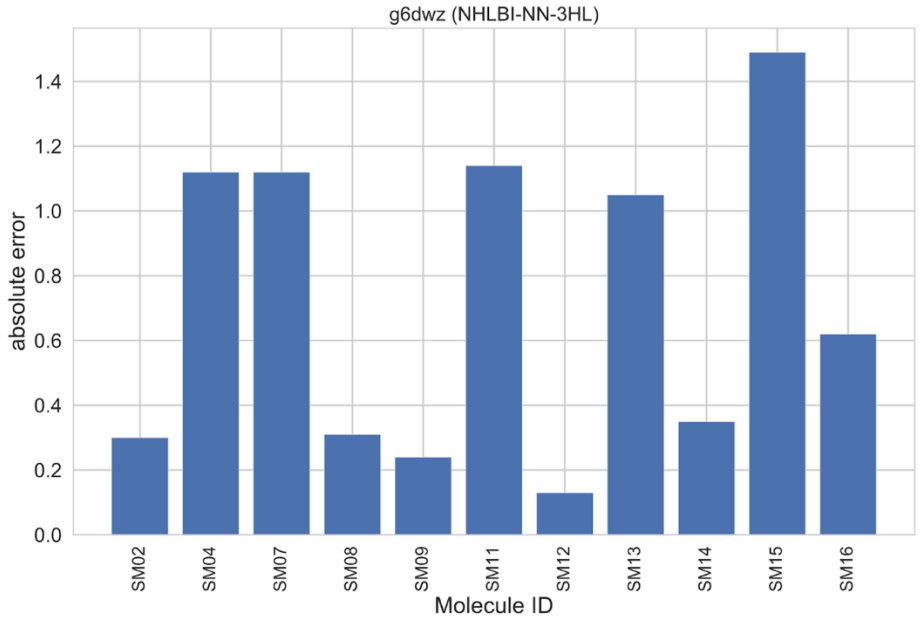


Figure 4.5: Absolute error for a. the 5 hidden layer and b. the 3 hidden layer models. Plots available at SAMPL6 logP repository as well.

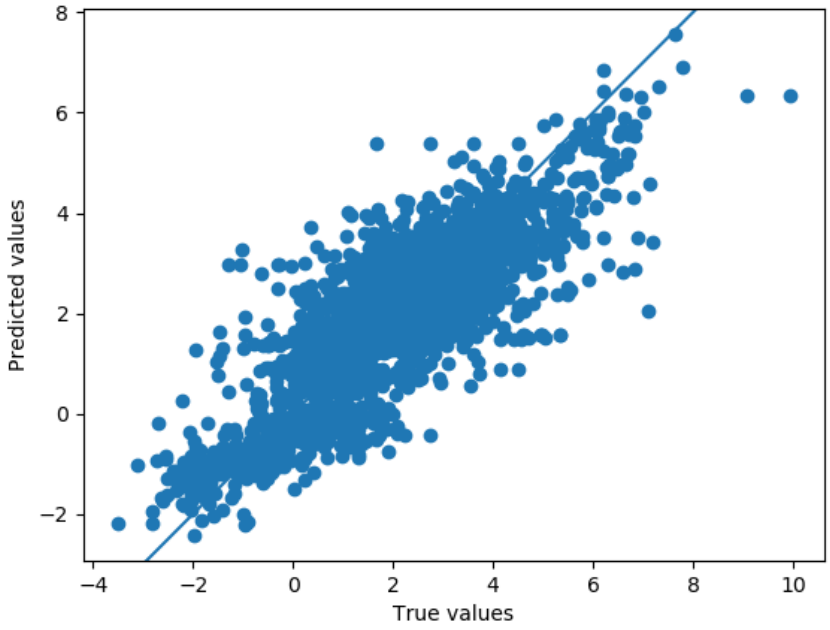


Figure 4.6: Plot of the predicted vs true logP values for 2000 molecules chosen randomly from the dataset.

SAMPL6 Molecule ID	isomeric SMILES	Experimental	ML predictions
SM02	c1ccc2c(c1)c(ncn2)Nc3cccc(c3)C(F)(F)F	4.09	4.25
SM04	c1ccc2c(c1)c(ncn2)NCc3ccc(cc3)Cl	3.98	3.29
SM07	c1ccc(cc1)CNc2c3ccccc3ncn2	3.21	2.64
SM08	Cc1ccc2c(c1)c(c(c(=O)[nH]2)CC(=O)O)c3ccccc3	3.10	3.36
SM09	COc1cccc(c1)Nc2c3ccccc3ncn2	3.03	3.32
SM11	c1ccc(cc1)n2c3c(cn2)c(ncn3)N	2.10	1.02
SM12	c1ccc2c(c1)c(ncn2)Nc3cccc(c3)Cl	3.83	3.99
SM13	Cc1cccc(c1)Nc2c3cc(c(cc3ncn2)OC)OC	2.92	3.91
SM14	c1ccc(cc1)n2cnc3c2ccc(c3)N	1.95	1.45
SM15	c1ccc2c(c1)ncn2c3ccc(cc3)O	3.07	2.14
SM16	c1cc(c(c(c1)Cl)C(=O)Nc2ccncc2)Cl	2.62	2.65

Table 4.1: List of the experimental and logP numbers

Method	**RMSE**	**MAE**	**R^2**	**m**
5 hidden layers	0.62	0.51	0.66	1.21
3 hidden layers	0.85	0.72	0.52	1.18

Table 4.2: Metrics of the results

Layer	Number of units	Number of parameters
Hidden Layer 1	512	524800
Hidden Layer 2	512	262656
Hidden Layer 3	512	262656
Hidden Layer 4	256	131328
Hidden Layer 5	256	65792

Table 4.3: Number of tunable parameters in the 5 hidden layer model.

Chapter 5

Implementation of CHARMM Molecular dynamics on GPU

Abstract

Chemistry at Harvard Molecular Mechanics (CHARMM) has been one of the most widely used molecular dynamics simulation package over the last few decades. However, the lack of an efficient CUDA implementation of CHARMM has hindered its usage the graphical processing units. In this chapter we discuss the migration of the code to the GPUs and provide technical details of the choices made for several key aspects of the implementation. Our results show that the new CHARMM-CUDA package provides similar speed of simulation as the other MD packages. Additionally, we support several other features of CHARMM including P21 periodic boundary condition, Enveloping Distribution Sampling (EDS) and others. The codebase has been redesigned to assist further extension in future.

1 Introduction

Molecular dynamics simulations are used in a variety of fields including material science and biomolecular sciences. At the heart of molecular dynamics simulations lies the calculation of forces on each atom from all the atoms in the primary box as well as their infinite images. Additionally, while the time step of a simulation is typically of the order of a femtosecond, most functionally relevant motions occur at microsecond or higher time scales. Even converged sampling of an ensemble requires more than 10^8 timesteps or more. Hence there is a need for high performance in this field.

Graphical processing units (GPU) have had an exemplary impact on scientific computation in general including molecular dynamics, computational fluid dynamics, astronomy, quantum chemistry etc. Several MD packages including Amber[1,2], OpenMM[3–67], NAMD[8], AceMD[9] and Gromacs[10,11] have been ported over to GPUs in the last few years.

Some of the initial GPU generations from NVIDIA included the modest Fermi, Kepler and Maxwell microarchitectures. The newer ones including the Pascal and Volta are improved versions with higher numbers of streaming multiprocessors (SM), higher memory bandwidths and higher floating-point (FP) operations per second. Table 5.1 gives a comparison of the salient features of Pascal GP100 and Volta V100 architectures. Each SM has 32 FP64 cores and 64 FP32 cores. Additionally, Volta has 8 tensor cores apart which support FP16 operations.

A preliminary version of GPU implementation for CHARMM was done in 2014 by Antti-Pekka Hynninen and colleagues. This implementation was based on a heterogeneous CPU-GPU design wherein only the direct space component of the non-bonded calculation was implemented on the GPU and the rest of the calculations were carried out on (possibly multiple) CPUs. This scheme was a natural extension of the eighth-shell method of the domain decomposition approach over multiple CPU nodes with the GPUs being used to offload the most time-consuming calculations. Subsequent attempts were also made to offload the bonded and reciprocal space electrostatic energy calculations to the GPU as well.

In this work we have changed the approach to switch from a heterogeneous CPU-GPU to a GPU-only implementation. This eliminates the need for transfer of data to and from the CPU's DRAM

and GPU's main memory at each time step. Our new version of GPU implementation does not use the eighth shell-based design. Rather it is optimized for a single-node efficiency. Data structures have been designed to support multiple PSFs in order to support Enveloping Distribution Sampling (EDS), MSCALE and PERT features of CHARMM in the next release. Special emphasis has been made for modular design of the codebase so that future extension of the code is convenient. We describe the computational designs in Section 2. This is followed by benchmark results in Section 3. In Section 4, we give concluding remarks and our plans for future development of the code.

2 Computational Details

In this section we will first discuss the software architecture of the CHARMM CUDA and then look specifically at the use of mixed precision computations for optimization of the direct space calculations.

Since we have periodic boundary conditions during simulation, non-bonded energy depends not just on atoms close to the site but also on their infinite images. However, this sum is only conditionally convergent and has a long tail. Ewald split the term into two terms - short and long range. Short range term decays quickly in the real space while the long-range component decays fast in the reciprocal space. In CHARMM, we calculate the long-range component using the SPME algorithm of Darden.

2.1 Direct space calculations

Direct space calculations are the most expensive steps during the energy and force calculations. They take more than 60% of the total time. Hence, this term is generally the site of most algorithmic optimizations.

2.2 Preparation of Neighborlist

Direct space component of the non-bonded energy calculation is performed in a tile-based interaction. This approach is the classic cuda-based approach for n-body problem used in many different fields. Since the warp size is fixed to 32 threads, we first divide the atoms into groups of 32. A naïve solution would be calculating the interaction of each tile in the simulation box with every other tile. However, direct space energy calculations have a distance-based cutoff for interaction. Hence, we would like to calculate the interactions of only those tiles which have any pair of atoms within the cutoff distance.

How do we divide the atoms into tiles such that we have to look at the least pairs of tiles while covering all the pairs of atoms that are within the cutoff? Another naïve method would be to assign the first 32 atoms to first tile, next 32 to the second tile and so on. Since we expect that the atoms close in sequence are also close in space, this would ensure that, to a certain extent, out of the 32*32 interactions of a tiles-pairs, many would fall within the cutoff. This assumption is valid for a long polypeptide; however, the assumption ceases to hold true for water molecules. During the

course of the simulation, water molecules can move away from each other, i.e. spatial and sequence proximity will provide very poor performance.

2.3 Reordering of atoms

A solution that performs well uses idea from cell-list and atoms reordering. First, we calculate the minimum and maximum extent of the coordinates along the X, Y and Z directions. Next, we calculate the dimension of the cell along the x and y axes. Using a uniform distribution approximation, this can be written as:

$$Delta = \left(\frac{xsize * ysize * zsize}{\frac{\#coords}{32}} \right)^{\frac{1}{3}}$$

$$Cell_x = \frac{xsize}{\Delta}$$

$$Cell_y = \frac{ysize}{\Delta}$$

$$Cell\_z\_max = 2\left(\frac{\frac{\#coords}{32}}{Cell_x * Cell_y} \right)$$

We are currently taking only an upper bound on the number of cells along the z-direction. This will be refined as we assign the atoms to the cells.

Having calculated these quantities, we can now proceed with the actual sorting of the atoms. First, we calculate the number of atoms in each z column and the z-column index for each atom in the simulation box. Next, we calculate the maximum number of atoms among all the Z-columns.

This is followed by calculation of the position of each of the atoms in the z-column. This is based on the z-coordinate of the atom. This is performed by a parallel prefix sum. This is followed by reordering of the atoms in the z-columns. Since we may have a race condition in this case, we use an atomic addition method to carry out this step. Finally, we can now sort the atoms according to the z-coordinates. For distributed computing, bitonic sort has previously been shown to be the most efficient. We describe the method in brief here.

2.4 Building of the neighborlist

Direct space non-bonded energy calculates the interactions on the content of the neighbor list data structure. In this subsection we discuss the building this data structure in detail. The entries in the neighborlist are based on the boundaries of the cells into which the simulation box was divided in the previous step rather than the coordinates directly. Coordinate information is present indirectly as the boundaries were created based on the coordinate. Building of the neighbor list happens in 3 stages:

1. In the first stage we build the basic data structure with the neighing cells with which a cell is within a cutoff.
2. Next, we add the topological exclusions for each pair
3. Finally, we sort the list of interacting cells for each i-cell in order to optimize the calculations.

First, we make an estimate of the number of tiles. Each cell can interact with cells along the positive direction which have any atom up to the short-range cutoff. This serves as the upper bound on the number of pairs of tiles that might have even one interaction. We can estimate the number of neighboring cells in each direction to cover up to cutoff.

Before we discuss the process of actual building of the neighborlist, it is interesting to look at the parallelization being invoked at this stage. One warp (consecutive set of 32 threads), takes care of one cell. If we have a block of 128 threads, each block has 4 warps. In other words, each block handles 4 cells. So, the number of blocks needed is the total number of cells divided by 4. Nvidia scheduler distributes the warps over the streaming multiprocessors available and efficiently switches between warps when one is interrupted to I/O or yields the execution or for any other reason,

As mentioned earlier, our implementation is optimized for a single GPU system. So, only one handles the entire simulation box. Hence, to find the cells that the current cell interacts, we search for neighboring cells for each of the 26 images and the primary cell. This is done three nested loops looking along the X, Y and Z axes.

We looked at the warp and block level distribution of work, i.e. each warp handles one cell and that warps are divided among the SM. We will now look at the thread level distribution of work within a warp. Each thread now works on a particular Z-column and iterates over the cells in the column to find the upper and lower bounds of the z-cells to search for interaction. Two shuffle

operations are performed among the threads in the warp to find the overall upper and lower bounds for the cellz along the Z-axis that the present cell can interact with.

The above search gives the cells with triplet(x,y,z)-index to consider for storing. At this stage, work is distributed among the 32 threads of the warp. Each thread computes the minimum distance that can be achieved between the boundaries of the original cell (image or primary) and the candidate thread-cell being considered. If this distance is within the cutoff, the candidate is added to the data structure. We also store the shift along the X, Y and Z axes that is used. For the primary cell within the simulation box, there is no shift while for all other images, the shifts are cached in order to reproduce the coordinate correctly at the time gradient calculation.

2.5 Adding exclusions

There are four different scenarios when a pair needs to be excluded:

1. There is no atom corresponding to the thread of Ith or jth cell for the tile. This is often the case when the number of atoms in the cell is less than 32
2. When i and j cells of a tile are the same - self interaction should not be calculated in short-range.
3. Topological exclusions: In most force-fields, including CHARMM, 1-2 and 1-3 terms are not included.
4. Avoiding the double counting: Each pair of interactions should be calculated only once. Hence only the top triangular region of a tile needs to be calculated. The lower triangular region is masked out.

2.6 Calculation of direct space forces, energy and virial

Here we iterate over the neighborlist we prepared in the previous step. Each warp (a collection of 32 contiguous threads) takes care of one tile (a collection of 32 atoms). So, in a block of 128 threads, we have 4 such tiles. Each warp runs together while a block has access to the same shared memory. Accumulation of forces is done in fixed precision[12]. So, we configure the shared memory as such - 32 integers for x-component of force, next 32 integers for y-component of force followed by the last 32 integers for z-component of the force. These forces are initially set to 0.

Each warp loads in the information for the i-tile: iatomstart, shift integer for the tile, tile for the start of j-list and the endtile for the j-list. Using the shift integer we first recover the shift we need to perform in the x,y and z directions in order to get the coordinates of the image/primary atoms.

We now start the iteration over the j-tiles. Exclusion is set to the lane exclusion for the j-tile's exclusions. Van der waals parameters are stored on the texture memory as both the c6 and c12 terms remain unchanged for an atom during the simulation.

Calculation of the components of the force along the axes and accumulation on the atom specific buffers is performed at this stage. First the force is scaled by a constant value. Next, it is multiplied by the x-,y-,z- components of a unit r.. It is stored in this format in the shared memory using atomic operation to avoid the possibility of a race condition.

2.7 Reciprocal space calculations

Reciprocal space calculations are related to solving the Poisson equation. It involves 5 steps:

1. Spread charge: This is one of the most time-consuming steps of reciprocal space calculation. This is where we are using the half precision interpolation. But for now, let me first discuss the normal way of performing this computation. We will be calculating the b-splines on the fly.
2. Direct to reciprocal space transform: cuFFT module of CUDA is used for this step
3. Scalar sum for energy calculation: Separate warp level buffers is maintained to avoid the possibility of race conditions.
4. Reciprocal to direct space back transform
5. Gathering the force happens using a finite difference approach.

2.8 Bonded interaction

Bonded terms account for the bonds, angles, urey-bradley, torsions, improper dihedrals and cmap terms. They are given by

$$U_{bonded} = \sum_{bonds} K_b(r_{ij} - r_0)^2 + \sum_{angles} K_\theta(\theta_i - \theta_0)^2 + \sum_{dihedrals} K_X(1 + \cos(n_x - \delta)) + \sum_{impropers} K_{imp}(\phi - \phi_0)^2$$

In order to calculate each of these terms we need the:

1. coefficients (ex: equilibrium bond length and force constant for the bond-term),
2. atoms involved in an interaction and
3. coordinates of the atoms involved.

The first two sets i.e. the coefficients and atom lists remain unchanged during the simulation. Multiple PSFs are supported by separate calculations. A list of interactions is stored in the global memory. Each of the types of bonded terms are launched asynchronously in its own kernel and iterates over its list of interactions.

Furthermore, mixed precision calculations are supported for bonded interaction as well i.e. accumulation can be done in fixed integer format while the individual force term can be calculated in double or single precision.

2.9 Holonomic Constraints

We support two kinds of constraints - SHAKE and SHAPE. The former is generally used for constraining the bond lengths for water molecule. The latter is more versatile and can constrain a set of atoms based on user-defined constrain group. It ensures that the angular momentum of the group remains unchanged at each time step.

2.8 Restraints

Restraint terms are similar to the bonded interactions. These terms are added as a separate energy term and one kernel takes care of all the restraints.

2.10 Integrators

A number of integrators have been implemented which allow the simulation a variety of ensembles. The basic microcanonical ensemble is implemented using Leapfrog, Verlet and velocity verlet integrators.

2.11 Langevin piston

Langevin piston method modifies the Andersen's barostat[13] to give an additional random force on the piston degree of freedom. Different versions of the barostat are implemented wherein one, two or three degrees of freedom along the crystal can be associated with barostat[13].

2.12 Precision model

Nvidia GPUs provide 2X performance for single precision calculations as compared to the double precision calculations. However, the range of floating numbers that can be represented in single precision in limited and adding a small single precision number to a larger one can give incorrect result due to imprecise representation errors. Accumulation of forces in single precision is hence not advised. However, performing the accumulation in fixed integer can solve this issue.

In CHARMM, fixed precision number are used for accumulation of forces and while floating point representation is used for calculation. The fixed-point version uses 34 bits for mantissa and 30 bits for the exponent.

2.13 Support for extensibility

An important aspect of the CHARMM CPU implementation is the adherence to object-oriented programming and modern C++ API design principles. This has been done to make further extension of the code in the future as convenient as possible. For example, we have a virtual class "Integrator" and all the varieties of integrators can be written as derived classes for this base class.

3 Results and Discussion

The new design of CHARMM CUDA engine improves the performance results quite significantly. Table 5.1 gives the results for number of nanoseconds of simulation that is achieved on Nvidia's GP100 processors. We are currently able to achieve upto 267ns/day for a DHFR system with ~24,000 atoms. With a ApoAI system of ~90,000 atom, throughputs upto ~87 ns/day can be achieved. For a cellulose system with ~400,000 atoms, the achievable speeds are around ~17 ns/day. These results are obtained on GP100 processor with the Boost turned off, i.e. clock speed of these processors is 1.3Ghz. Since these are single GPU performance numbers and very minimal work is done on the CPU side (just the I/O), a typical workstation with 4 GPUs can run 4 different simulations at the same time and achieve 4X performance.

One important design choice that we have taken in our implementation is the focus on single GPU performance. This design choice stems from the limitation posed by the bandwidth of the interconnect between the CPU and GPU as well as between the GPUs. Host CPU memory to GPU transfer is only around 12 Gb/s and is not expected to improve in the future at the saame rate as increase in number of Streaming Multiprocessors (SM) on the GPU. Using NVlink communication, peer to peer communication between device memory is ~380GB/s. However, this will still require keeping a coherent copy of the data (coordinates, forces etc.) and hence performance does not scale appreciably.

One of the major limitations of the earlier versions of molecular dynamics packages on GPU has been the lack of energy conservation due to the limit imposed by single precision floating units used for efficient calculation. Using single precision floating points is preferred as we can store more data in a similar space in memory. Also, more importantly, the throughputs supported by floating point computing units on the GPUs are twice that of double precision units. However, IEEE754 representation of single precision provides only 8 bits in the exponent and 23 bits in the fraction (and the most significant bit for the sign). The other related problem with the floating-point arithmetic is that mathematical operations are not commutative i.e. a + b is not the same as b+a. This leads to the force calculation not being deterministic. In a massively parallel architecture where the warps can be scheduled on the streaming multiprocessors non-deterministically, this poses a serious challenge to the reversibility in the simulation.

These limitations are handled by the use of fixed-point integers for the accumulation of forces. After performing the calculation in the single precision float, the force calculated on each atom of

a pair is converted to long integer (64-bit in size). A scale factor of 2^40 is also used to ensure that an underflow doesn't occur. After this, forces on each atom are accumulated in 64-bit integer format. To avoid race condition where multiple threads might be trying to add to the same memory location, the additions are performed as atomic operations. Finally, forces are converted back to a double or float based on the requirement.

Energy accumulation does not pose the same problem as in this case only it is only a scalar number has to be calculated. Since the numerical value for energy scales up with the size of the system, a single precision representation will overflow very quickly. Hence double precision is used for storing the energy terms. Also, energy is calculated only for the purpose of reporting and doesn't appear in the propagation of the dynamic's equations. So small precision errors in its value does not affect the simulation.

Another feature of the code is that the integration (update of the position and velocities) is performed on the GPU as well. This avoids the need to move data back and forth between the host and device memory at every time step. As shown in Table 5.2, Occupancy of the threads in each of the streaming multiprocessor is very high as a result

Simulation of different ensembles are now supported as well. For a microcanonical ensemble that conserves the total energy, verlet and velocity verlet algorithms have been implemented. For constant temperature (canonical) ensembles, Andersen thermostat has been added. I am currently working on implementing Nose Hoover chains on the device side code. For an isobaric ensemble, Langevin piston method has been added. This method allows one, two or three degrees of freedom

along the crystal axes to be scaled in order to match the internal pressure with the user expected average pressure value.

4 Conclusion

In this work we have implemented an entire GPU-only version of molecular dynamics package CHARMM. This implementation is more than an order of magnitude faster than the previous version of CHARMM-cuda. The code has been redesigned to move from a heterogeneous CPU-GPU architecture to one optimized on a single GPU. Since the phase space can be sampled in parallel by running independent simulations starting from different initial structures, just one node with 4 Nvidia V100 is capable of running more than 1 microsecond of simulation in a day for DHFR benchmark.

Usage of these improvements is under the hood with respect to the end-user. Instead of using the normal 'energy domdec gpu on' command to invoke the gpu implementation, user needs to use the command 'energy domdec gpu only'. All the energy calculations and integration are performed on the GPU and the host CPU will be used only at the time reporting values back to the user for input/output. This removes the frequent memory transfer between device and host memories and improves the throughput of the simulations.

We use single precision for performing the force calculation while the forces are accumulated into 64-bit integers. This not only increases the precision of the summation but also makes it

commutative. This allows the utilization of the single processing units, the work force of the GPUs, for all the force calculation. When used along with hydrogen bond constraints, this scheme allows excellent conservation of shadow Hamiltonian for a microcanonical ensemble i.e. the energy drift in the simulation is minimized.

5 References

(1) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9* (9), 3878–3888.

(2) Götz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *J. Chem. Theory Comput.* **2012**, *8* (5), 1542–1555.

(3) Eastman, P.; Pande, V. S. Efficient Nonbonded Interactions for Molecular Dynamics on a Graphics Processing Unit. *J. Comput. Chem.* **2010**, *31* (6), 1268–1272.

(4) Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. Accelerating Molecular Dynamic Simulation on Graphics Processing Units. *J. Comput. Chem.* **2009**, *30* (6), 864–872.

(5) Eastman, P.; Grønbech-Jensen, N.; Doniach, S. Simulation of Protein Folding by Reaction Path Annealing. *J. Chem. Phys* **2001**, *114*, 3823.

(6) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.;

Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; et al. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2013**, *9* (1), 461–469.

(7) Eastman, P.; Pande, V. S. OpenMM: A Hardware Independent Framework for Molecular Simulations. *Comput. Sci. Eng.* **2015**, *12* (4), 34–39.

(8) Stone, J. E.; Hynninen, A.-P.; Phillips, J. C.; Schulten, K. Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. *High Perform. Comput. 31st Int. Conf. ISC High Perform. 2016, Frankfurt, Ger. June 19-23, 2016, Proceedings. ISC High Perform. (31st 2016 Frankfurt, Ger.* **2016**, *9945*, 188–206.

(9) Harvey, M. J.; Giupponi, G.; Fabritiis, G. De. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *J. Chem. Theory Comput.* **2009**, *5* (6), 1632–1639.

(10) Kutzner, C.; Páll, S.; Fechner, M.; Esztermann, A.; De Groot, B. L.; Grubmüller, H. Best Bang for Your Buck: GPU Nodes for GROMACS Biomolecular Simulations. *Journal of Computational Chemistry*. 2015.

(11) Lindahl, E. Combined CPU-GPU Simulation in Gromacs Royal Institute of Technology.

(12) Le Grand, S.; Götz, A. W.; Walker, R. C. SPFP: Speed without Compromise—A Mixed Precision Model for GPU Accelerated Molecular Dynamics Simulations. *Comput. Phys. Commun.* **2013**, *184* (2), 374–380.

(13) Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. Constant Pressure Molecular

Dynamics Simulation: The Langevin Piston Method. *J. Chem. Phys.* **1995**, *103* (11), 4613–4621.

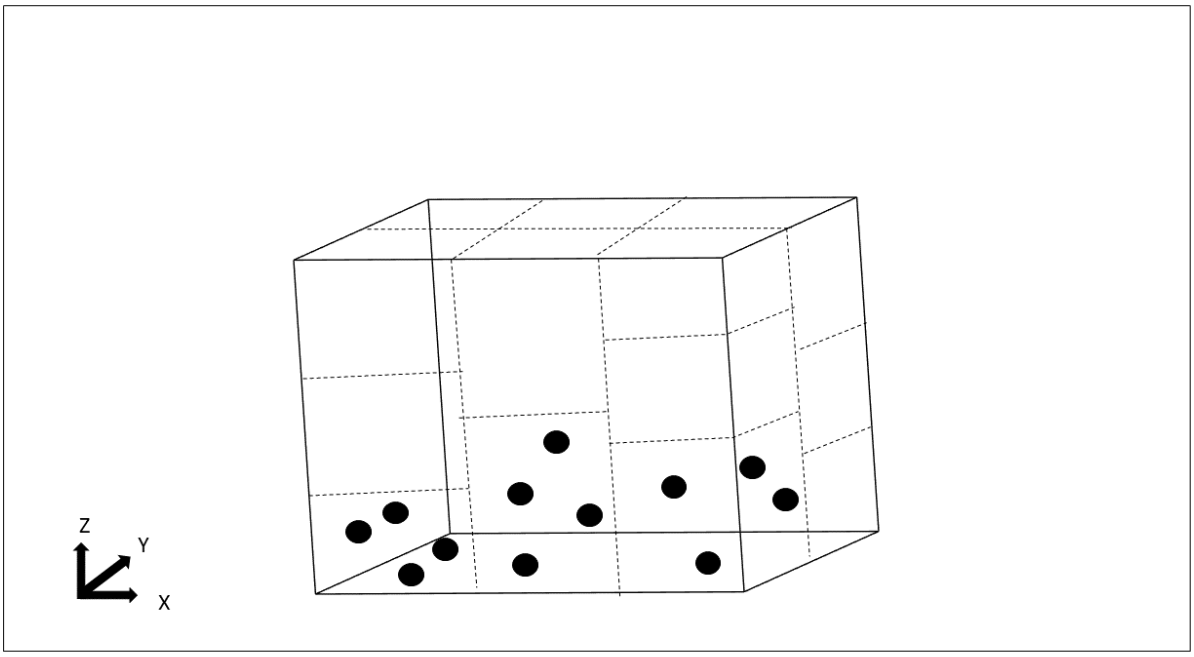


Figure 5.1: Schema showing division of the box into 32-atom cells. The x-y plane is first divided into smaller squares based on the density of the atoms in the box. Vertical z-columns are created and 32 atoms from the bottom of the column and sequentially collected to form a cell. The top cell in any column might not have 32 atoms in it.

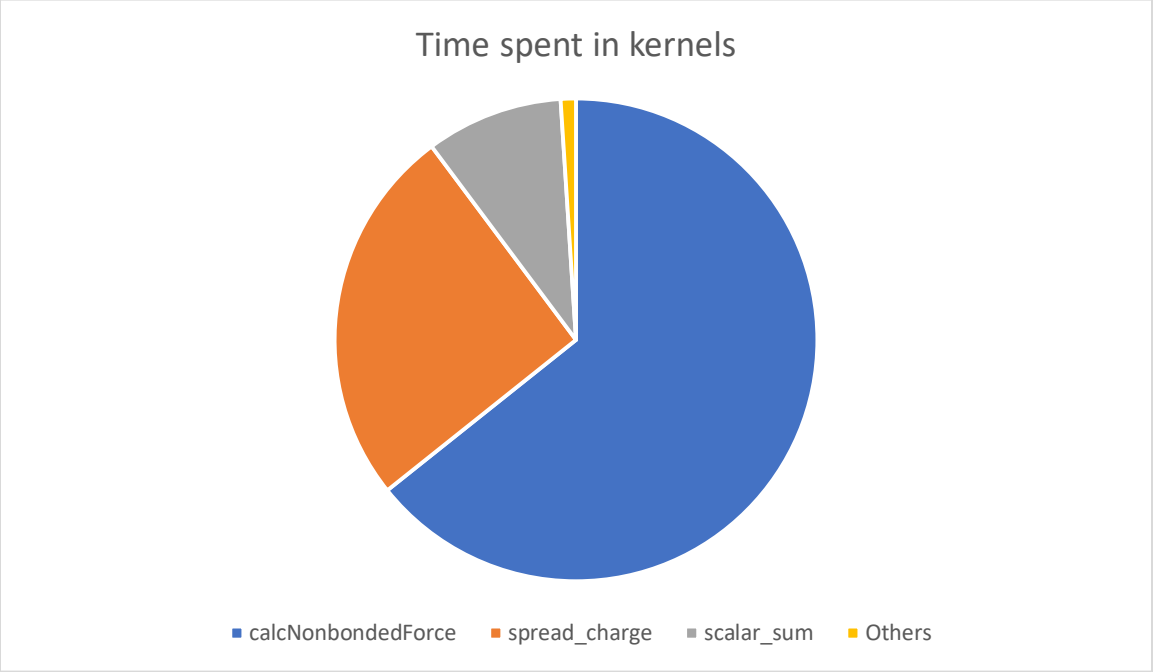


Figure 5 2: Split of the time between different kernels for a DHFR benchmark system of 23k atoms with 62.3 A box. Reciprocal space parameters are kappa: 0.34 and grid size of 108 along each of the axes. Nonbonded force calculation remains the most time-consuming section of the calculations.

	Pascal P100	Volta V100
Stream multiprocessors	56	80
FP32 cores/GPU	3584	5120
FP64 cores/GPU	1792	2560
Peak FP32 TFLOPS	10.6	15.7
Peak FP64 TFLOPS	5.3	7.8
Shared memory size/ SM	64 kB	Configurable up to 96 kB

Table 5.1: Comparison between Pascal and Volta architectures

Processor	Old-CHARMM	new-CHARMM CUDA
P100	7%	96%
V100	6%	92%
K20	7%	95%

Table 5.2: Occupancy of the streaming multiprocessors in a test run

Chapter 6

Conclusion

This dissertation has explored the application of accelerated computing and machine learning in the field of molecular dynamics simulations. The Extended Eighth Shell (EES) method solves a long-standing problem in the field of bilayer simulations to allow the balance of chemical disequilibrium created during the insertion of a drug molecule or peptide into one of the layers. Since area per lipids between the two layers changes, a chemical potential is created between the two layers. EES allows the lipids in one to move into the other layer. In contrast the normal periodic boundary conditions bring the lipids back into the same layer when they leave the simulation box. One of the most important aspects of EES is that it scales as efficiently as the best-known method for P1 PBC molecular dynamics.

Most physical phenomenon of interest in the dynamics of biomolecules occur in time scales higher than several nanoseconds, it is important for MD engine to be able to scale up to similar time scales. Since the MD engine has to calculate millions of interactions for billions of time steps, it is important for the energy and gradient calculations to be very fast. Keeping this in mind we have migrated the CHARMM molecular dynamics engine to the GPUs. The newer architectures provide more than thousands of cores. Such massive parallel machines have a more complicated programming model in order to use the underlying cores efficiently. We show that our new implementation, which performs all the operations on the device itself has the right precision model for the calculations of the gradients and that the hardware is utilized efficiently. We are currently adding more features to this engine.

In addition, we participated in SAMPL challenges which involved blind prediction of physical properties for drug like molecules. In the first phase, we developed a hybrid QM and MM method

to predict pKa of the molecules in explicit solvent. The method is interesting because it tries to combine the best of both the quantum and molecular mechanics world. Molecular mechanics cannot model breaking of bonds during deprotonation. Hence this method uses quantum level theory for that portion of the calculation. Modeling the solvent molecules explicitly is difficult at the quantum level due to the size of the system. Hence, we use molecular mechanics description at this level. The modified thermodynamic cycle allows the use of one aqueous phase and one gas phase calculation for each acid base pair instead of two aqueous phase calculations. Hence the method is much faster than the conventional thermodynamic cycles.

In the second phase, we built a deep learning model to predict the logP values of the molecules. The model is trained on a dataset of 12000 molecules with known logP values. We tried different architectures of the neural network with increasingly higher level of depth in the hidden layers. The model with fully connected 5 hidden layers performed the best among our models and gave close agreement with the blind dataset. The model can be extended further in the future for other physical properties as well.

PUBLICATIONS

(publication name: Samarjeet Prasad)

(related to the work done in Dr. Brooks group)

- Samarjeet Prasad, Simmonett AC, Brooks BR. **Extended-eighth shell method for periodic boundary conditions with rotations.** (in review)
- Samarjeet Prasad, Kraemer A, Jones MR, Hudson PS, Brooks BR. **A deep learning approach for blind prediction of logP values of drug-like molecules in SAMPL6 challenge**. (in review, to appear in September special issue of Journal of computer-aided molecular design)
- Samarjeet Prasad, Brooks BR. Implementation of optimized version of CHARMM on GPU. (manuscript in prep)
- Samarjeet, Huang J, Brooks BR. **A hybrid QM and MM approach for blind prediction of pKa of drug-like molecules for SAMPL6 challenge**. Journal of computer-aided molecular design 32 (10), 1191-1201.
- Braun E, Gilmer J, Samarjeet Prasad et al.. **Best Practices for Foundations in Molecular Simulations** [Article v1. 0]. Living Journal of Computational Molecular Science 1 (1), 5957
- Allen B, Chodera JD, Mey Antonia, Michael,J, Mobley DL, Naden L, Prasad Samarjeet, Rice J, Rizzi Andrea, Scheen J, Shirts M, Xu H**. Best Practices for Alchemical Free Energy Calculations**. (manuscript in preparation)
- Hudson PS, Kraemer A, Jones MR, Samarjeet Prasad, Brooks BR. **Blind logP prediction for SAMPL6 challenge using Alchemical free energy differences**. (in review, to appear in September special issue of Journal of computer-aided molecular design)

- Jones MR, Samarjeet Prasad, Hudson PS, Kraemer A, Brooks BR. **Blind logP prediction for SAMPL6 challenge using Qunatum mechanical approach**. (in review, to appear in September special issue of Journal of computer-aided molecular design)
- Konig. et.al. J Comput Aided Mol Des. 2016 Nov;30(11):989-1006**. Calculating distribution coefficients based on multi-scale free energy simulations: an evaluation of MM and QM/MM explicit solvent simulations of water-cyclohexane transfer in the SAMPL5 challenge.**

CURRICULUM VITAE

SAMARJEET

samarjeet@jhu.edu 443-627-1987

EDUCATION

- Johns Hopkins University – School of Medicine, (Oct 2012-July 2019)

  Ph.D. candidate, BCMB Program

  National Institute of Health (Aug 2016-July,2019)

  Graduate Partnership Program
- Indian Institute of Technology – Kanpur (2007-2011)

  Bachelor of Technology – Biological Sciences and Bioengineering

FELLOWSHIPS AND AWARDS

- Silver Medalist for 1st rank in the graduating class of the department at IIT-Kanpur
- Award of academic excellence (3 consecutive years, given to top 5% students)
- Best B.Tech Project in the department in the graduating class
- Mona and Paramjit Singh Scholarship, IIT-Kanpur
- Selected for Khorana Fellowship - 15 students are selected from India for summer internship at UW-Madison
- Awarded with CSIR Program for Leadership in Science (CPYLS) – for the top set of students in the state after Class 10th exams

PUBLICATIONS

- Samarjeet Prasad, Simmonett AC, Brooks BR. **Extended-eighth shell method for periodic boundary conditions with rotations.** (in review)
- Samarjeet Prasad, Brooks BR. et.al. **A deep learning approach for blind prediction of logP values of drug-like molecules in SAMPL6 challenge**. (in review)
- Samarjeet Prasad, Brooks BR. Implementation of optimized version of CHARMM on GPU. (manuscript in prep)
- Samarjeet, Huang J, Brooks BR. **A hybrid QM and MM approach for blind prediction of pKa of drug-like molecules for SAMPL6 challenge**. Journal of computer-aided molecular design 32 (10), 1191-1201.
- Braun E, Gilmer J, Samarjeet Prasad et al.. **Best Practices for Foundations in Molecular Simulations** [Article v1. 0]. Living Journal of Computational Molecular Science 1 (1), 595 .
- Allen B, Chodera JD, Mey Antonia, Michael, J, Mobley DL, Naden L, Prasad Samarjeet, Rice J, Rizzi Andrea, Scheen J, Shirts M, Xu H**. Best Practices for Alchemical Free Energy Calculations**. (manuscript in preparation)
- Hudson PS, Kraemer A, Jones MR, Samarjeet Prasad, Brooks BR. **Blind logP prediction for SAMPL6 challenge using Alchemical free energy differences**. (in review, to appear in September special issue of Journal of computer-aided molecular design)
- Jones MR, Samarjeet Prasad, Hudson PS, Kraemer A, Brooks BR. **Blind logP prediction for SAMPL6 challenge using Qunatum mechanical approach**. (in review, to appear in September special issue of Journal of computer-aided molecular design)

- Konig. et.al. J Comput Aided Mol Des. 2016 Nov;30(11):989-1006**. Calculating distribution coefficients based on multi-scale free energy simulations: an evaluation of MM and QM/MM explicit solvent simulations of water-cyclohexane transfer in the SAMPL5 challenge.**
- MS Kim, ..,Samarjeet,..., et.al. **Nature**. 2014. **A draft map of human Proteome**.
- Parashar P., Samarjeet et. al. **Dev. Bio**.2014. **Microarray meta-analysis identifies evolutionarily conserved BMP signaling targets in developing long bones.**
- Choi KD., ..,Samarjeet, et. al. **Cell Rep.** 2012. **Identification of hemogenic endothelial progenitor and its direct precursor in human pluripotent stem cell differentiation cultures.**

PRESENTATIONS

- ACS meeting April 2019. A parallel implementation of P21 PBC in CHARMM
- ACS meeting. August 2017. A method to balance the chemical potential difference between the bilayers using p21 periodic boundary condition
- SAMPL6 meeting. February 2018. A hybrid QM and MM approach for blind prediction of pKa of drug-like molecules for SAMPL6 challenge
- University of Delaware. Chemistry Department. March 2018. A hybrid QM and MM approach for blind prediction of pKa of drug-like molecules for SAMPL6 challenge

TEACHING EXPERIENCE

- Introduction to Bioinformatics (Teaching Assistant)

- Course instructor for BIOL262 at FAES NIH
- Mentored Dr. Julie Kim (contractor at NIAID) for machine learning under the NIH mentorship program

WORK EXPERIENCE

- Worked under Dr. Igor Slukvin at UW-Madison (Sep2011-June2012)

PROFESSIONAL SERVICE

- Vice President of the Graduate Student Association, JHMI (Aug2013-July2014)
- Alumni Contact Program, IIT Kanpur