

**Advancement of Molecular Mechanics Based Drug Discovery
Through the Use of Machine Learning**

by

Murchtricia K. Jones

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in The University of Michigan
2021

Doctoral Committee:

Professor Charles L. Brooks III, Chair
Assistant Professor Aaron Frank
Professor Kayvan Najarian
Professor Zaneta Nikolovska-Coleska
Professor Nils G. Walter

Murchtricia Kia Jones

murchkia@umich.edu

ORCID iD: 0000-0002-7193-6282

© Murchtricia Kia Jones 2021

*To Jesus Christ, My Lord and Savior.
This promise was made in 2015 and is now fulfilled.*

*To my husband, Jonathan, my daughter, Gabriella,
my mother, Patricia and my siblings.*

ACKNOWLEDGMENTS

As a young, black girl from the Virgin Islands with no access to scientific enrichment, I knew that the odds were against me. Combine that with being the child of a single mother and an incarcerated father; I was destined to become a statistic, but I knew that there was more for me. I had a greater calling on my life. There are so many people that inspired and assisted me on this journey. It would take an entire dissertation to thank you all. So, for those that I miss, I thank you!

Firstly, I would like to thank my advisor, Dr. Charles L. Brooks III. During the course of my PhD, he has been a source of wisdom; not only in my scientific work but personal life as well. He advised me on when to slow down and when to pick it up. He was understanding of my many personal life changes and health issues. I could not be more grateful for his caring heart. I thank you Dr. Brooks.

In addition, I would like to thank the Brooks Lab for being amazing colleagues and collaborators. Thank you to T.J, Luis and Jonah for running ML-MATCH into the ground time and time again so that it would improve. I thank you all for the guidance and insight into this project. I specifically thank Luis for all the hard work that he put in to run simulations for the testing of this software. Thanks to Ryan for his honest and constructive feedback during my PhD studies. Sincere thanks to David Braun for maintaining Gollum /Satyr and thank you for your patience in the many times that I broke things. To Amanda,

Heidieh, Efrosini and Sara, you women made lab doable. I thank you for your support and willingness to lend a listening ear. I appreciate you ladies more than you know.

To my committee members, Dr. Kayvan Najarian, Dr. Nils G. Walter, Dr. Zaneta Nikolovska-Coleska and Dr. Aaron Frank, I thank you for taking this journey with me.

Your patience and willingness to show your humanity is something I hope to emulate as a professor. I appreciate the time that each of you have taken out of your busy schedules to guide my dissertation. I thank you. To Dr. Aaron Frank, thank you for being a safe space over the years.

To Dr. Teresa Turner, you saw something in me and many other students at the University of the Virgin Islands that we did not see in ourselves. You encouraged me to attend summer research programs for every single summer of undergrad. You wrote countless recommendation letters and listened to my many sob stories. You were a scientific mentor who grew into a friend. I thank you for the amazing work that you have done for the students in the Virgin Islands, and I hope that one day I would have even a tenth of the impact that you have had in my life, in others.

To Dr. Douglas Iannuci, I thank you for sparking my love of mathematics. I “hated” math prior to being taught by you. Your knack for real world application and willingness to guide and mentor me will never be forgotten. You touched the lives of many with your charisma and character. May you rest in eternal peace. You are so missed.

To my DCMB Family, I thank you for your support in my scientific journey but more significantly in my outreach efforts. The development of InnoWorks would not have been possible without you all. To Julia, your bright and shining face has always been a light to me in our department and I so appreciate all the hard work you do. To Margit, thank you

for your counsel and support throughout the years. You were always available to me and pushed me. To Vy, Rucheng, Ashton and Negar, getting through the first two years of grad school would have been impossible if not for you all. From the study groups to dinner parties, you all keep me sane. We have gone our separate ways now and I wish each of you the best in your future endeavors.

To my H2O Campus Church Family, you were my rock during this time. I thank you for helping me to put things into perspective and your constant reminders to keep God first, always. To Chris, Julie, Nino, Tammy and Cindy, I thank you for your leadership and guidance.

Lastly, I thank my family. To my siblings, I thank you for always being a listening ear and source of unending encouragement. To my mother, Patricia Gomes, you have provided an amazing foundation for my success and growth. You were the single mother of five children and did your absolute best and I am so grateful for you and your sacrifices. You have supported me in every STEM program that I wanted to participate in so that I may have greater success. As your youngest child and first child to attend college, I know that you must have been so worried about me. You had many concerns about how I would thrive and how others would treat me. I am happy to tell you, "*Mami, I made it!*".

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vii
LIST OF FIGURES.....	viii
ABSTRACT	x
Chapter 1 Introduction	
1.1 Advancement of Drug Discovery Using CADD.....	1
1.2 Molecular Mechanics and Drug Discovery.....	4
1.3 The Small Molecule Parameters Issue	7
1.4 Process of Atomic Parameterization	9
1.5 Existing Solutions to the Small Molecule Issue.....	16
1.6 Machine learning and Force Field Development	23
1.7 Potential Drawbacks in ML based Force Field Design.....	24
Chapter 2 Machine Learning Multipurpose Atom Typer for CHARMM Methodology	
2.1 Background	27
2.2 Data Set	31
2.3 Approach and Algorithms	40
2.4 Discussion	51
Chapter 3 Machine Learning Multipurpose Atom Typer for CHARMM Results and Application	
3.1 Selection of Machine Learning Algorithm.....	62
3.2 Algorithm Performance on Test Set.....	71
3.3 ML-MATCH Models in Comparison to ParamChem Model	74
3.4 Validation by Free Energy of Hydration Calculations.....	76
Chapter 4 Implications and Future Directions	95

LIST OF TABLES

2.1 Count of generated atomic fingerprints per elements grouping.....	39
2.2 Random Forest model parameters considered for hyperparameter optimization.	44
2.3 Bayesian optimized hyperparameters for each atom type classification model...	44
2.4 Bayesian optimized hyperparameters for each partial charge regression model .	45
3.1 ML-MATCH vs ParamChem results for 6 FreeSolve molecules	75
3.2: Free energy of hydration results for GBMV2 and FACTS models	80
3.3 : Free energy of hydration results for GBMV2 with removal of trifluoromethyl containing molecules.	82
3.4 GBMV2 model results for ML-MATCH and ParamChem	83
3.5 Free energy of hydration results for FACTS with removal of trifluoromethyl containing molecules.	88
3.6 FACTS model results ML-MATCH and ParamChem.....	89
4.1 Results for ML-MATCH trained on AM1-BCC.....	97
4.2 MSLD binding free energy results for BACE inhibitors	103

LIST OF FIGURES

1.1 Report for European HTS market analysis	2
1.2 Representative workflow for computer-aided drug design.....	3
1.3 Graphic depicting that NNRTIs block a key hinge region in the polymerase region of RT.	7
1.4 Depiction of CGenFF overlapping local atomic environment descriptors	11
2.1 Workflow for the generation of ML-MATCH	30
2.2 Insight into structural makeup of CGenFF.....	32
2.3 Histogram of the element grouping counts in CGenFF	32
2.4 Atom type representation in CGenFF	38
2.5 Schematic of the generation of representative bonds, angles and dihedrals for a given force field	48
2.6 Schematic for the calculation of basis scores	49
3.1 Naïve Bayes classification model results on training set	66
3.2 Bayesian regression results for the prediction of partial charges	67
3.3 KNN training set results	69
3.4 Random Forest training set results	70
3.5 Testing results for Random Forest classification models.....	72
3.6 Testing results for Random Forest regression models	73
3.7 Six FreeSolve molecules that depict the span of the FreeSolve chemical space .	75
3.8 Depiction of the thermodynamic cycle for solvation free energy calculations.	77
3.9 GBMV2 model free energy of hydration calculations result	81
3.10 Atom types correlation matrix for FreeSolve molecules with differentially calculated FEHs by GBMV2	84
3.11 Partial charge comparison for FreeSolve molecules with differentially calculated FEHs by GBMV2.....	85
3.12 FACTS model free energy of hydration calculations results	87
3.13 Partial charge comparison for FreeSolve molecules with differentially calculated FEHs by FACTS.....	90
3.14 Atom types correlation matrix for FreeSolve molecules with differentially calculated FEHs by GBMV2	91
4.1 Crystal structure of β -Secretase PDB ID: 3SKF	99
4.2 Collection of BACE inhibitors	99

4.3 ML-MATCH vs ParamChem correlation matrices for atom type assignment ..	101
4.4 ML-MATCH vs ParamChem scatter plots for partial charge assignment	102

ABSTRACT

Drug discovery is the leading motivation for the development of new chemical entities. Improving computational methodologies is an important scientific endeavor for facilitating the development and optimization of new therapeutic agents. Particularly, this dissertation focuses on increasing the accuracy of molecular dynamics simulations which employ molecular mechanics force fields (MMFFs). MMFFs provide an atomistic representation of drug-target binding which enables the elucidation of structural information necessary to evolve compounds into viable drug candidates. The accuracy and efficiency of such computational assays are highly dependent on the initial set of force field parameters required to begin the simulation. Through many years of training and refinement, the parameters developed for macromolecules are well developed; however, the generation of force field parameters for novel chemical scaffolds can be challenging due to the vastness of small molecule chemical space. The work herein addresses this obstacle by employing machine learning models for the development of a framework which facilitates small molecule parametrization across various MMFFs.

The presented framework, Machine learning based Multipurpose AtomTyper for CHARMM (ML-MATCH), considers each molecule from an atom-centric viewpoint. This framework has two components, with the first being the machine learning

application. Using Random Forest, two key parameters can be predicted: atom types and partial charges. With the CHARMM General Force Field (CGenFF) as the training set, we found an average accuracy score of 96% for the classification of atom types and a Pearson R-value of 0.974e and RMSE of 0.028e for the assignment of partial charges. To validate the models, we compared ML-MATCH derived parameters to that of PARAMCHEM, the current gold standard for CGenFF based parameterization, for molecules within the FreeSolve Database. This resulted in an accuracy score of 90% for atom types and RMSE of 0.049e for partial charges. The second component of this framework is the MATCHing algorithm which serves to identify the closest MATCH between the bonded parameters of the query and those which exists in the force field's training set. ML-MATCH derived bonded parameters were validated by conducting free energy of hydration calculations for benzene derivatives within FreeSolve which were subsequently compared to both experimental free energies and calculated hydration free energies computed using PARAMCHEM derived parameters. With the GBMV2 implicit solvent model, we found an average Pearson R-value of 0.7223 and 0.4635 for ML-MATCH and ParamChem when compared to experiment, respectively. Similarly, for the FACTS model, we found an average Pearson R-values of 0.7505 and 0.5353. These findings show that ML-MATCH derived parameters are well-suited for reproducing experimental data in simulation. Application of ML-MATCH derived parameters in more complex simulations and retraining on various force fields, shows that this framework goes beyond the status quo of current atom parameterization software in its ability to identify the underlying rules and assumption for a given force field without being explicitly programmed to

do so. Therefore, the novel developed ML-MATCH platform for small molecule parametrization will be particularly useful for ligands in the studies of computer-aided drug design and developing therapeutic agents.

Chapter 1

Introduction

1.1 Advancement of Drug Discovery Using CADD

Over the last three decades, the utilization of computers for the prediction of chemical properties and structures of biomolecules has grown in both prevalence and necessity. Molecular modeling and computational chemistry have quickly become integral approaches for the modeling of complex molecular systems. Such approaches facilitate the understanding of complex molecular systems and prediction of their activity at an atomic level [1-2]. Theoretical chemistry, when coupled with efficient computer algorithms, allow for the imitation or mimicking of molecular behavior in an *in-silico* environment. These methodologies have a broad range of applications from material sciences, biophysics, biomedical engineering, and quite notably in the past few decades, the field of drug discovery [2].

Drug discovery is the process of identifying new chemical entities or repurposing existing ones to generate a medicinal therapeutic for a disease state. Concisely, one attempts to identify a lead compound that shows pharmacological activity against a biological target. Researchers must select the macromolecular target or pathways whose inhibition or activation will result in a positive disease resolution. This target's structure, which may range from that of a specific strand of RNA to a large membrane-

bound receptor, must be 'druggable,' i.e., able to bind a specific compound [3]. Once the target is identified, researchers must begin the long and risky process of lead identification [4]. High throughput screening (HTS) is employed to determine a large compound library's activity directed against the characterized target to determine those compound(s) with the most significant efficacy. Although this is the method of choice in the pharmaceutical industry, it has its limitations, including high cost and uncertainty of the mechanism of action (4). According to the 2018 Grand View Research HTS Market Report, the market size of HTS in 2016 was valued at 15.62 billion and is expected to expand 7.86% over the forecasted period until 2025. We see in Figure 1 from this report that more that 50% of the HTS market in Europe is targeted toward drug discovery.

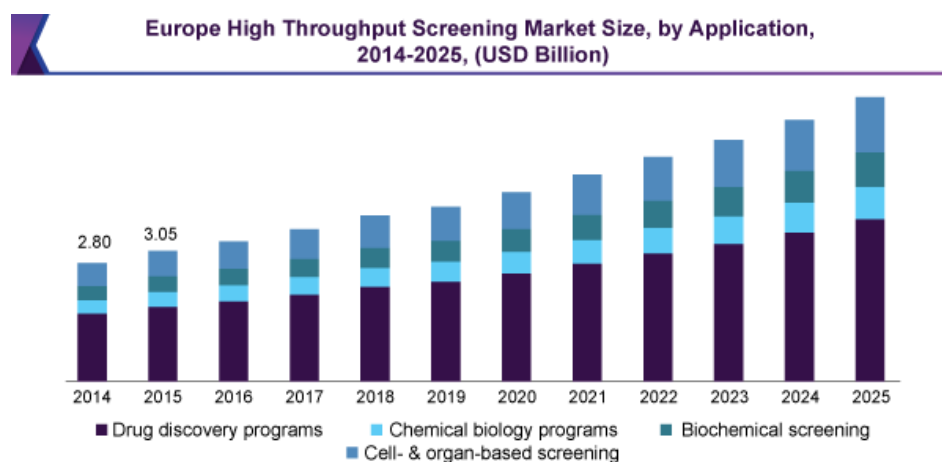


Figure 1.1: Report for European HTS market analysis. Graphic of the prediction of the HTS market comparing various HTS applications. From Grand View Research HTS Market Report.

To combat this cost and increase the certainty of activity between a potential drug and receptor, researchers have made strides in developing and refining computer-aided

drug design (CADD) methodologies. CADD, combined with wet-lab experiments, has been used to more rapidly elucidate the relationship between a potential drug candidate and its target [4]. CADD modeling strategies are categorized into structure-based drug design (SBDD) and ligand-based drug design (LBDD) methodologies. Generally, SBDD approaches use the 3D macromolecular structure of the target to identify potential modulators. As shown in Figure 1.2 taken from Macalino et.al., docking and scoring methodologies are used to evaluate ligands based on their intra-/intermolecular interactions within the binding region of the macromolecule [5]. Conversely, LBDD focuses predominately on a collection of molecules, normally with dissimilar structures, to determine their functionality when bound to a specific macromolecule which elucidates significant structural and physiochemical properties within the complex. In this project, we are particularly focused on SBDD and the use of molecular mechanics (MM) in solving macromolecule and ligand interactions.

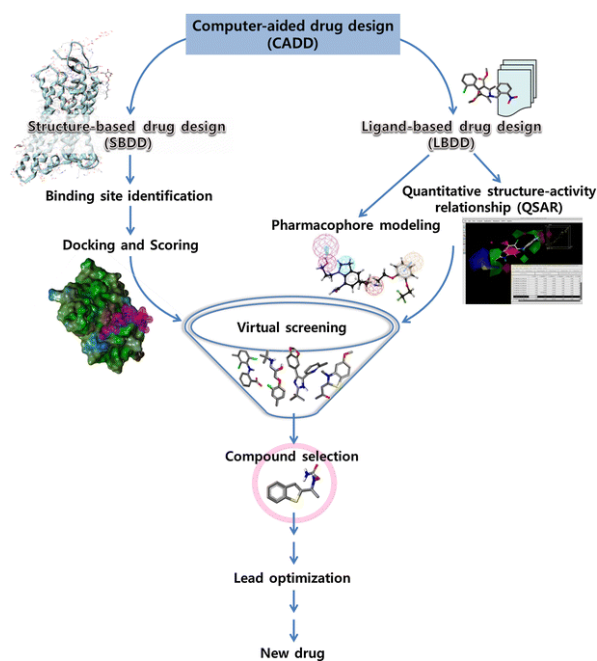


Figure 1.2: Representative workflow for computer-aided drug design. From *Arch. Pharm. Res.* **38**, 1686–1701 (2015).

1.2 Molecular Mechanics and Drug Discovery

Molecular Mechanics provides an atomistic depiction of drug-target binding interactions, which enable the elucidation of pertinent structural information necessary to evolve lead compounds into viable drug candidates through the use of MMFFs [6]. MMFFs are mathematical expressions that consist of an analytical form of the interatomic potential energy and the set of parameters that enter this form [7]. Due to the simplicity of the MM potential energy functional form, one can simulate very large systems. While the potential energy functional form is simple, which results in rapid and effortless calculations; the accuracy of such empirical methods greatly depends on the set of empirically derived parameters that enter this form to describe the atoms and their interactions. When an MMFF is well parameterized, it has a comparable or higher accuracy [8] when compared to high-level quantum mechanical methodologies. It is important to note that the generation of the necessary pre-defined parameters is time-consuming and computationally expensive. Initial simulation parameters are usually generated by *ab-initio* quantum mechanical calculations or by fitting to experiment.

As shown in Eq.1, an MMFF [1-2] quantifies both intramolecular and intermolecular forces within a simulation. The intramolecular part calculates the energies of four covalent bonded interactions including bond stretching terms, angle bending terms, torsional terms and improper terms. Bonds and angles are approximated as a function

of bond length (b), valence angle (θ), and their associated equilibrium force constants $[b_0, \theta_0]$, respectively. While the bond and angle terms dominate the local covalent structure around an atom, there are instances where the angular force constants (K_θ) are not high enough to reproduce the energetics of out-of-plane motions. Motions such as this are accounted for using improper dihedral terms as a function of the out-of-plane angle (ϕ) and its equilibrium force constant (K_ϕ). Lastly, the dihedral terms are a sum of cosine functions and a function of amplitude (ϕ) and phases (δ_n). In nonbonded interactions, the electrostatics are accounted for using Coulomb interactions between fixed point charges q_i and q_j , centered on the atoms. These electrostatic interactions are referred to as additive because the charges do not affect each other, and all the individual atom-atom electrostatic interactions can be summed to yield the total electrostatic energy of the system. For the van der Waals interaction component, a classical LJ 6-12 potential defined by radius ($R_{(min,ij)}$) and well depth (ϵ_i) is used.

Equation 1.1

Bonded (intramolecular, internal) terms

$$\begin{aligned}
 E_{bonded} = & \sum_{bonds} K_b(b - b_0)^2 + \sum_{angles} K_\theta(\theta - \theta_0)^2 + \sum_{\substack{improper \\ dihedrals}} K_\phi(\phi - \phi_0)^2 \\
 & + \sum_{dihedrals} \sum_{n=1}^6 K_{\phi,n}(1 + \cos(n\phi - \delta_n)) +
 \end{aligned}$$

Nonbonded (intermolecular, external) terms

$$E_{nonbonded} = \sum_{\substack{\text{nonbonded pairs} \\ ij}} \frac{q_i q_j}{4\pi D r_{ij}} + \sum_{\substack{\text{nonbonded pairs} \\ ij}} \varepsilon_{ij} \left[\left(\frac{R_{min,ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{min,ij}}{r_{ij}} \right)^6 \right]$$

Many empirical force fields exist. Those designed for biological macromolecules are AMBER [9-10], CHARMM [11] and GROMOS [12]. GAFF [13] and CGenFF [14] were developed to represent small organic molecules in complex with macromolecules. OPLS [15] and COMPASS [16] were initially developed to simulate condensed phase matter. GLYCAM [17] was specifically developed for carbohydrates. While many of these force fields' functional form is similar to that of Equation 1.1, they regularly differ in non-bonded terms and atomic parameters.

MMFFs have shown great success in predicting the affinities of ligands within the binding site of specific target and the experimental binding modes [5]. An example of this was shown in a study done by Ivetic and McCammon [18] in which they successfully elucidated the inhibition mechanism of HIV-1 Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs) when in complex with HIV-1 Reverse Transcriptase (HIV-1 RT) through the use of molecular dynamics which employ MMFFs. Using all-atom MD simulations of HIV-1 RT, both in apo form and in

complex with the Nevirapine, this project found that this NNRTI constrains a key rigid-body motion between the “fingers” and “thumb” domain of the p66 subunit in HIV-1 RT, shown in the figure below. This impaired movement resulted in the loss of catalysis for the polymerase activity of this enzyme. This was a key finding for this particular disease as it obstructs the HIV-1 retroviral proliferation by inhibiting its conversion into DNA which is necessary for viral replication.

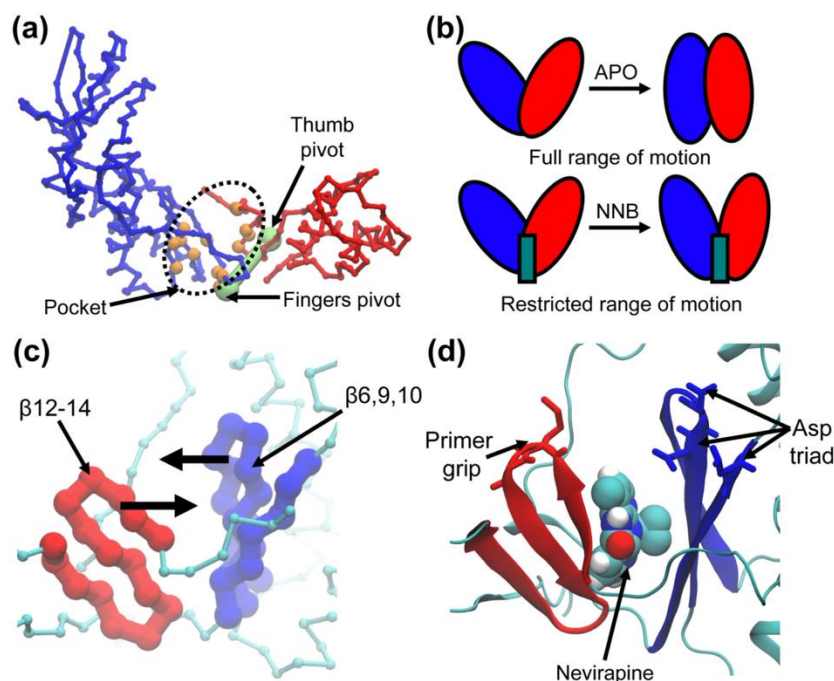


Figure 1.3: Graphic depicting that NNRTIs block a key hinge region in the polymerase region of RT. From J Mol Biol. 2009 May 8;388(3):644-58.

1.3 The Small Molecule Parameters Issue

Generally, to perform such molecular modeling experiments one must utilize two (or more) MMFFs. This first is a macromolecular force fields which represents the target. The second is a general (organic) force field which represents the small drug-like

molecule. These force fields are used to parameterize which moiety which provides the initial forces and interactions of the complex in simulation. In the example above, HIV-1 RT was parameterized using the GROMOS force field while the ligand, Nevirapine, was parametrized using the PROGRG2 program [19]. To ensure accurate interactions, researchers who have developed biomolecular force fields usually put forth the effort to create a matching small organic force field. Doing so is essential because of the inconsistent strategies used to optimize both bonded and nonbonded parameters and differential methods used to reproduce experimental data [20]. Thus, combining a random biomolecular force field and an arbitrary ligand force field would likely lead to unbalanced intermolecular interactions.

Due to many years of refinement, biomolecular force fields are well developed. However, the same cannot be said for small molecules force fields. This results from the vastness of chemical space and the virtually infinite ways in which functional groups may be bound. Creating small molecule force fields, which effectively spans this space, remains a challenge in the field. An even more significant challenge has been the efficient parameterization of many small molecules for high-throughput computational assays. To efficiently handle a substantial number of small molecules in MM calculations, one needs to develop a software framework which automatically assigns atom types, charges, bond types and then generate proper topologies that encode force field parameters for an arbitrary molecule.

1.4 Process of Atomic Parameterization

The paradigm of parameterization is vital in molecular mechanics. Forces fields and their associated parameters sets must be capable of reproducing experimental data for the molecules on which they were trained and chemical moieties outside of the training set [21-23]. As a result of this interest, researchers frequently thread the delicate balance between increasing the force field's accuracy and ultimately making a force field impracticable. There are three main steps in the parameterization process: atom typing, charges and parameter assignment which account for this imbalance [24].

1.4.1 Assignment of Atom Types

Atom typing is the task of assigning descriptive terms, called atom types, to each atom in a given system. Atom types aim to describe an atom's chemical environment such that it is readily distinguishable between atoms with different properties (chemical, structural, and electronic). Differing force fields have unique methodologies for defining atom types and in some cases an MMFF may not have atom types available. This task is quite simple and well developed in protein force fields. However, we have seen significant challenges in the assignment of atom types for small molecules due to the various ways functional groups may be arranged around a particular atom. The assignment of atom types is a compounding issue as different atom types are first assigned for each element. Concurrently, we must consider the differing hybridization states and chemical environment of each representation of that element within the training set. A solution for this has been to create more atom types which better depict the distinguishing features between atomic environments. However, having a higher

number of atom types escalates the chance that a user's molecule of interest may contain an arrangement of atom types that was not measured during the force field's design. Thus, decreasing the transferability of the force field from explicitly parametrized chemical groups to novel moieties.

In an effort to make these force field more transferable, researchers may use the same atom type for similar but not identical local atomic environments. This results in a compounding issue when we attempt to quantify the relationship between local atomic environment and atom types. This is due to the fact that there is no one-to-one mapping that exists for a given atom type because many local environments may describe a single atom type. Below is an example of this in CGenFF. The HGPAM1 hydrogen atom type is described as polar hydrogen whose environment can be both in a neutral dimethylamine and terminal alkyne. We see this particular atom type as both H1 in a reduced nicotinamide and HN1 in a dimethylamine. As seen in the figures below, the hydrogens have quite different local atomic environment, but according to the force field, they are assigned the same atom type. We see this across differing MMFFs, thus, we must take this into account when building a machine learning base atom typer.

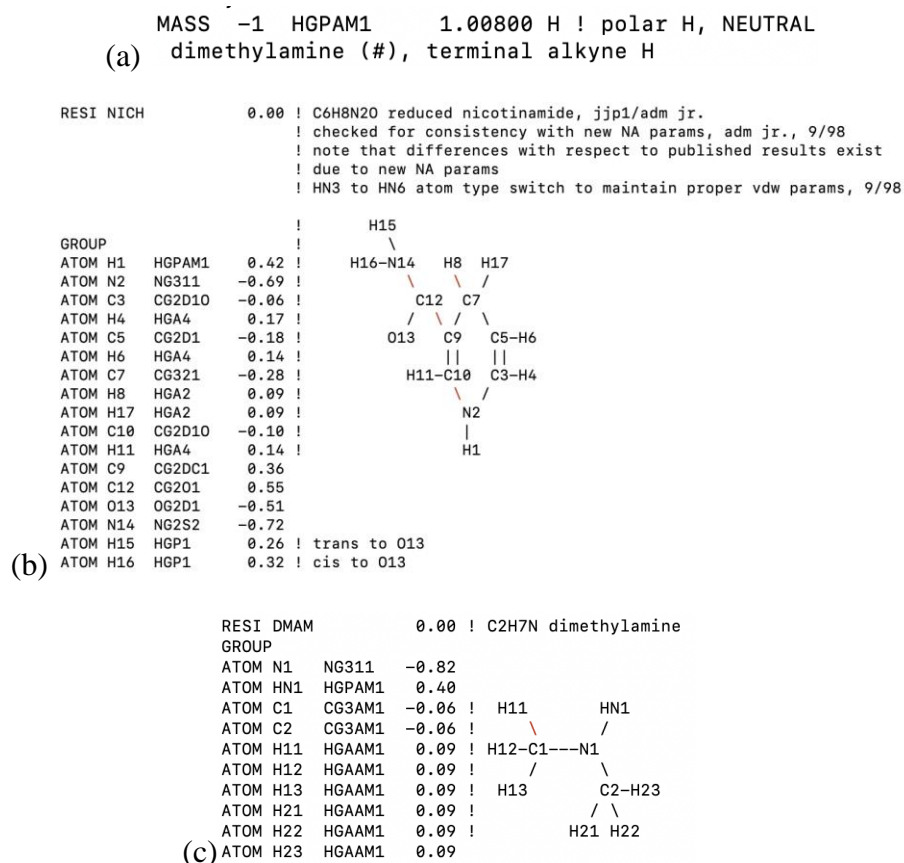


Figure 1.4: Depiction of CGenFF overlapping local atomic environment descriptors. (a) The description of HGPAM1 as defined in the CHARMM General Force Field. (b) Reduced nicotinamide parameters in CGenFF which depict HGPAM1 with a ring containing local environment. (c) Dimethylamine parameters in CGenFF which depict HGPAM1 in a less complex local atomic environment.

1.4.2 Assignment of Atomic Partial Charges

The second step of parameterization is the assignment of a partial charge to each atom in the system. Partial charges are typically assigned independently of atom types. The paradigm of charge assignment has been well-thought-out for protein force fields. They generally have set charges for each atom in a monomer, assigned using quantum mechanical target data then further optimized. Unfortunately, this method is not feasible for small molecules. Existing charging schemes primarily use “on-the-fly” charging methodologies including techniques such as bond charge increment schemes and electronegativity equalization. These methodologies can quantify atomic charges significantly faster than usual *ab initio* [25] approaches.

Bond charge increments (BCIs) describe the direction and magnitude of charge transfer between two covalently bonded atoms. These are often expressed in terms of bond charge increment rules related to the two atoms associated with the chemical bond. The purpose of bond charge increment rules is to define a set of BCIs can be extrapolated to novel chemical moieties [25-26]. This is accomplished by decomposing the atomic charge, defined by an *ab initio* method, of training set molecules into these increments. This method has seen success in a number of small molecule parameterization paradigms including ANTECHAMBER [27] and the Multipurpose Atom Typer for CHARMM (MATCH) [28]. BCIs are calculated for each bonded atom type pair that is represented in the training set. BCIs for each atom type combination are typically readable from a predetermined table. Once given a query molecule, BCIs are used to approximate the atomic partial charge of each atom.

The formal charge of the molecule is assigned by first setting a charge of 0 for all atoms except those in a chemical group having a net charge. The charges are then iteratively transferred between bonded partners.

Just as with the bond charge increment paradigm, the electronegativity equalization method [29-30] (EEM) is based on previously defined *ab initio* charges. The basis of EEM is closely related to the density functional theory (DFT) [31-32]. According to DFT, the charge-dependent electronegativity of an atom i in a molecule may be calculated as [33-36] ,

Equation 1.2

$$x_i = A_i + B_i \cdot q_i + \kappa \cdot \sum_{j=1 (j \neq i)}^N \frac{q_j}{R_{i,j}}$$

Where q_i and q_j are the atomic charges centered on atoms i and j , respectively, N is the number of atoms in the molecules, $R_{i,j}$ is the Euclidean distance between atoms i and j , and κ is the adjusting factor. Coefficients A_i and B_i are defined as,

Equation 1.3

$$A_i = x_i^* = x_i^0 + \Delta x_i$$

$$B_i = 2\eta_i^* = 2(\eta_i^0 + \Delta\eta_i)$$

where x_i^0 is the electronegativity of isolated neutral atom i , η_i^0 is the chemical hardness of atom i , Δx_i^0 and $\Delta \eta_i$ are descriptors of the molecular environment while the coefficients A_i , B_i and κ are empirical parameters defined by EEM parameterization. EEM [37] parameterization is done for the Hartree-Fock method with the STO-3G basis set and charges are calculated using Mulliken population analysis. The result of this parameterization will yield a readable table in which all defined atom types within the training set have a predetermined inherent electronegativity and chemical hardness.

To parameterize a new molecule, one must first calculate the instantaneous electronegativity of each atom as shown in Equation 1.4, x_i^0 is the inherent electronegativity, η_i is the chemical hardness and q_i is the predetermined atomic charge dependent on atom type.

Equation 1.4

$$x_i = x_i^0 + 2\eta_i q_i$$

Each atom's charge is then distributed in an iterative fashion until all atoms have an equivalent instantaneous electronegativity.

Although both methods have had some success, it is essential to note that there is no current "on-the-fly" charging scheme that is considered perfect. The development of such schemes remains a challenge in this field.

1.4.3 Parameter Assignment

The third step to parameterization is the assignment of parameters to all bonds, angles, torsion and improper dihedrals. This task is based on the previously defined atom types in Section 1.3.1. All covalent parameters are calculated using *ab initio* methodologies and readable from predetermined tables. Considering Equation 1.1, these parameters include the equilibrium bond length (b_o) and the bond force constant (K_b) for every covalent bond between existing atom types in the training set, the equilibrium angle (θ_0) and angle force constant (K_θ) for each covalent chain of three atom types and all improper [φ_0, K_φ] and dihedral [$\phi, \delta_n, K_{\phi,n}$] parameters for each relevant covalent chain of four atoms. Additionally, atom types carry a specific Leonard-Jones potential, which are averaged between atoms i and j to obtain parameters $R_{(min,ij)}$ and ϵ_{ij} . It is also important to note that empirical force fields calculate these averages using dissimilar methods. OPLS and GROMOS utilize the geometric mean to calculate both $R_{(min,ij)}$ and ϵ_{ij} . While AMBER and CHARMM use the Lorentz-Berthelot combining rules which quantifies the arithmetic mean for $R_{(min,ij)}$ and the geometric mean for ϵ_{ij} . It is because of this that it is ill-advised to transfer Leonard-Jones parameters between force fields.

My thesis will be focused on extending the current efforts of rapid atom parameterization. Below is an explanation of the currently available software.

1.5 Existing Solutions to the Small Molecule Issue

The overall purpose of parameterization is extrapolation. With *ab initio* calculations considered to be the “gold standard”, one expects to be able to parameterize novel chemical entities based on previously parameterized molecules. Thus, efforts have been put forth to generate parameterization models trained on this data to accelerate the process of parameterizing large compound libraries.

1.5.1 Antechamber Software

Antechamber is a software package for identification of bond and atom types, discernment of atomic equivalence, generation of topology files and investigation of missing force field parameters. Antechamber is made to be compatible with the AMBER molecular mechanics packages for automatic parameterization of small organic molecules. Antechamber is trained on the General AMBER Force Field (GAFF) which is made up of small molecules, selected to span a wide chemical space comprising of H, C, N, O, S, P, and halogens, which is compatible with the existing AMBER force fields for proteins and nucleic acids. This software package uses a simple functional form, similar to that of Equation 1.1, which has a limited number of atom types and incorporates empirical and heuristic models for the estimation of force constants and atomic partial charges. In GAFF’s functional form, K_r , K_u , and V_n , are the force constants for bond length stretching, bond angle bending and torsional angle twisting, respectively; r_{eq} , and θ_{eq} , are the equilibrium bond lengths and bond angles, respectively; γ is the phase angles of the Fourier series in the dihedral terms of out-of-plane angle ϕ ; A_{ij} , and B_{ij} , are the parameters of Lennard–Jones 12-6 potentials; q_i , and q_j , are the point charges of atoms i and j , respectively.

Equation 1.5

Bonded (intramolecular, internal) terms

$$E_{bonded} = \sum_{bonds} K_r (r - r_{eq})^2 + \sum_{angles} K_\theta (\theta - \theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2} [(1 + \cos(n\phi + \gamma))] +$$

Nonbonded (intermolecular, external) terms

$$E_{nonbonded} = \sum_{i < j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$

As with most parameterization paradigms, Antechamber begins with the automatic assignment of atom types based upon bond connectivity information within the provided input file (MOL2 or CSD). This methodology uses an atomic path concept that considers all possible paths from a particular atom in a molecule to a defined terminal atom. The path is then evaluated with a score function shown in Equation 1.6, where i is the position index in the path and an_i is the atomic number of the atom at position i . This score is then ranked by magnitude. If there exists an atom with the same score, those atomic environments are said to be equivalent. Atom types are then

assigned in a rule-based manner that considers bond connectivity (bond type, aromaticity etc.).

Equation 1.6

$$Score = \sum i * 0.11 + an_i * 0.08$$

Charges are generated using the AM1-BCC [31-32] model to which resemble restrained electrostatic potential (RESP) charges for a training set. Antechamber attempts to ensure that atoms with equivalent chemical properties have equivalent atomic charges. This is imperative for all automatic parameterization schemes for the accuracy of a MM simulation as well as to account for symmetric molecules and enantiomers. AM1-BCC has also been shown to have good performance in approximating molecular structure and conformational energy. The AM1-BCC scheme first calculates the Mulliken charges at the AM1 semi-empirical level and then conducts bond charge corrections to generate RESP-like charges. Jakalian and Bayly first introduced the method to develop a fast and efficient model AM1-BCC to generate high quality atomic charges which resemble RESP charges.

1.5.2 MATCH Software

MATCH [28] (Multipurpose Atom-Typer for CHARMM) is an automated toolset for the assignment of atom types and force field parameters to arbitrary organic molecules. MATCH was generated to be compatible with the CHARMM biomolecular force fields for protein, nucleic acids and carbohydrates and trained on the CHARMM

general force field (CGenFF). The CGenFF functional form is shown below in Equation 1.7. Equation 1.7 is very similar to Equation 1.1 with the main dissimilarity being the inclusion of the Urey-Bradley terms.

Equation 1.7

Bonded (intramolecular, internal) terms

$$\begin{aligned}
 E_{bonded} = & \sum_{bonds} \sum_{bonds} K_b(b - b_0)^2 + \sum_{angles} K_\theta(\theta - \theta_0)^2 + \sum_{improper\ dihedrals} K_\phi(\phi - \phi_0)^2 \\
 & + \sum_{dihedrals} K_{\phi,n}(1 + \cos(n\phi - \delta_n)) \\
 & + \sum_{Urey-Bradley} K_{UB}(r_{1,3} - r_{1,3,0})^2 +
 \end{aligned}$$

Nonbonded (intermolecular, external) terms

$$E_{nonbonded} = \sum_{nonbonded} \frac{q_i q_j}{4\pi D r_{ij}} + \epsilon_{ij} \left[\left(\frac{R_{min,ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{min,ij}}{r_{ij}} \right)^6 \right]$$

Unlike Antechamber, MATCH has the functionality such that it can be extrapolated to other existing force fields. MATCH is a fragment-based atom parameterization engine that allows for fragments from existing parameterized molecules to be applied or extrapolated to novel molecules. The MATCH algorithm represents molecular

structures as graphs, a methodology shown to excel in molecular pattern recognition. In MATCH, molecular graphs are used to quantify chemical environment similarity between atom types, ring identification and identification of out-of-plane geometry. Molecular graphs are generated based on structural information (PDB, MOL2, SDF etc.). Each atom is represented as a vertex which stores information about the atom's element, bond connectivity, ring membership and covalent neighbors. To calculate molecular graph similarity and identify the atom type, MATCH conducts a procedure similar to that of a tree data-structure comparison where first, an atom's feature must be contained in the larger graph and second, the element and bond connectivity of each existing node must be similar.

For the calculation of atomic charges, MATCH utilizes BCI rule libraries as described in Section 1.3.2. BCIs are generated for a specified training set and represented as a readable table which consists of the BCI between each existing covalently bonded atom types. A disadvantage of using the BCI rule is that the charging scheme is based solely upon connectivity that is present in the training set. If one attempts to parameterize a molecule that consists of a bond between two atom types which is not represented in the training, MATCH will be unable to type such molecule.

1.5.3 ParamChem

ParamChem [38-39] is a group of algorithms for the bond perception and atom typing of the CHARMM General Force Field, shown in Equation 1.7. As with other solutions, ParamChem first determines the valence, bond order and ring membership to the atom

and bonds in the molecule. It then assigns the partial charge to each atom in the system while using a matching algorithm for the assignment of other nonbonded parameters.

For the assignment of atom types, ParamChem uses a programmable decision tree to assign them based on a “language”. This algorithm is completely rule based and contains a number of categories that attempt to discern atom types from one another. ParamChem atom typer is based on a one-rule-per-atom-type scheme. The atom typer’s program first begins in the “main” or root node which first determines that atom’s element grouping in different subcategories. The tree then determines the hybridization and environment of said atom. This step is key to defining whether or not an atom is in or near a ring and ring type (sp^2, sp^3), aromaticity, bond order, resonance etc. As an atom moves through this decision tree, the atom typer moves closer to a decision on which CGenFF atom type class this atom most closely matches. Since this decision tree is specifically based on CGenFF, atom types are able to be defined with varying specificity. This scheme utilizes generic atom types for certain moieties and more complex atom types for others. It is also important to note that during the parametrization process of CGenFF, more and more specific atom types were added empirically as the need arose. A disadvantage to this approach is that this typing scheme will not be easily extensible to new chemical moieties and the rules for more specialized atom type would become exceedingly long and nontransparent in a one-rule-per-atom-type scheme.

As the purpose of small molecule force fields is extrapolation to unseen or unparameterized molecules, ParamChem consists of an algorithm to assign nonbonded parameters by analogy to determine the closest match for any missing bond, angle, torsion and improper parameters in the query molecule that does not exist in CGenFFF. For the charging of each atom in the molecule, ParamChem (34) uses an extended bond charge increment scheme. In this scheme, which looks at charge from an “atom-centric” view, the final partial charge of an atom can be formalized as $q_i = q_i^0 - \sum_j \beta_{ij} n_j$ where q_i is the final partial charge on atom i , q_i^0 is the previously assigned partial charge and β_{ij} is the bond charge increment between atoms i and j , where $\beta_{ij} = -\beta_{ji}$. Although this scheme is similar to that described in Section ..., ParamChem not only assigns a BCI to each covalent bond present in CGenFF but also to all angles and dihedrals. Thus, angles are assigned two charge increments (α_{ij}, α_{jk}) and dihedrals are assigned three charge increments ($\delta_{ij}, \delta_{jk}, \delta_{kl}$). If such a parameter does not exist, ParamChem uses charge increments from a similar atom type grouping.

A major pitfall of ParamChem is that it is specifically programmed to assign CGenFF parameters and is not easily extensible to other force fields. In addition, although ParamChem can be extended to atom groupings not parameterized in CGenFF it cannot parameterize unique atomic binding unrepresented in CGenFF. Thus, work needs to be done to build parameterization engines which may use alternative small molecule force fields as well as parameterize unique atomic bindings.

1.6 Machine learning and Force Field Development

This dissertation work addresses the above-mentioned pitfalls by creating a new automated atom parameterization scheme based on machine learning. Recent work has shown the ability of machine learning to capture the non-linear relationship between atomic configurations and potential energy [40-42]. In 1989, Feymann theorized that the forces experiences on atom is directly related to the configurations of the atoms around it [43]. Thus, if one can accumulate enough atomic environment to force examples, one should be able to derive the non-linear connections of said relationship and predict an atom's force (and the force acting upon it) from its structure. This has been done successfully in force field development research [44-46]. In addition to this finding, Bleizffer et.al has found that one can accurately predict partial atomic charges from the local atomic environment when depicted as atomic fingerprints [47]. Thus, a key question in this project is, "Can we predict empirical parameters for small organic molecules to provide an initialization configuration for simulation based on the individual atom environment?" We hope to predict atom types and partial charges while assigning non-bonded parameters by analogy.

With the use of machine learning, we have the ability to use QM derived parameters from a given force field to predict parameters of newly synthesized or not yet parameterized molecules. This project seeks to increase both the efficiency and accuracy of small molecule parameterization and provide a more easily updatable parameterization engine which is extensible across organic force fields as it represents atomic environments in a more generalized manner. This dissertation lays out the

accomplishment of these goals with the creation of a Machine Learning based Multipurpose Atom Typer for CHARMM (ML-MATCH).

1.7 Potential Drawbacks in ML based Force Field Design

As in the other applications of machine learning in force field development, there are expected hurdles that may be necessary to overcome in the creation of ML-MATCH. The most obvious and important hurdles being the set of molecules that form the basis on which the algorithm is trained [40]. Empirically fitted force fields are unable to traverse a vast space of atomic configurations. As a result, algorithms fit on such data will be unable to accurately describe molecules outside of this set, particularly those with diverse structural environments [41]. In the case of small molecules force fields such as CGenFF and GAFF, it would be difficult to well parameterize molecules with exceedingly complex chemical moieties, including rare functional group connectivity, that are not present in the force field's basis set. A well optimized machine learning algorithm can help overcome this hurdle [42]; however, one must be aware of the effects that such a drawback may have in simulation as a result of the uncertainty in parameter prediction. ML-MATCH provides a readily updateable pipeline and a single resource that can encompass many known force fields and can be easily extended to newly generated force fields.

References

1. Genheden, S., Reymers, A., et al., 2017, *Computational Tools for Chemical Biology*, pp. 1-38
2. Ramachandran, K.I., Deepa, G. & Namboori, K. *Computational Chemistry and Modeling: Principles and Applications*, Springer, 2008.
3. Hughes, J. P., Rees, S. S., Kalindjian, S. B., & Philpott, K. L. 6, 2012, *British Journal of Pharmacology*, Vol. 162, pp. 1239-1249.
4. Lionta, E., Spyrou, G., Vassilatis, D., & Cournia, Z. 16, 2014, *Current Topics in Medicinal Chemistry*, Vol. 14, pp. 1923-1938.
5. Macalino, S.J.Y., Gosu, V., Hong, S. et al., 2015, *Archives of Pharmacal Research*, Vol. 38, pp. 1686–1701.
6. Mohs, R. C., & Greig, N. H. 4, 2017, *Translational Research and Clinical Interventions*, Vol. 3, pp. 651-657.
7. Yu, Wenbo, and Alexander D MacKerell Jr. 20, 2017, *Methods in Molecular Biology*, Vol. 15, pp. 85-106.
8. De Vivo, M., Masetti, M., Bottegoni, G., & Cavalli, A. 9, 2016, *Journal of Medicinal Chemistry*, Vol. 59, pp. 4035-4061.
9. Pérez, A., Marchán, I., Svozil, D., et al., 2007, *Biophysical Journal*, Vol. 92, pp. 3817 —3829.
10. Duan, Y., et al., 2003, *Journal of Computational Chemistry*, Vol. 24, pp. 1999 —2012.
11. Brooks, B.R., Brooks, C.L. III, Mackerell, A.D. Jr., et al., *Journal of Computational Chemistry*, Vol. 30, pp. 1545-1614.
12. Oostenbrink C., Villa A., Mark A.E., van Gunsteren W.F., 2004, *Journal of Computational Chemistry*, Vol. 25, pp.1656–76.
13. Wang, J., Wolf, R. M., et al. 2004, *Journal of Computational Chemistry*, Vol. 25, pp. 1157-1174.
14. K. Vanommeslaeghe, E. Hatcher, et al., 2010, *Journal of Computational Chemistry*, Vol. 31, pp. 671-690.
15. Jorgensen, W.L., Maxwell, D.S., & Tirado-Rives, J., 1996, *Journal of the American Chemical Society*, Vol. 118, pp. 11225 —11236.
16. Sun, H, 1998, *Journal Physical Chemistry B*, Vol. 102 , pp. 7338 —7364.
17. Kirschner, K.N., et al., 2008, *Journal of Computational Chemistry*, Vol. 29, pp. 622 —655.
18. Ivetic, A., & McCammon, J.A., 2009, *Journal of Molecular Biology*, Vol. 388, pp. 644-58.
19. van Aalten DM, Bywater R, Findlay JB, Hendlich M, Hooft RW, Vriend G. P, 1996, *Journal of Computer Aided Molecular Design*, Vol. 10, pp. 255–62.
20. Ponder, J. & Case, D.A. 3, 2003, *Advances in Protein Chemistry*, Vol. 66, pp. 27-85.
21. Liang, G., Fox, P. C. and Bowen, J. P. 1996, *Journal of Computational Chemistry* , Vol. 17, pp. 940-953.
22. Norrby, P.-O. and Liljefors, T. J. 1998, *Journal of Computational Chemistry*, Vol. 19, pp. 1146-1166.

23. Faller, R., et al. 1999, *Journal of Computational Chemistry* , Vol. 20, pp. 1009-1017.
24. Pearlstein, A. J. Hopfinger R. A. 5, 1984, *Journal of Computational Chemistry*, Vol. 5, pp. 486-499.
25. Momany, F.A., 1978, *Journal of Physical Chemistry*, Vol. 82, pp. 592-601.
26. Halgren TA, Bush BL. 2, 1996, *Abstr Pap Am Chem S.* , Vol. 212, p. COMP.
27. Wang, J., Wang, W., Kollmann, P.A., & Case, D.A. 2001, *Journal of American Chemical Society*, Vol. 222, p. U403.
28. Yesselman, J.D., Price, D.J., Knight, J.L., & Brooks III, C.L. 2, 2012, *Journal of Computational Chemistry*, Vol. 33, pp. 189-202.
29. Radka Svobodová Vařeková, Zuzana Jiroušková Jakub Vaněk , Šimon Suchomel and Jaroslav Koča. 2007, *International Journal of Molecular Science*, Vol. 8, pp. 572-582.
30. P. Bultinck, W. Langenaeker, P. Lahorte, F. De Proft, P. Geerlings, M. Waroquier, J.P. Tollenaere. 36, 2002, *Journal of Physical Chemistry*, Vol. 106, pp. 7887-7894.
31. Yang, Parr and W. *Density-functional theory of atoms and molecules*. New York : Oxford Univeristy Press, 1989.
32. Flurchick, Libero J. Bartolotti Ken. *An Introduction to Density Functional Theory*. *Reviews in Computational Chemistry*. 1996.
33. Yang, Parr and W. *Density-functional theory of atoms and molecules*. New York : Oxford Univeristy Press, 1989.
34. Flurchick, Libero J. Bartolotti Ken. *An Introduction to Density Functional Theory*. *Reviews in Computational Chemistry*. 1996.
35. Wilfried J. Mortier, Swapan K. Ghosh, and S. Shankar. 15, 1986, *Journal of American Chemical Society*, Vol. 108, pp. 4315-4350.
36. Mortier, Karin A. Van Genechten and Wilfried J. 1987, *Journal of Chemical Physics*, Vol. 86, pp. 5063-5071.
37. Tomasi, E. Scrocco and J. VCH Publishers : *Reviews in Computational Chemistry*, New York, Vol. 5, pp. 171-227.
38. K. Vanommeslaeghe et al., 2012, *Journal of Chemical Information and Modeling*, Vol.52, pp. 3144-3154
39. K. Vanommeslaeghe et al., 2012, *Journal of Chemical Information and Modeling*, Vol.52, pp. 3155-3168
40. Deringer, V.L., et. al., 2020, *Nature Communications*, Vol. 11.
41. Deringer, V.L., et. al., 2019, *Advanced Materials*, Vol. 31.
42. Botu, V., et. al. 2017, *Journal of Physical Chemistry*, Vol 121, pp. 511-522.

Chapter 2

Machine Learning Multipurpose Atom Typer for CHARMM

Methodology

Murchtricia Jones and Charles L. Brooks III

2.1 Background

Computational drug discovery tools are rapidly developing to meet the challenge of designing and optimizing new chemical scaffolds [1]. In particular, the role of classical molecular dynamics (MD) and molecular mechanics simulations have been well established for the computational analysis of structure-based drug design including *in silico* high-throughput docking methods and statistical mechanical free energy approaches [1,2]. These methods provide an atomistic description of protein/ligand binding interactions using molecular mechanics force fields (MMFFs) [3]. MMFFs are mathematical expressions which represent the molecular interactions comprising the interatomic potential energy (U) and the set of parameters that best represent the fit of these expressions to a particular collection of molecules or molecular fragments, as shown in Equation 1.1 [3].

In a protein-ligand molecular simulation, the protein is represented by a specialized macromolecular force field, while the ligand is represented by a corresponding general organic force field [4,5]. Protein force fields have, over the years, been well-developed

and tested; however, organic force fields are continually growing due to the vastness of small molecule chemical space [5]. Thus, the parameterization of transferable and precise force fields for such entities has proven to be difficult. The task of parameterization also serves as a major bottleneck for subsequent molecular simulations due to the computationally extensive nature of representing this complicated quantum chemical behavior as a simplified analytical form.

Individual quantum mechanical (QM) calculations [6] for single molecules in a high throughput computational assay has its drawbacks [7, 8]. The most notable being the computational resources needed to perform such calculations. To accomplish this, researchers tend to use less accurate but rapid QM calculations that may cause improperly balanced intermolecular interactions when force field parameterization methods are “mixed and matched”. To address this bottleneck, efforts have been made to establish automatic atom typing toolkits; tools such as Antechamber [9] for the AMBER [10,11] biomolecular force field and its corresponding general AMBER force field (GAFF) [12], MATCH [13] and ParamChem [14] for the CHARMM additive biomolecular force field [15] and its corresponding CHARMM general force field (CGenFF) [16], LigparGen [17] for the OPLS [18] force field, and most recently the Open Force Field Initiative (OpenFF) [19]. However, the status quo as it pertains to current parameterization software is that each is specifically designed around a specific set of rules and assumptions for each developed force field and can be difficult to expand once force fields are updated.

This work aims to expand upon the efforts noted above with the generation of a newly automated ligand parameterization tool that utilizes a knowledge-based approach that exploits previously calculated force field parameters to infer information of not yet parameterized chemical scaffolds through inference based on a machine learning (ML) model. Utilization of machine learning to predict semi-empirical and force field parameters from QM reference data has been shown to have great success. Despite this success, only a few efforts have applied ML approaches for the learning and prediction of force field parameters within the context of existing force fields. In this project, we describe the development of the Machine Learning based Multipurpose Atom Typer for CHARMM (ML-MATCH); a framework based on machine learning for the classification of atomic environments and identify atom types and prediction of atomic charges based on a database of parameterized molecules. We developed a novel atomic fingerprint and were able to type molecules based on learned associations between the perceived environments and atomic force field parameters. Following this assignment of atom types and nonbonded parameters, internal energy terms were established using a hierarchical matching algorithm. ML-MATCH relies on the ability of the underlying algorithm to accurately predict the atom type and partial charge of each atom in a given molecule based on the local environment of that atom. Thus, it is critically dependent on the quality of data which is utilized in its training which is derived from *ab initio* calculations. This algorithm and subsequent work shows the ability of ML-MATCH to be extended to various general force fields depicting its independence of the physical principles of the force field on which it is based. We note that our objective here is to extend the scope of existing force fields in a manner that

is consistent with their underlying parameterization, and hence, presumably compatible with the remaining components of the molecular force field family of interest, e.g., CHARMM, AMBER, OPLS, etc.

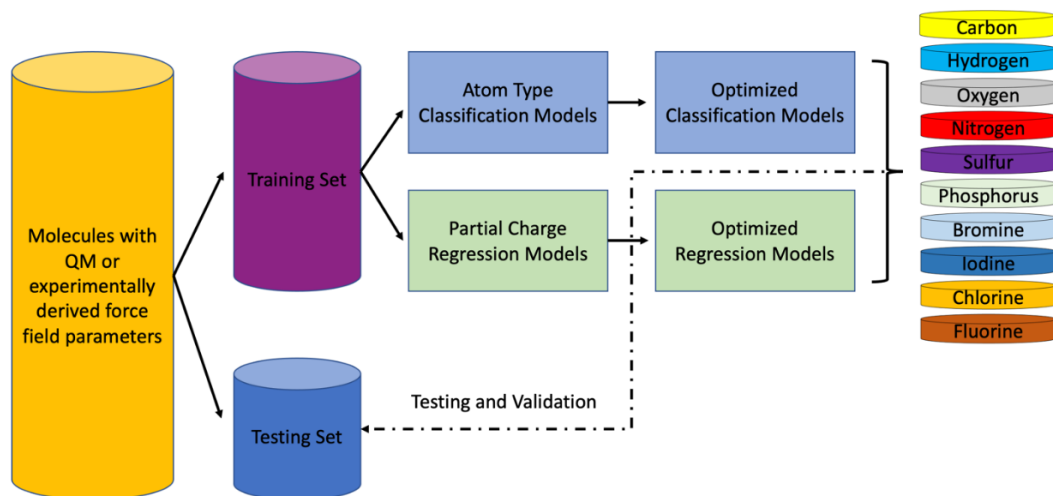


Figure 2.1: Workflow for the generation of ML-MATCH. ML-MATCH model generation follows a common preparation pipeline. Beginning with a curated group of parameterized molecules which are then split into training and testing set. The training set is used to select and optimize the ML algorithm of choice. ML-MATCH creates a classification and regression model for each element present in the basis set of molecules. Optimized models are then tested and validated using the testing set.

2.2 Data Set

2.2.1 Training Data

The objective of this effort is to take existing “well-vetted” force field models and extend the scope of represented molecules by the use of machine learning methods to map chemical environments to molecular force field parameters. However, the accuracy of a machine learning model greatly depends on the size of the data set on which it is trained. More recently, however, particularly in the field of drug discovery, it has been shown that one can use fragments of small organic molecules to predict characteristics of larger systems [20]. Thus, ML-MATCH will have capability to utilize “learned” parameters from small fragments to predict parameters for larger chemical moieties. CGenFF was compiled with various molecules and fragments meant to span the chemical space of drug-like molecules. CGenFF consists of ~500 lead-like molecules and fragments, i.e., predominately ≤ 7 rotatable bonds and molecular weights of 250-350 g/mol as shown in Figure 2.2. This force field and its corresponding database of molecules and molecular fragments contains organic molecules consisting of elements C, H, O, N, I, Br, Cl, I, P, and S. The prevalence of element groupings is shown in Figure 2.3. As is evident from this figure, some element types are abundant whereas others are minimally represented in the underlying molecular dataset. It is important to note that the parameterization of new chemical entities cannot go beyond the element types represented in the given force field as the atom types and intermolecular terms will be absent.

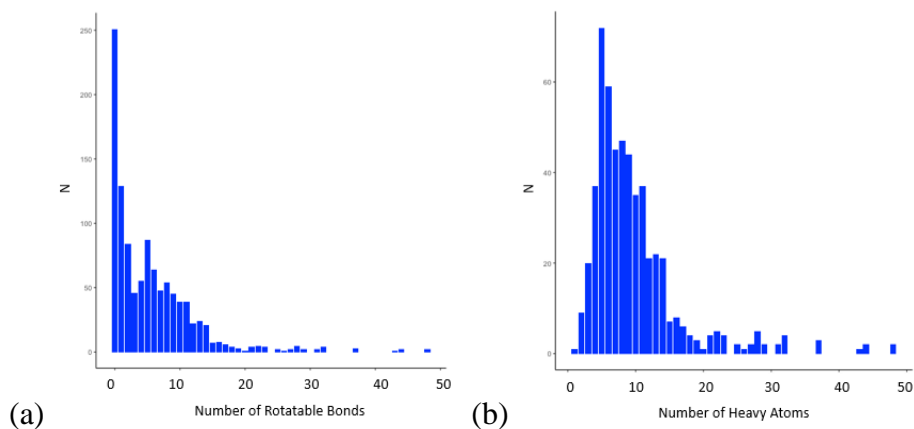


Figure 2.2: Insight into structural makeup of CGenFF. Histograms of number of (a) rotatable bonds and (b) heavy atoms for molecules in CGenFF.

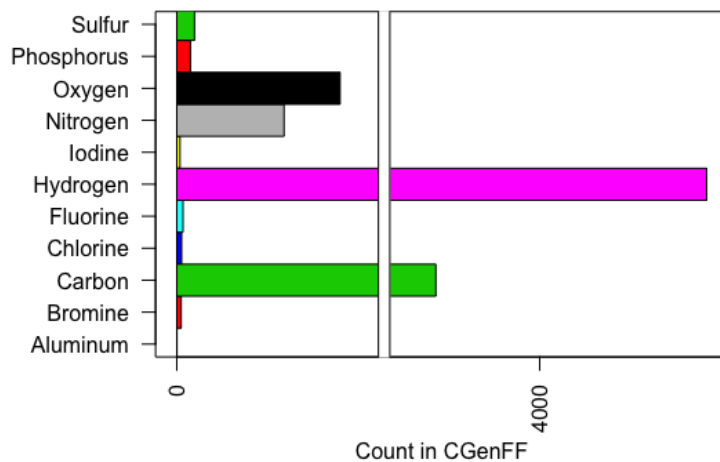


Figure 2.3: Histogram of the element grouping counts in CGenFF.

2.2.2 Chemical Space

To understand the extent to which our generated models can be extrapolated to novel chemical moieties (within reason) we must quantify the chemical space that our training set spans. To explore this question, we compared the CGenFF training set to

publicly available drug-like molecules in the ZINC15 [21] database. We used the Haider et.al checkmol [22] software to quantify the prevalence of over 200 functional groups. Using this software, we were able to identify the functional groups which are in common between CGenFF and ZINC15 FDA (a database of all FDA approved drug molecules). In addition, we identified the functional groups in which CGenFF has no representation when compared to ZINC15 FDA. While extending the range of molecules within CGenFF is beyond the scope of this work, the identification of missing and potentially important chemical entities or subsequent parameterization could increase the accuracy of both CGenFF and ML-MATCH as well as expand the chemical diversity of the force field such that one is able to better calculate pertinent ADMET characteristics of small organic molecules.

Once we acquired the count of existing functional groups in both datasets, the normalized count was calculated in order to more accurately compare the prevalence of each functional group between datasets. The normalized count was calculated as $x_{norm} = \frac{x - \min(X)}{\max(X) - \min(X)}$, where x is the actual count of a specific functional group and X represents the vector of all counts of that specific functional group within the database. The normalized counts [(0,1)] are shown in Appendix I Figures 1 and 2.

We see that for many moieties the CHARMM General Force Field shows lower prevalence than in ZINC15 FDA, these include groups like phenol rings and ring bound alkyl and aryl groups. In Appendix I Table 1, we see that there are 36 chemical moieties, as defined by the checkmol software, that are not present in CGenFF when compared to ZINC15 FDA. Appendix I Table 2 shows that there are 6 moieties that are present in CGenFF and not in ZINC15 FDA. To be more accurately used as a resource for the calculation of ADMET properties or protein-ligand binding, efforts should be made to expand the chemical space of CGenFF. This exercise gave us a good starting point in determining the chemical moieties for which ML-MATCH may not perform well due to lack of presence in the training set.

2.2.3 Newly Developed Atomic Descriptors

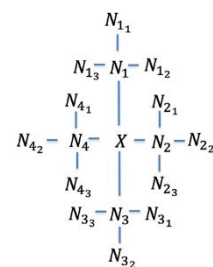
ML-MATCH looks at every molecule from an atom-centric view. As an effort to generate atom centered predictions, atomic fingerprint vectors have been utilized to relate the local atomic environment to a specific chemical or physical property. Atomic fingerprints are developed similarly to those generated for molecular fingerprints used for similarity substructure matching. Such vectors have shown great success solving the electronic structure of a molecule, understanding the physical properties of grain boundaries in crystalline materials and recently, structure-based predictions in drug discovery, to name a few [23-27]. The development of the numeric representation of the atomic environment is based on Feymann's findings [28], which essentially states that the force that an atom experiences is based on the configuration of its neighbors.

Consequently, atoms with similar environments are expected to have similar force field parameters, making this methodology a great fit for our efforts.

Effective development of these local atomic descriptors requires the fulfillment of mathematical properties such that developed models may be generalizable to unseen data. First, they must be correlated to the target property of interest, in the case of this project, the atomic environment must be able to identify the relationship between the local atomic environment with the atom type and partial charge, respectively. Secondly, this fingerprint must be invariant to the physical molecular structure while generalizable to a three-dimensional molecular representation. Thirdly, these vectors must be capable of capturing long distance interactions between atoms.

We developed new atomic fingerprints that are aimed to describe the local environment of each atom in a given molecule. The generation of these fingerprints was aided by the cheminformatics software, OpenBabel [29]. OpenBabel enables us to encode each molecule in CGenFF as a python object and through embedded functionalities provides chemical and geometric information of each atom in a molecule given an accurate starting structure representation.

Features considered include:



1. Atomic number; encoded by one-hot encoding (1,6,7,8,9,15,16,17,35,53)
2. Ring size; {openbabel.OBMol.OBAtom.MemberofRingSize() }
3. Hybridization {openbabel.OBMol.OBAtom.GetHyb() }
4. Valency; {openbabel.OBMol.OBAtom.GetValence() }
5. Additional Characteristics for functional group identification; encoded by one-hot encoding {openbabel.OBMol.OBAtom.[IsAromatic(), IsCarboxylOxygen(), IsPhosphateOxygen(), IsSulfateOxygen(), IsAmideNitrogen(), isNitroOxygen()]. }

Compilation of these features results in a vector length of 20 elements and when extended to second-nearest neighbor, by covalent bonds, each vector is extended to 340 elements. Inclusion of second-nearest neighbor was done to provide distinguishing features between similar atom types. Zero padding is used for atoms without second-nearest neighbors, i.e. if molecule of interest is a small molecule like methane. Permutations of each atomic fingerprint is performed to ensure invariance of bond paths from each central atom, (X), to the first $[(N_k)$ where k in $\epsilon [1,2,3,4]$] and second-nearest neighbor $[(N_{kj})$ where j in $\epsilon [1,2,3]$]. This yielded a total of 31,104 permutations with the removal of duplicates. In addition to accounting for invariance, permutations act to increase the representation of those atomic environments with limited presentation in a given force field. Figure 2.5 shows the varied representations of atom types (i.e. atomic environments) in CGenff.

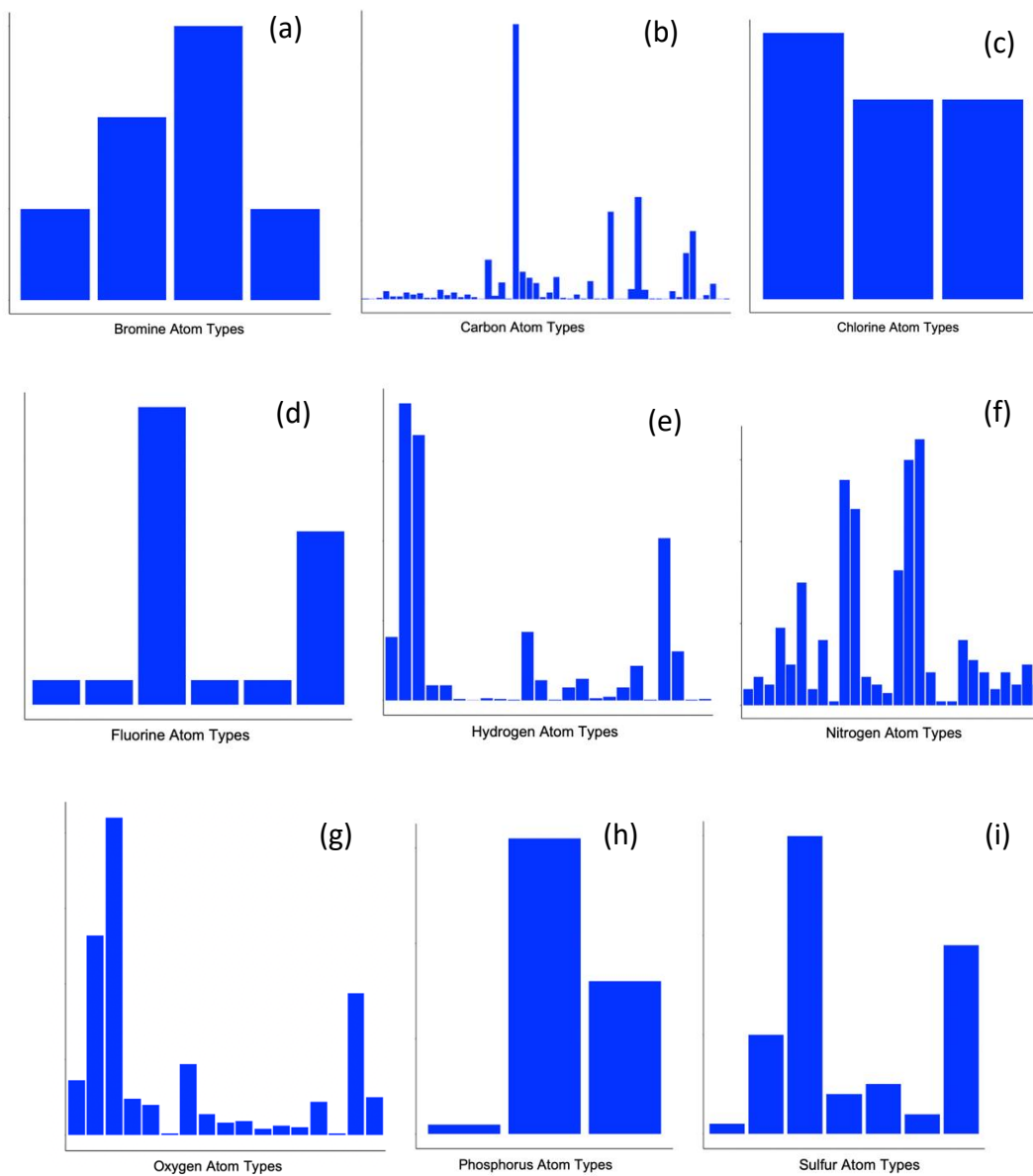


Figure 2.4: Atom type representation in CGenFF. Only 9 of 10 are shown in the figure since Iodine has one representative in this force field. In each subsection histogram there are differing levels of representation of atom types in each element grouping.

Element Grouping	Original AFp Count	Permutated AFp Count
Bromine	7	108
Carbon	3485	1217811
Chlorine	8	120
Fluorine	31	412
Hydrogen	4828	74476
Nitrogen	537	233996
Oxygen	810	53902
Phosphorus	86	4860
Sulfur	88	7826
Iodine	1	12

Table 2.1: Count of generated atomic fingerprints per elements grouping. Permutations of generated AFps for all atom in CGenFF allow a large training set and more chemical environment to parameter examples.

2.2.4 Atom type and Partial Charge Extraction from CGenFF

CGenFF uses the CHARMM additive potential energy function which consists of two terms: the intramolecular potential energy (bonded terms) and intermolecular potential energy (non-bonded terms). The CGenFF functional form is shown in Equation 1.2. The description of molecules within CGenFF consists of two data sources, the topology file, which consists of approximately 500 molecules whose atomic partial charges and atom types have already been assigned, and the parameter file, which contains all previously QM calculated bonded terms as well as Leonard-Jones potentials.

2.3 Approach and Algorithm

2.3.1 Machine Learning Algorithms

For each element grouping, i.e., [C, H, O, N, I, Br, Cl, I, P, S], a classification model and regression model was constructed, resulting in 20 different models. The newly developed atomic fingerprints were used to train the ML models. Classification was used for the prediction of atom types, labels which describe the local chemical environment around an atom, while regression was used for the prediction of all partial charges. Random Forest [30] was chosen as the underlying supervised machine learning algorithm used in ML-MATCH. Random Forests are a compilation of decision trees in which each tree is dependent on an independently and identically distributed random vector sampled from provided feature vectors (atomic fingerprints). An atom type classification decision, aT_{pred} , is predicted using the margin function described in Equation 2.2, where $h_{N_{trees}}$, $av_{N_{trees}}$ represent the individual tree-like classifier and average number of votes, respectively. X , Y represent the input atomic fingerprint and the randomly distributed feature set on which $h_{N_{trees}}$ is formed, respectively. The margin measures the extent to which the average number of votes for the correct class exceeds the average vote for any other class, otherwise known as bootstrap aggregation. While the prediction of atomic partial charge, q_{pred} , quantifies and average of all predicted charges given by each tree in the forest, as shown in Equation 2.3.

Equation 2.2.

$$aT_{pred} = mg(X, Y) = av_{N_{trees}} I(h_{N_{trees}}(X = Y)) - \max_{j \neq Y} (av_{N_{trees}} I(h_{N_{trees}}(X = j)))$$

Equation 2.3

$$\overline{q_{pred}} = \frac{1}{N_{trees}} \sum_{j=1}^{N_{trees}} (h_j(X))$$

For each atom in the molecule, the regression model predicts the partial charge independent of all other atoms within the molecule. As a result, a method of normalization, which distributes this excess charge around the molecule, is needed such that the sum of all partial charges in a molecule is equal to the molecules formal charge. A methodology such as this was employed recently by Rai et.al [31] and Bleiziffer et al. [32].

A molecule's formal charge, Q_{formal} , is computed algorithmically using the OpenBabel toolkit. Q_{pred} is calculated as the sum of the predicted charges, q_{pred_i} , where N_{atoms} is the number of atoms in the molecule. $q\Delta$ is the difference between the formal charge and ML-MATCH total predicted charges for a specific molecule.

Equation 2.4

$$Q_{pred} = \sum_{i=1}^{N_{atom}} q_{pred_i}$$

Equation 2.5

$$q\Delta = Q_{pred} - Q_{formal}$$

The standard deviation of the predicted charge for each atom in the molecule is defined as,

Equation 2.6

$$\sigma_i = \sqrt{\frac{\sum_i^{N_{trees}} (q_{pred_i} T_j - \overline{q_{pred_i}})^2}{N_{trees}}}$$

$q_{pred_i} T_j$ represents the predicted atomic charge for tree, T_j . $\overline{q_{pred_i}}$ is the overall predicted charge of the model given as the average of all tree predictions. Finally, the normalized charge is given as,

Equation 2.7

$$q_{norm} = q_{pred_i} - \frac{\sigma_i |q_{pred_i}| q\Delta}{\sum_a^{N_{atom}} \sigma_a |q_{pred_a}|}$$

The charge renormalization scheme is shown in Appendix I.

2.3.2 Calculation of Parameter Metrics

As an effort to provide a metric of predictive certainty, our algorithm provides scores which describe the algorithm's certainty of atom type classification and standard deviation of partial charge assignment. These metrics take advantage of the ensemble nature of the random forest algorithm. The certainty of prediction refers to the percentage of trees in the random forest which made the decision that corresponds to the selection made by the margin function in Equation 2.8, where k is the number of trees with max vote and N is the total number of trees.

Equation 2.8

$$\alpha = \frac{k - N_{trees}}{N_{trees}}$$

While the standard deviation-based charge metric is calculated as in Eq. 2.8. This metric gives the user of ML-MATCH an idea of how well each tree in the random forest correlates

with each other, and consequently a measure of the confidence of the atomic charge. Lastly, we provide the user with the out-of-bag error for each prediction.

2.3.5 Optimization of Random Forest Models

Each model in ML-MATCH is implemented using the sci-kit [33] learn module in Python. To assign atom types the RandomForestClassifier ensemble algorithm was employed while RandomForestRegressor algorithm was employed for the assignment of partial charges. To ensure optimal performance for each classification and regression model, we must carefully tune them to identify the model parameters leading to the most accurate predictions. We used the Bayesian optimization [34-35] cross validation search in sci-kit learn to determine the hyperparameters for each model. The BayesSearchCV functionality allows for rapid traversing of a search space to quickly determine the model with the best generalization estimate. Table 2.2 defines the parameters. All model parameters are not shown as not many diverted from default model parameters in sci-kit learn. The hyperparameters for each model are in the tables below. All models were trained to predict CGenFF atom types and partial charges. As such, Tables 2.3 and 2.4 reflect only the element groupings existing in that given force field. It is important to note that there is only one iodine atom type in CGenFF. Thus, a classification model was not generated for that element group and all query molecules typed by CGenFF based ML-MATCH will automatically be assigned that atom type.

Meaning of Parameters	
Number of Trees	number of trees in the Forest
Max Depth	depth of each tree in the forest. The deeper the tree the more information is captured. If = None, then tree traverses until pure.
Min Sample Split	minimum number of samples required to split and internal node
Max Features	number of random feature subsets to consider when splitting a node. If = None, then max features = number of features.
Bootstrap	If True, bootstrap sampled are used when building trees. If False, the entire dataset is used to build each tree.

Table 2.2: Random Forest model parameters considered for hyperparameter optimization. Parameters for which Bayesian Optimization hyperparameters diverted from default scikit learn model parameters.

Atom Type Classification Models						
Model	Number of Trees	Max Depth	Min Sample Split	Max Features	Bootstrap	Class Weight
Bromine	100	None	2	None	TRUE	None
Carbon	100	None	2	None	TRUE	None
Chlorine	100	None	2	None	TRUE	None
Fluorine	100	None	2	None	TRUE	None
Hydrogen	100	None	2	None	TRUE	None
Nitrogen	100	None	2	None	TRUE	None
Oxygen	100	None	2	None	TRUE	None
Phosphorus	100	None	2	None	TRUE	None
Sulfur	200	None	3	None	TRUE	None

Table 2.3: Bayesian optimized hyperparameters for each atom type classification model.

Model	Number of Trees	Max Depth	Min Sample Split	Max Features	Bootstrap
Bromine	100	None	2	None	FALSE
Carbon	100	None	2	None	TRUE
Chlorine	100	None	2	None	FALSE
Fluorine	100	None	6	None	TRUE
Hydrogen	100	None	2	None	TRUE
Nitrogen	200	None	3	None	TRUE
Oxygen	100	None	2	None	TRUE
Phosphorus	100	None	3	None	TRUE
Sulfur	100	None	2	None	TRUE
Iodine	100	None	2	None	TRUE

Table 2.4: Bayesian optimized hyperparameters for each partial charge regression model.

2.3.4 Matching Algorithm

The training set of a general organic force field consists of molecules and fragments curated to span the chemical space of drug-like molecules. However, due to the vastness of this space and computational constraints, these chemical moieties are normally unable to traverse such a large area in its entirety. This shortfall results in bond, angle and dihedral covalent connections, as defined by atom types, which are present in a query molecule of interest but unrepresented in a given force field training set. Thus, we must quantitatively determine the best MATCH between the unrepresented bond, angle or dihedral of the query molecule and what currently exists in the training set. With the ML-MATCHing algorithm, we seek to determine this best fit by analogy.

To quantitatively determine the parameters in a force field that are best suited to the query molecule, we begin by generating a representation of each bond, angle and torsion that exists within our training set, which can be used comparatively with the query molecule. This is accomplished by generating representative atomic fingerprints. It is important to note that when considering a training set of molecules whose parameters have been calculated to fit to a given force field, there may be numerous instances of a covalent connection between atom type i (aT_i) and atom type j (aT_j). Thus, it is imperative to calculate a representation of that bond. This calculation is conducted for all bonds, $[aT_i, aT_j]_{i=j, i \neq j}$, angles, $[aT_i, aT_j, aT_k]_{i=j=k, i \neq j \neq k}$, and torsions $[aT_i, aT_j, aT_k, aT_m]_{i=j=k=m, i \neq j \neq k \neq m}$. A schematic of how each representation is calculated is shown in Figure 4.

The first step of this process is to convert the 2D representation of all bonds, angles and dihedrals between atom types to atomic fingerprints. This is done identically to the process described previously for the generation of atomic descriptors. Secondly, we simply calculate the average atomic fingerprint representation of each covalent connection in the training set. The third step involves calculating the Euclidean distance between the average representation and all instances to determine the instance of that bond with the shortest or minimum distance from the overall average instance. This instance is used as the representative of that specific bond in the MATCHing algorithm. Calculating the representative bond also reduces the computational resources and time involved in making a MATCH as it drastically reduces the number of distance calculations needed to find the closest MATCH to the query molecule.

To quantify how different the missing moieties within the query molecule are from covalent bond groupings (bonds, angles, torsions) in each force field term and then identify the most similar MATCH, we must first determine how environmentally dissimilar all instances of a given parameter is within the training set. This is accomplished by calculating what we call a “basis score” (β_{score}). Once all instances of a particular covalent bond grouping are represented as an atomic fingerprint, we calculate the Euclidean distance between all instances and compute the standard deviation of those calculated differences, where N is the total number of instances. To normalize these distances based on the number of atoms in the covalent bond grouping, we divide this the summed standard deviation by the number of atoms in the groupings ($N_{bonds} = 2, N_{angles} = 3, N_{torsions} = 4$). The schematic of this calculation is shown in Figure 5. Basis scores for every bond, angle and dihedral present in the training set are stored in a readable table.

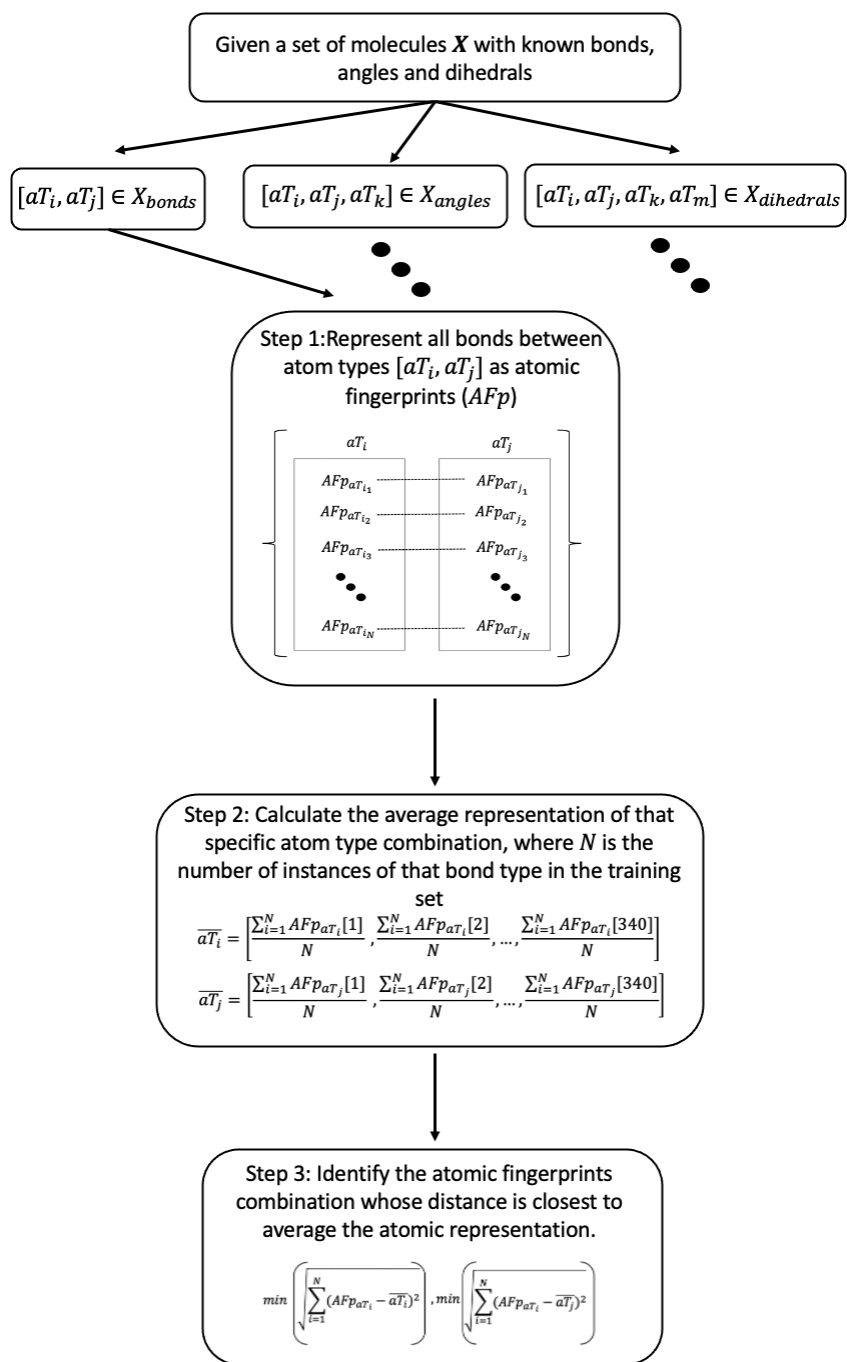


Figure 2.5: Schematic of the generation of representative bonds, angles and dihedrals for a given force field. Specifically shown for bonds, however, this quantification is the same for angles and dihedrals.

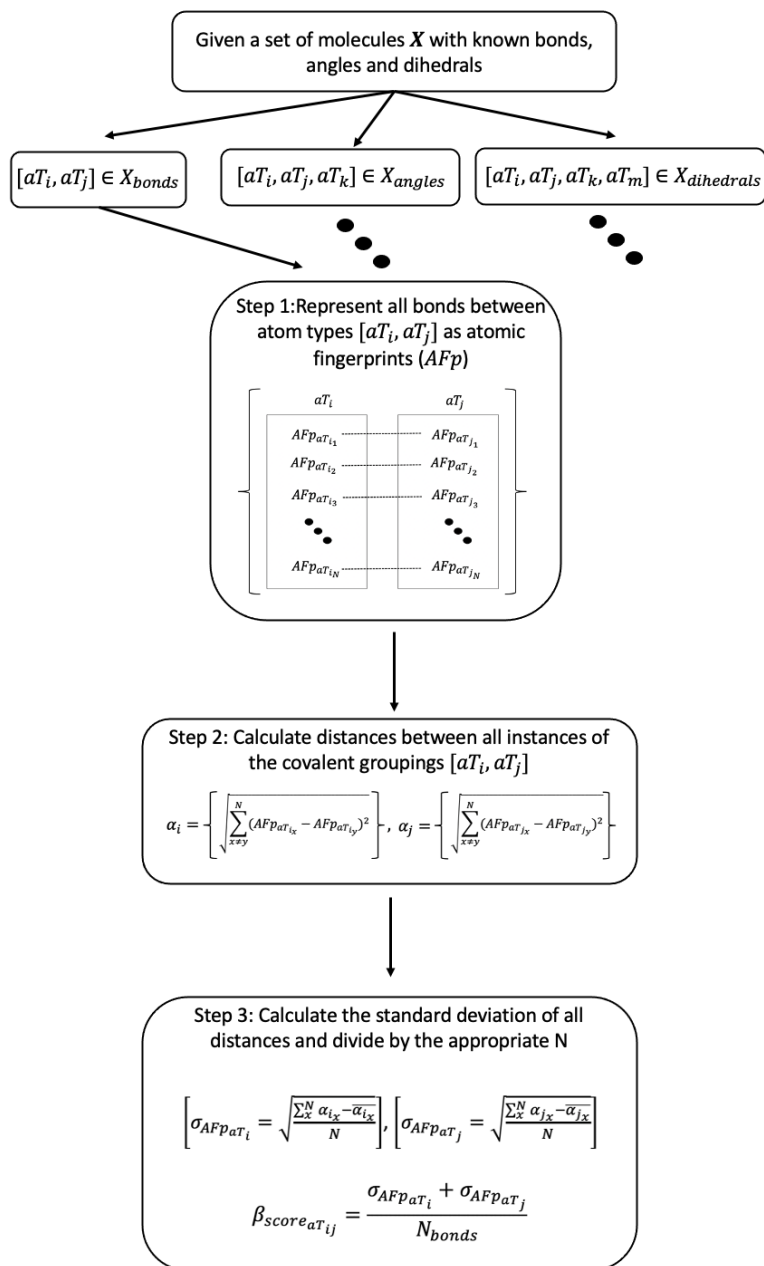


Figure 2.6: Schematic of the calculation of basis scores. Specifically shown for bonds, however, this quantification is the same for angles and dihedrals.

The final step of developing the MATCHing algorithm is the integration of a functionality for the quantification of the dissimilarity between the missing moiety in the query molecule to the existing moieties in the force field. This allows us to identify the closest MATCHed parameter. As all atoms in the query molecule are represented as atomic fingerprints and all covalent bond groupings are represented as covalent fingerprints, we simply calculate the Euclidean distance between all AFps of that missing moiety to the existing atomic fingerprints from the training set. The grouping with the smallest average distance is considered to be the closest MATCH. The existing parameters for that closest match are then assigned to the missing covalent moiety in the query molecule. In addition, we calculate the Δ_{score} , which is the computed absolute value of the difference between the overall average distance between the missing moiety and the closest MATCH and the β_{score} of the closest MATCH. To reduce the computational time, we only consider the grouping of specific elements in the training set. For example, if there does not exist a perfect MATCH between the query dihedral with atom types reflecting elements *N-C-C-N*, we only compare that dihedral to the existing *N-C-C-N* groupings in the training set.

Equation 2.9

$$\Delta_{score} = \left| \frac{\sqrt{\sum_k^N (AFp_{query_i} - AFp_k)^2} + \sqrt{\sum_k^N (AFp_{query_j} - AFp_k)^2}}{N_{bonds}} - \beta_{score_{ij}} \right|$$

, where N is the number of instances of the that particular element grouping the in representative groups of the training set.

2.4 Summary

The use of molecular mechanics force fields for computer aided drug development has Machine Learning based Atom Typer for CHARMM offers a novel framework for both the prediction of atom types and partial charges as well as an analogous matching algorithm for the assignment of bonded terms. ML-MATCH is expected to increase both the efficiency and accuracy of small molecule atom parameterization. This proposed algorithm takes advantage of the ensemble nature of the Random Forest to provide the user with a quantified confidence in the assignment of partial charges and atom types. While the MATCHing algorithm exploits the atomic fingerprint representation for the identification of similar bonds, angles and dihedrals that may be present in a query molecule but not represented in a force field's training set.

Chapter 3 will show that the ML-MATCH framework has promising applications in drug discovery and may be able to be extrapolated other fields in which accurate parameterization of small organic molecules are necessary.

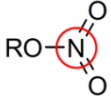
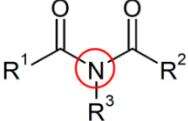
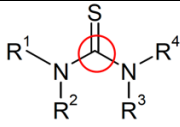
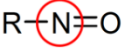
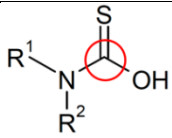
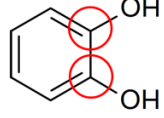
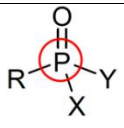
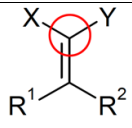
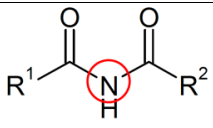
References



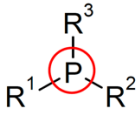
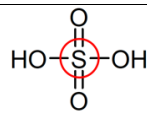
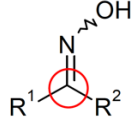
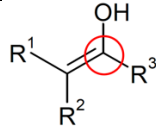
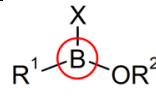
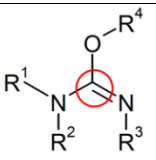
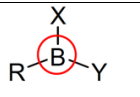
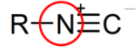
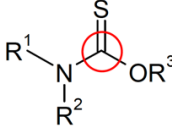
1. Sliwoski, G., Kothiwale, S., Meiler, & Lowe Jr., E.W., 2014, *Pharmacology Review*, Vol. 66, pp. 334-395.
2. Macalino, S.J.Y., Gosu, V., Hong, S., & Choi. S., 2015, *Archives of Pharamacal Research*, Vol. 38, pp. 1686–1701.
3. Vanommeslaeghe, K., Guvench, O., & Jr., MacKerell Jr., 2014, *Current Pharmaceutical Design*, Vol.20, pp. 3281-3292.
4. Shao, Q., & Zhu, W., 38, 2019, *Journal of Physical Chemistry*, Vol. 123, pp. 7974-7983.
5. Hospital, A., Goñi, J.R., Orozco, M., & Gelpí, J.L., 2015, *Advances and Applications in Bioinformatics and Chemistry* ,Vol. 8, pp. 37-47.
6. Bauschlicher, JR., C.W., & Langhoff, S.R., 1991, *Science*, Vol. 254, pp 394-398.
7. Grotendorst, J., Attig, N., Blu'gel, S., & Marx, D, 2009, *Multiscale Simulation Methods in Molecular Sciences*, Vol. 42, pp 203-214.
8. Lin, H. & Truhlar, D., 2007, *Theoretical Chemistry Accounts*, Vol. 117, pp. 185-199.
9. Wang, J., Wang, W., Kollmann, P.A., & Case, D.A. 2001, *Journal of American Chemical Society*, Vol. 222, p. U403.
10. Wang, J., Wang, W., Kollman P. A. & Case, D. A. 2006, *Journal of Molecular Graphics and Modelling*, Vol. 25, pp. 247-260.
11. D.A. Case, T.E. Cheatham, III, T. Darden, H. Gohlke, R. Luo, K.M. Merz, Jr., A. Onufriev, C. Simmerling, B. Wang & R. Woods. 2005, *Journal of Computational Chemistry*, Vol. 25, pp. 1668-1688.
12. Wang, J., Wolf, R. M., et al. 2004, *Journal of Computational Chemistry*, Vol. 25, pp. 1157-1174.
13. Yesselman, J.D., Price, D.J., Knight, J.L., & Brooks III, C.L. 2, 2012, *Journal of Computational Chemistry*, Vol. 33, pp. 189-202.
14. Vanommeslaeghe, K. & MacKerell, Jr., A.D. 2012, *Journal of Chemical Engineering and Modeling*, Vol. 52, pp. 3144-3154.
15. Brooks BR, Brooks CL III, Mackerell AD Jr, Nilsson L, Petrella RJ, Roux B, Won Y, Archontis G, Bartels C, Boresch S, Caflisch A, Caves L, Cui Q, Dinner AR, Feig M, Fischer S, Gao J, Hodoscek M, Im W, Kuczera K, Lazaridis T, Ma J, Ovchinnikov V, & Paci E, Pa. 10, 2009, *Journal of Computational Chemistry*, Vol. 30, pp. 1545-1614.
16. K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, & A. D. MacKerell Jr. 4, 2010, *Journal of Computational Chemistry*, Vol. 31, pp. 671-690.
17. Dodda, L. S.;Cabeza de Vaca, I.; Tirado-Rives, J.; Jorgensen, W. L., 2017, *Nucleic Acids Research*, Vol. 45, pp. 331-336
18. Jorgensen, W. L., & Tirado-Rives, J., 2005, *Proceedings of the National Academy of Sciences. USA* Vol. 102, pp. 6665-6670.
19. <http://open-forcefield-toolkit.readthedocs.io>
20. Unke, O.T., Chmiela, S., Sauceda, H.E., Gastegger, M., Polshvsky, I., Shutt, K.T., Tkatchenko, A., & Muller, K.R., 2021, *Chemical Reviews*

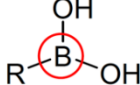

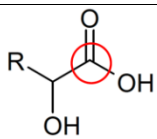
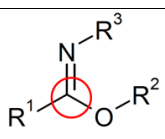
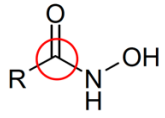
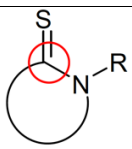
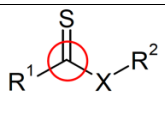
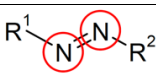
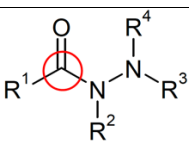
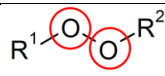
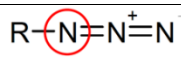
21. Sterling, T. & Irwin, J.J., 2015, *Journal of Chemical Information and Modeling*, Vol. 55, pp. 2324-2337.
22. Haider, N., 2010, *Molecules*, Vol 15, pp. 5079-5092
23. Nirmalraj, P., La Rosa, A., Thompson, D. et al., 2016, *Scientific Reports*, Vol. 6, 19009
24. Rosenbrock, C.W., Homer, E.R., Csányi, G. et al., 2017, *npj Computational Materials*, Vol. 3, pp.1-29.
25. Sandfort, F., Strieth-Kalthoff, F., Kühnemund, M., et. al., 2020, Vol.6, pp. 1379-1390.
26. Leidner, F., Yimaz, N.K., & Schiffer, C.A., 2019, *Journal of Chemical Information and Modeling*. Vol. 59, pp. 3679–3691.
27. Graziano, G., 2020, *Nature Reviews Chemistry*, Vol.4.
28. Feymann, R.P., 1939, *Physical Reviews*, Vol. 56, pp. 340-343.
29. O'Boyle, N.M., Banck, M., James, C.A. et al., 2011, *Journal of Cheminformatics*, Vol. 3
30. Breimann, L., 2001, *Random Forests*
31. Rai, Brajesh K.; Bakken, Gregory A., 2013, *Journal of Computational Chemistry*, Vol. 34, pp. 1661-1671.
32. Bleiziffer, P., Schaller, K., & Riniker, S., 2018, *Journal Chemical Information and Modeling*, Vol. 58, pp. 579–590.
33. Pedregosa et al., 2011, *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830.
34. Snoek, J., et.al., 2012, arXiv.1206.2944
35. Mockus, J., et. al., 1978, *Towards Global Optimization*, Vol. 2, pp. 117–129.

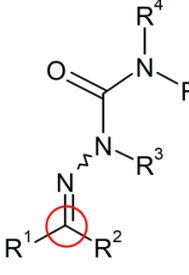
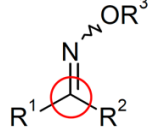
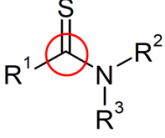
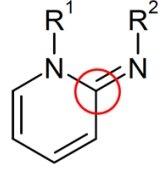
2.5 Appendix I.

Appendix I Table 1: Depicts the functional moieties that exist in ZINC15 FDA and not in the CHARMM General Force Field as defined by the checkmol software.

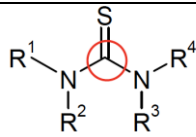
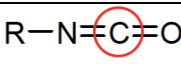
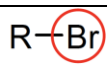
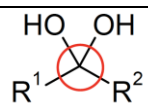
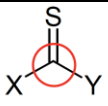
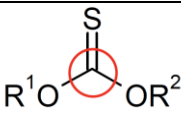
1.	nitrate	 <p>R = alkyl, aryl</p>
2.	carboxylic acid imide-N-substituted	 <p>R¹, R² = H, alkyl, aryl R³ = anything but H</p>
3.	thiourea	 <p>R¹, R², R³, R⁴ = H, alkyl, aryl</p>
4.	nitroso compound	 <p>R = alkyl, aryl</p>
5.	thiocarbamic acid	 <p>R¹, R² = H, alkyl, aryl</p>
6.	1-2-diphenol	
7.	phosphonic acid ester	 <p>R = alkyl, aryl X, Y = any O, N, Hal residue</p>
8.	ketene acetal or derivative	 <p>R¹ = H, alkyl, aryl R² = H, alkyl, aryl X = any hetero atom Y = any hetero atom</p>
9.	carboxylic acid imide-N-unsubstituted	 <p>R¹, R² = H, alkyl, aryl</p>

10.	hemithioaminal	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ $R^3 = \text{H, alkyl, aryl}$ $R^4 = \text{H, alkyl, aryl}$ $R^5 = \text{H, alkyl, aryl}$ </p>
11.	enediol	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ </p>
12.	phosphine oxide	 <p> $R^1, R^2, R^3 = \text{alkyl, aryl}$ </p>
13.	sulfuric acid	
14.	oxime	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ </p>
15.	enol	 <p> $R^1 = \text{H, acyl, alkyl, aryl}$ $R^2 = \text{H, acyl, alkyl, aryl}$ $R^3 = \text{H, acyl, alkyl, aryl}$ </p>
16.	boronic acid ester	 <p> $R^1, R^2 = \text{alkyl, aryl}$ $X = \text{any O, N, Hal residue}$ </p>
17.	isourea	 <p> $R^1, R^2, R^3, R^4 = \text{H, alkyl, aryl}$ </p>
18.	boronic acid derivative	 <p> $R = \text{alkyl, aryl}$ $X, Y = \text{any O, N, Hal residue}$ </p>
19.	isonitrile	 <p> $R = \text{alkyl, aryl}$ </p>
20.	thiocarbamic acid ester	 <p> $R^1, R^2 = \text{H, alkyl, aryl}$ $R^3 = \text{alkyl, aryl}$ </p>

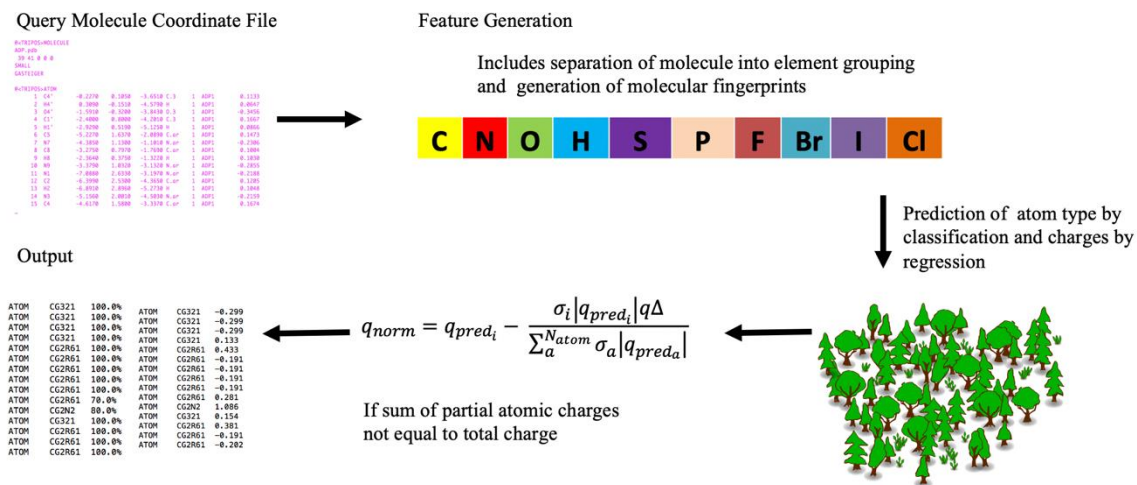
21.	boronic acid	 $\text{R}-\text{B}(\text{OH})_2$ <p>R = alkyl, aryl</p>
22.	carbonic acid derivative	 $\text{X}-\text{C}(=\text{O})-\text{Y}$ <p>X, Y = any hetero atom</p>
23.	alpha-hydroxyacid	 $\text{R}-\text{CH}(\text{OH})-\text{C}(=\text{O})\text{OH}$ <p>R = H, alkyl, aryl</p>
24.	imido ester	 $\text{R}^1-\text{C}(\text{O}-\text{R}^2)=\text{N}-\text{R}^3$ <p>R¹ = H, alkyl, aryl R² = alkyl, aryl R³ = H, alkyl, aryl</p>
25.	hydroxamic acid	 $\text{R}-\text{C}(=\text{O})\text{N}-\text{OH}$ <p>R = H, alkyl, aryl</p>
26.	thiolactam	 $\text{S}=\text{N}-\text{R}$ <p>R = H, alkyl, aryl</p>
27.	thiocarboxylic acid ester	 $\text{R}^1-\text{C}(=\text{S})-\text{X}-\text{R}^2$ <p>R¹ = H, alkyl, aryl R² = alkyl, aryl X = O, S</p>
28.	azo compound	 $\text{R}^1-\text{N}=\text{N}-\text{R}^2$ <p>R¹, R² = alkyl, aryl</p>
29.	carboxylic acid hydrazide	 $\text{R}^1-\text{C}(=\text{O})\text{N}-\text{N}(\text{R}^3)-\text{R}^2$ <p>R¹ = H, alkyl, aryl R² = H, alkyl, aryl R³ = H, alkyl, aryl R⁴ = H, alkyl, aryl</p>
30.	peroxide	 $\text{R}^1-\text{O}-\text{O}-\text{R}^2$ <p>R¹ = alkyl, aryl R² = alkyl, aryl</p>
31.	azide	 $\text{R}-\text{N}=\text{N}=\text{N}$ <p>R = alkyl, aryl</p>

32.	semicarbazone	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ $R^3 = \text{H, alkyl, aryl}$ $R^4 = \text{H, alkyl, aryl}$ $R^5 = \text{H, alkyl, aryl}$ </p>
33.	oxime ether	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ $R^3 = \text{alkyl, aryl}$ </p>
34.	thiocarboxylic acid amide	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ $R^3 = \text{H, alkyl, aryl}$ </p>
35.	imino(het)arene	 <p> $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$ </p>

Appendix I Table 2: Depicts the functional moieties that exist in the CHARMM General Force Field and not in ZINC15 FDA as define by the checkmol software.

1.	isothiurea	 $R^1, R^2, R^3, R^4 = \text{H, alkyl, aryl}$
2.	isocyanate	 $R = \text{alkyl, aryl}$
3.	alkyl bromide	 $R = \text{alkyl}$
4.	carbonyl hydrate	 $R^1 = \text{H, alkyl, aryl}$ $R^2 = \text{H, alkyl, aryl}$
5.	thiocarbonic acid derivative	 $X, Y = \text{any hetero atom}$
6.	thiocarbonic acid diester	 $R^1, R^2 = \text{alkyl, aryl}$

Appendix I Figure 1: Charge renormalization workflow for query molecule parameterized with ML-MATCH.



Chapter 3

Machine Learning Multipurpose Atom Typer for CHARMM

Results and Application

Murchtricia Jones and Charles L. Brooks III

3.1 Selection of Machine Learning Algorithm

The ML-MATCH framework is a two-part system. The first partition is the machine learning engine developed for the prediction of atom types and partial charges for each atom in a query molecule. To this end, we examined simple machine learning algorithms to test how well they could predict both the atom types and partial charges from a general description of the local atomic environments. We incorporated all molecules in the CGenFF basis set (500). All atoms were encoded using the newly developed atomic fingerprint described in Chapter 2.

To determine the algorithm which best captured the relationship between the local atomic environment and atom types/partial charges, we employed sci-kit learn. We separated all atomic fingerprints based on element grouping and developed separate classification, for the assignment of atom types, and regression, for the assignment of partial charges, algorithms for each grouping, respectively. We used 70% of the dataset for training and optimization of our models and 30% for testing and validation. We tested three simple

and well vetted algorithms for this task, Naïve Bayes (classification) [1-2], Bayesian Regression [2], K- Nearest Neighbors [3] and Random Forests [4]. In the following, we describe the methodologies and results for each algorithm tested.

3.1.2 Naïve Bayes Methodology and Results for Assignment of Atom Types

Naïve Bayes classifiers are considered to be naïve due to the working assumption that the features are independent of a given class. This is shown in Equation 3.1, where $\mathbf{X} = (X_1, \dots, X_n)$ is a feature vector and C which is a class. When applied to this research question, \mathbf{X} is the atomic fingerprint where n is a natural number from 1 – 340 which is the length of each feature vector and C is the associated atom type as defined by CGenFF. This methodological explanation was adapted from Rish [1].

Equation 3.1

$$P(\mathbf{X}|C) = \prod_{i=1}^n P(X_i|C)$$

Although the assumption of independence is naïve, this classifier has been successful in the fields of medical diagnosis, text classification and drug target identification. Due to this shown success in diverse fields, we believed that it could be well suited for our research question. The basis of this algorithm is Bayes theorem. With this theorem we can find the probability of event A occurring given that B has occurred, where A is the hypothesis and B is the evidence or known information and in this, we assume that the presence of one feature does not affect another.

Equation 3.2

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Given that our atomic fingerprint is represented as $\mathbf{X} = (X_1, \dots, X_n)$, where each feature takes values from its domain D_i . The set of all AFps is denoted as $\omega = D_1 \times \dots \times D_n$ and C is an unobserved random variable which represents the class of an example. C can take the value of any m value where $C \in \{0, \dots, m - 1\}$. This algorithm uses a functional mapping of $h: \omega \rightarrow \{0, \dots, m - 1\}$ where the $h(\mathbf{x}) = C$ would always assign the same atom type, C , to a given example, \mathbf{x} . Essentially, the idea is to associated a given class, $C = i$, using a discriminant function $f_i(\mathbf{x})$ where $i \in \{0, \dots, m - 1\}$ and the classifier selects the class with the maximum value of the discriminant function on a given example.

Equation 3.3

$$h(x) = \arg \max_{i \in \{0, \dots, m-1\}} f_i(\mathbf{x})$$

Thus, the Bayes a classifier $h^*(\mathbf{x})$ uses a discriminant function to calculate the posterior probability of C given a feature vector defined as $f^*(\mathbf{x}) = P(C = i | \mathbf{X} = \mathbf{x})$. When the Bayes theorem is applied it gives the equation below.

Equation 3.4

$$P(C = i | \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} | C = i) P(C = i)}{P(\mathbf{X} = \mathbf{x})}$$

Where $P(X = x)$ is the same for all classes so the Bayes classifier is given as,

Equation 3.5

$$h^*(x) = \arg \max_{i \in \{0, \dots, m-1\}} P(X = x | C = i) P(C = i)$$

However, quantifying $P(\mathbf{X} = \mathbf{x} | C = i)$ becomes increasingly difficult with high dimensional data. So, we must use an approximate by assuming all atomic fingerprints are independents from the given atom type. Thus, the discriminant function becomes,

Equation 3.6

$$f_i^{NB}(x) = \prod_{j=1}^n P(X_j = x_j | C = i) P(C = i)$$

similar, to what we see in Equation 3.1.

Specifically, in sci-kit learn, we used the GaussianNB() functionality which has been developed to extend this methodology to real-valued attributes [5-6]. In this implementation the likelihood of features is calculated as,

Equation 3.7

$$P(X_j = x_j | C = i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_j - \mu_i)^2}{2\sigma_i^2}\right)$$

To determine how well this method approximates the relationship between atomic environment and atom type, we calculated the accuracy score. We found that in some element groupings Naïve Bayes performed well, specifically in bromine, chlorine, fluorine and iodine. However, these were not trustworthy results for a few reasons. The first being that these particular groups have the lowest representation as shown in Section 2.2.3. This low sample count caused overfitting. In addition, the low performance for the carbon, hydrogen, nitrogen and sulfur groupings indicated that this method has difficulty discerning atom types from one another.

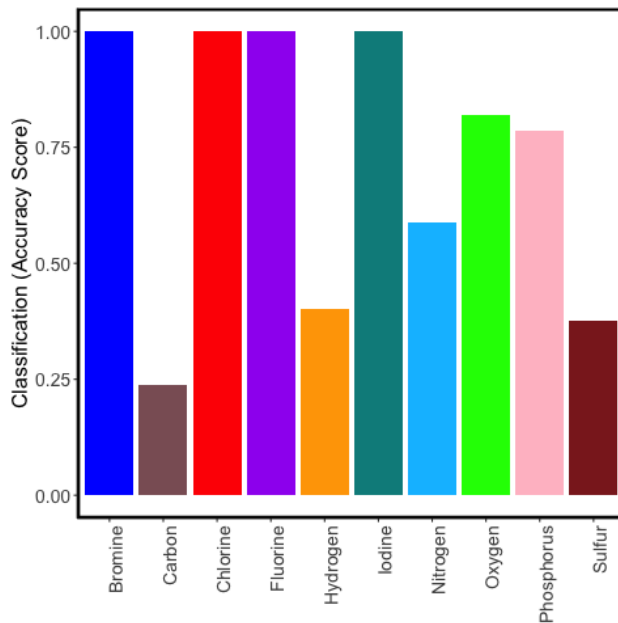


Figure 3.1: Naïve Bayes results on training set. Naïve Bayes algorithm shows varied performance across the element groupings.

These results were not promising and shows that the application of this algorithm for this task is not sufficient for determining the relationship between atomic environment and atom types for this training set.

3.1.3 Bayes Regression Methodology and Results for Assignment of Partial Charges

Bayes regression is very similar to Bayes classification at its foundation as both are based on the Bayes Theorem shown in Equation 3.2. Bayesian regression assumes both the parameter set, β , and samples, \mathbf{X} , are from a Gaussian distribution.

Equation 3.8

$$C \sim N(\beta\mathbf{X}, \sigma^2)$$

C is generated from the Gaussian distribution which is characterized by the mean and variance. β is the regression coefficient matrix and the variance is calculated as the standard variation squared. The posterior probability is given by,

Equation 3.9

$$P(\beta|C, X) = \frac{P(C|\beta, X) P(\beta, X)}{\int P(C, X|\beta_i) d\beta_i}$$

where $P(C|\beta, X)$ is the likelihood of data, $P(\beta, X)$ is the prior probability of parameters and the denominator is the marginal probability. Figure 3.2 shows the results of this regression methodology as defined by the RMSE metric for model analysis. The average RMSE across all groupings is 0.05. This is not an acceptable RMSE when compared to the RMSE of 0.008 for the ParamChem atom parameterization software [8] from Vanommeslaeghe et. al.

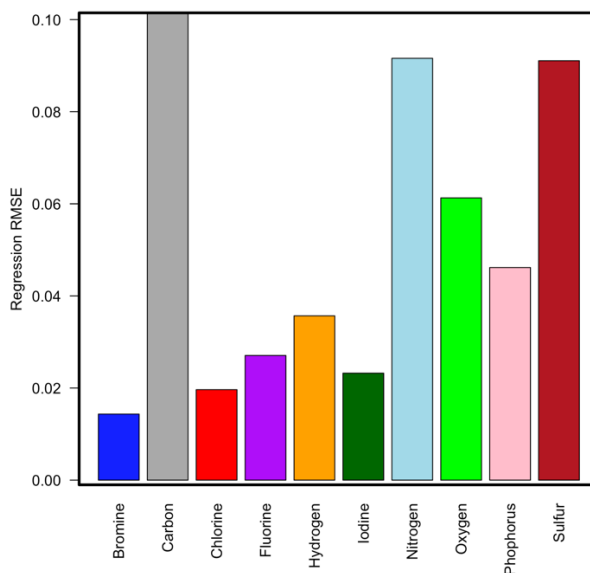


Figure 3.2: Bayesian regression results for the prediction of partial charges.

3.1.4 K-Nearest Neighbors Methodology and Results for Assignment of Atom Types and Partial Charges

K-Nearest Neighbors (KNN) is a non-probabilistic classification procedure. The basis of this method is the assumption that observations that are closest to another would have the same classification or similar attributes. In this project we measured nearness using the Euclidean distance between samples in $\mathbf{X} = (X_1, \dots, X_n)$. This distance between atomic fingerprints X_i and X_j is computed as,

Equation 3.10

$$d(X_i, X_j) = \sqrt{(X_{i_1} - X_{j_1})^2 + \dots + (X_{i_n} - X_{j_n})^2}$$

This model is dissimilar to others in that the training procedure is not based on determining the relationship between atomic environment and atom type/partial charge but consists of only storing the atomic fingerprints and labels. The algorithm then uses the distance formula to determine the closest neighbors to the query atom. In classification, the atom type is selected by plurality vote of its neighbors; meaning that the class assigned is the one which is most common among its neighbors. In regression, the partial charge is calculated as the average partial charge of its neighbors. Although this model is simple in nature, we found it to have good performance for our dataset.

Figure 3.3(a) shows the accuracy scores for the KNN Classifier for atom type assignment. The average accuracy score is 99.6% which depicts very high performance in relating atomic environment to atom type. While Figure 3.3(b) depicts the regression models RMSE with an average RMSE of 0.0165 which is large improvement from the Bayes regression model.

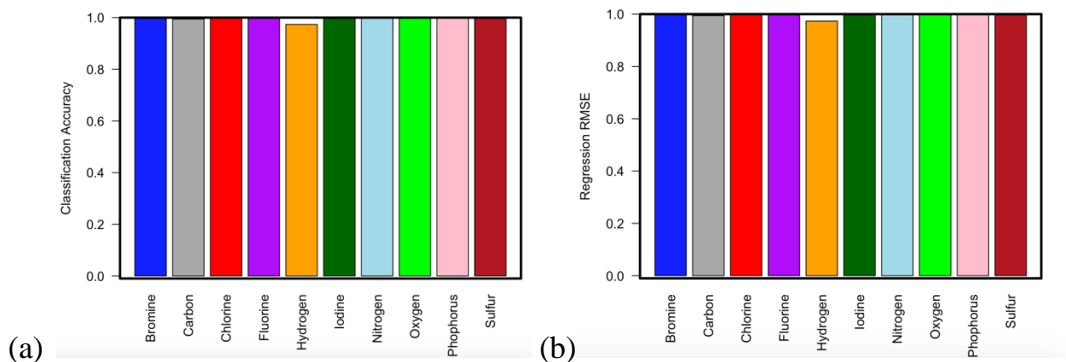


Figure 3.3: KNN training set results. (a) KNN model accuracy scores for all atom type prediction models. (b) KNN model RMSE for all partial charge prediction models.

3.1.5 Random Forest Methodology and Results for Assignment of Atom Types and Partial Charges

Decision trees are tree-like models which are flowchart-like in structure. A random forest is a combination of decision trees which depend on random vectors that are independently sampled. In classification, the ensemble of trees vote for the most popular class while in regression tasks that the average of the decisions is the output. For a given ensemble of tree-like classifiers, $h_1(X_i), \dots, h_K(X_i)$, where K is the number of trees in the forest. The training set is drawn at random from the distribution of the random vector \mathbf{X} , C and the margin function is defined as

Equation 3.11

$$mg(\mathbf{X}, C) = av_k I(h_k(\mathbf{X}) = C) - \max_{j \neq C} av_k I(h_k(\mathbf{X}) = j)$$

The margin, $mg(\mathbf{X}, C)$, measures the extent to which the average number for votes at (\mathbf{X}, C) for the correct class exceeds the average vote for any other class. $I(\cdot)$ is the indicator function. The larger the margin the more confidence the ensemble has in the classification. Random Forests for regression are formed in a similar manner where the

output values are numerical rather than class labels. A random forest regressor is formed by taking the average of partial charge prediction over k trees of the forest.

Equation 3.12

$$C = \frac{1}{K} \sum_{i=1}^K h_i(\mathbf{X})$$

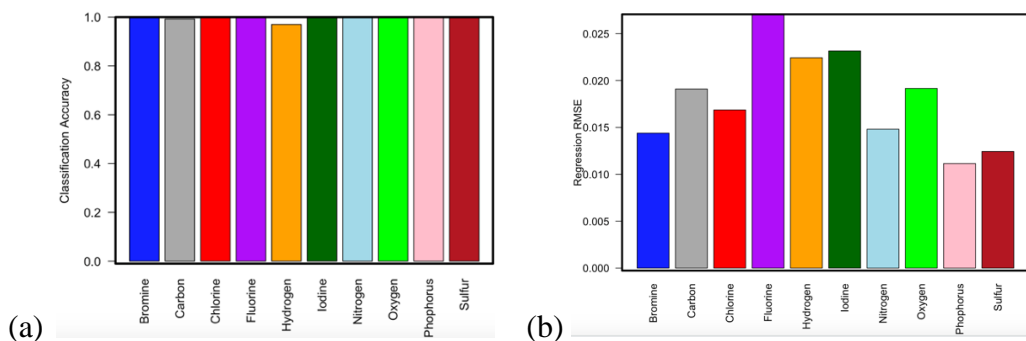


Figure 3.4: Random Forest training set results. (a) Random Forest model accuracy scores for all atom type prediction models. (b) Random Forest model RMSE for all partial charge prediction models.

The Random Forest models have similar performance to the KNN models with an average classification accuracy of 99.6% and an average partial charge regression RMSE of 0.018.

Based on these findings, we decided to go forward with Random Forest as there are more tuning parameters for this algorithm and as such, we would have a greater space to traverse for model improvement. The optimization of the random forest models can be found in Section 2.3.5.

3.2 Algorithm Performance on Test Set

The Random Forest-derived models were tested using 30% of the CGenFF AFps for each respective element grouping. Atom type assignment is the first step of parameterization. It is important to note that atom types vary with the force field. The following data is for the prediction of atom types specifically for the CHARMM General Force Field. These results are a first step in determining whether the ML-MATCH algorithm is effective in capturing such atomic characteristics. Using a Random Forest classification model, we generated 9 models for atom type assignment. These models exclude iodine atom type assignment since only 1 such atom type exists in CGenFF. Figure 3.5 shows correlation matrices for each model based in the test set. For the classification of each atom type, we have an average of 96% accuracy. Our lowest accuracy is in the H model. This was expected due to the nature of CGenFF because different H atom types are assigned to environments that are quite similar to each other. As a result, the AFps that extends to the second nearest neighbors do not span a large enough space to capture these differences. We note, however, that the charges and van der Waals radii, as well as intramolecular force constants are not largely varying within this atom type either. In creating these models, we had to balance the length of the AFps, as too much information may cause overfitting, with model accuracy.

In each correlation matrix, we have drawn red horizontal and vertical lines which group similar atom type environments together. This allows us to depict that in areas where we see misclassification, that event is minimal in that they normally reside in that boxed area. In addition, if the misclassification occurs between similar atom types, we expect that the bonded parameters will also be similar based on the similarities in atomic environment.

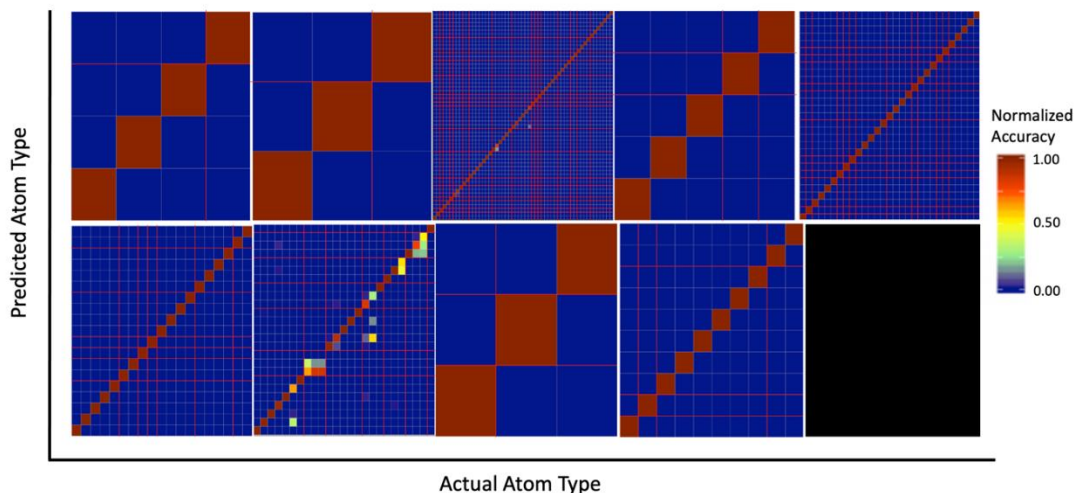


Figure 3.5: Testing results for Random Forest classification models. Correlation matrices for each atom type assignment model in CGenFF based ML-MATCH. (Top: Br, Cl, C, F, N Bottom: O, H, P, S).

The second step of parameterization is charge assignment. We have created 9 random forest regression models for the prediction of partial atomic charges (e^-), again excluding iodine atoms. The result for charge assignment is shown in Figure 7. Charges are calculated independently from atom types. For the CGenFF test set based models, the average Pearson R-value and average RMSE of charge assignment is 0.974 and 0.028e. respectively.

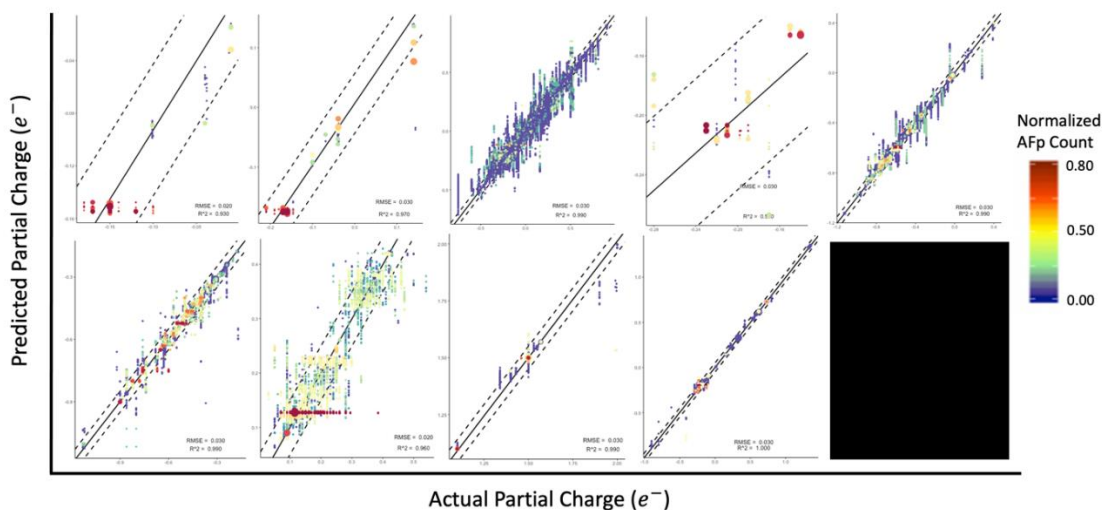


Figure 3.6: Testing results for Random Forest regression models. Graphs for each charge assignment model in CGenFF based ML-MATCH. (Top: Br, Cl, C, F, N Bottom: O, H, P, S). The solid black line is $x=y$ while the two bordering dashed lines represent $\pm 0.05e^-$.

3.3 Validation Test Set

The Free Solvation Database [5] contains 642 drug-like molecules and fragments. It is described as a *curated database of experimental and calculated experimental hydration free energies for small neutral molecules in water*. The solvation of small molecules is of particular importance because macromolecule and ligand binding interactions typically involve a partial transfer of the ligand from solution to the binding site. The ability to accurately model the hydration and dehydration of small molecules suggests the level of precision that one may expect under ideal conditions in a binding free energy calculation. More thoroughly, one can not expect to find higher accuracy in binding free energy calculations compared to what one calculates for hydration free energies. Thus, we argue that this database is a good validation set for both parameterization and accuracy of force field as it provides a set of molecules that traverse a large chemical space, shown in Figure

6, and contains the experimental hydration free energies needed to test the force field parameters produced by ML-MATCH.

3.3 ML-MATCH Models in Comparison to ParamChem Model

As a method to determine the accuracy of the underlying Random Forest Algorithms in ML-MATCH, we compare our models to ParamChem [6], which is described in Section 1.5.3. ParamChem is currently the gold standard for atom parameterization trained with the CHARMM General Force Field. For all FreeSolve molecules, we used both ML-MATCH and ParamChem to generate the atom types and partial charges for each molecule. Overall, we found an average RMSE of 0.0494e for partial charge prediction between models and an accuracy score of 90.3% for the assignment of atom types compared with results from ParamChem. These results are very promising for ML-MATCH performance. In addition, within the FreeSolve paper, examples were given to show the large chemical space for which the database spans. Below are those specific examples and the calculated accuracy score and RMSE for model comparison. As shown in Table 3.1, we see varying correlation across molecules. It is also important to note the atom type classification model performance in ML-MATCH does not directly correlate with the partial charge regression model results.

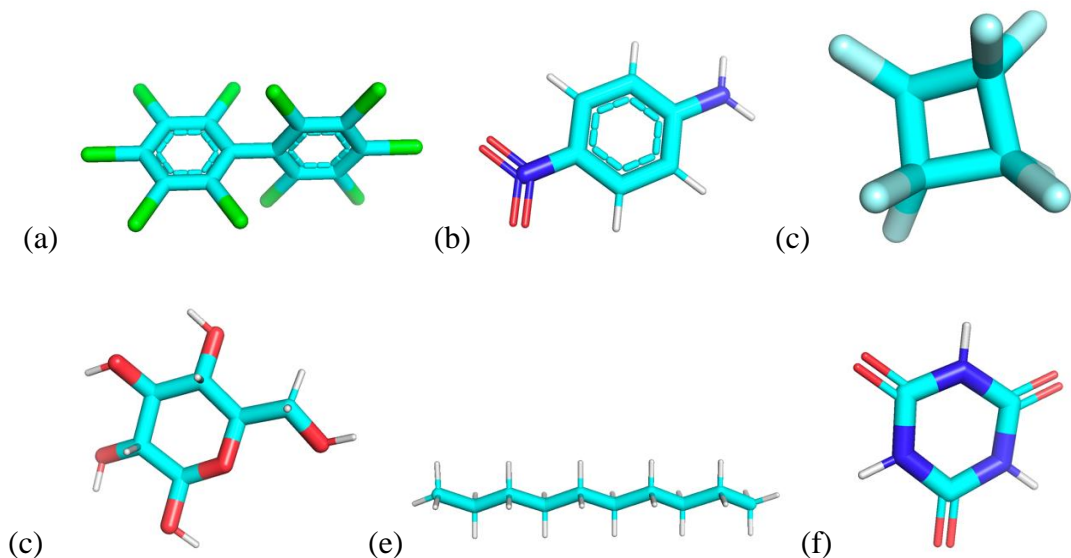


Figure 3.7: Six FreeSolve molecules that depict the span of the FreeSolve chemical space. (a) 1,2,3,4,5-pentachloro-6-(2,3,4,5,6-pentachlorophenyl)benzene, (b) 4-nitroaniline, (c) 1,1,2,2,3,3,4,4-octafluorocyclobutane, (d) (2R,3R,4S,5S,6R)-6-(hydroxymethyl)tetrahydropyran-2,3,4,5-tetrol, (e) decane and (f) 1,3,5-triazinane-2,4,6-trione. Table attached contains the average RMSE and accuracy metrics comparing ML-MATCH and ParamChem atom types and partial charges.

	Chemical	Average RMSE ML-MATCH vs ParamChem	Average Accuracy ML-MATCH vs ParamChem
(a)	1,2,3,4,5-pentachloro-6-(2,3,4,5,6-pentachlorophenyl)benzene	0.203607561	0.4375
(b)	4-nitroaniline	0.294438382	1
(c)	1,1,2,2,3,3,4,4-octafluorocyclobutane	0.101294641	0.333333333
(d)	(2R,3R,4S,5S,6R)-6-(hydroxymethyl)tetrahydropyran-2,3,4,5-tetrol	0.0368448	1
(e)	decane	0.000685056	1
(f)	1,3,5-triazinane-2,4,6-trione	0.227667656	0.25

Table 3.1: ML-MATCH vs ParamChem results for 6 FreeSolve molecules. Shows the average RMSE and accuracy metrics comparing ML-MATCH and ParamChem atom types and partial charges.

3.4 Free Energy of Hydration Calculations

The results in Section 3.3 show that for the FreeSolve database, we find an RMSE of 0.0494e for the assignment of partial charges and an accuracy of 90.3% for the prediction of atom types. Coupled with an average Pearson R-value of 0.974 and average RMSE 0.028 across ML-MATCH models for the testing set taken from CGenFF for which ParamChem is trained, we expect to see similar results between ML-MATCH and ParamChem in simulation. To test this, we extracted all benzene derivatives from the FreeSolve database as an effort to more efficiently identify those chemical moieties for which ML-MATCH may produce parameters which are insufficient for reproducing experimental data in simulation. This was done due to ML-MATCH predicted atom types and partial charges are nearly identical to those produced by ParamChem for benzene. Thus, benzene derivatives give us a good common core substructure such that we can identify moieties to which it is bound that may lead to inaccurate simulation when using ML-MATCH or ParamChem for small molecule parameterization. We used exactly 60 molecules from the FreeSolve database. Relative hydration free energies were calculated via FACTS and GBMV2 implicit solvation models, whose results were then compared to experimental data reported in the database. The Mobley database has been used as a standard for benchmarking force fields and various hydration free energy prediction methods since many of the functional groups present in these molecules are relevant for drug design purposes and are representative of drug-like chemical space [9-11].

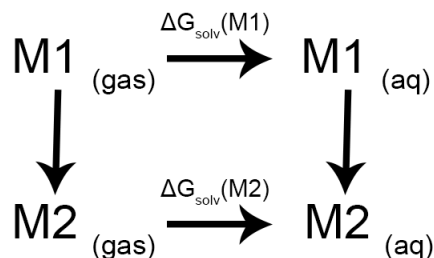


Figure 3.8: Depiction of the thermodynamic cycle for solvation free energy calculations.

The relative free energies of hydration were calculated using two different parameterization schemes independently of one another: the novel ML-MATCH presented herein and ParamChem, both of which are based on CGenFF. For the FACTS simulations, the “lone pair” charges, associated with halogens in aromatic rings, which both ML-MATCH and ParamChem output in their topology files, were reincorporated to their respective halogens. This was done because the FACTS module in CHARMM does not yet support the inclusion of lone pairs.

Molecular dynamics simulations were performed using the CHARMM molecular dynamics package, developmental version 45a2.[13], [14] All atoms were coupled to a Langevin heat bath and maintained at a temperature of 298K with a frictional coefficient of 10ps^{-1} . Trajectories for each molecule over a 10.5 ns period using a 1.5 fs time step were generated for each molecule in vacuum, GBMV2, and FACTS environments, of which the first 1.5 ns were used as equilibration and therefore not used during the free energy calculations. The SHAKE algorithm was used to constrain hydrogen bond lengths.[18] Electrostatic and van der Waals interactions were switched off between 10\AA and 12\AA .

The GBMV2 implicit solvent simulations used Still's geometric cross-term and spherical polar integration grid with 5 phi angles. The FACTS implicit solvent model used all default parameters, except for a nonpolar surface tension coefficient, $\gamma=0.015$ kcal/ (mol Å²). All other specifications for these implicit solvent models were as described in the paper by Knight & Brooks[12].

The FastMBAR solver [13] was used to calculate hydration free energies as the molecule is transferred from a gas into an implicitly hydrated state. The solver input was a total of 3000 energy difference values that were calculated for each molecule for each state (vacuum and implicit water medium). The relative hydration free energies were then centered about the experimental mean and therefore converted to free energy values, as specified in the paper by Wang, et al.[14]

3.4.1 Overall Free Energy of Hydration Calculations Results

For this particular dataset, the results summarized in Table 3.2 show that ML-MATCH generally achieves better agreement with experiment and significant improvement from ParamChem. For both implicit solvent models, ML-MATCH outperforms ParamChem regarding linearity (Pearson coefficient) and ranking (Spearman coefficient) of the relative hydration as compared to experiment. This is very surprising given that both parameterization schemes use the same underlying force field and that the correlation between ML-MATCH and ParamChem for the assignment of atom types and partial charges is very high, as evidenced by the Pearson and Spearman correlation coefficients comparing both sets of results to each other.

Where ParamChem and ML-MATCH differ the most with respect to each other is their mean unsigned difference (MUD) and root mean square difference (RMSD). Therefore, while agreement with experiment, as measured by MUE and RMSE, is slightly better for ParamChem in most cases (yet still within 0.5kcal/mol from each other), ML-MATCH can predict better free energies for a molecule in comparison to another. This is especially useful in a prospective binding affinity study, for example, where limited experimental data is available for only a few hits (so agreement with experiment takes second priority) and the goal is to increase the potency relative to already identified hits.

The fact that two different implicit solvation models yielded comparable overall statistics for ML-MATCH compared to ParamChem reinforces the claim that ML-MATCH is well-suited as a parameterization engine for CGenFF and can make comparable assignment decisions that are not solely based on legacy rules and conditions. Thus, we find that we have generated a machine learning based framework which enables the learning of underlying force field rules and assumptions for arbitrary but consistently parameterized molecules. The next two sections offer a more thorough explanation of these findings.

	GBMV			FACTS		
	MLM/ Exp.	ParamChem /Exp.	MLM /ParamChem	MLM/Exp	ParamChem /Exp.	MLM /ParamChem
Pearson	0.7223	0.4635	0.7794	0.7409	0.5979	0.8892
MUE	1.1352	1.1140	0.8879	2.2093	2.2155	1.0218
Spearman	0.7296	0.5881	0.8417	0.7454	0.6474	0.9306
RMSD	1.0655	0.9883	1.5864	2.7363	2.9544	1.7014

Table 3.2: Free energy of hydration results for GBMV2 and FACTS models.

3.4.2 GBMV2 Free Energy of Hydration Results

When comparing the experimental errors between both parameterization schemes, we find that ML-MATCH and ParamChem produce comparable results. Particularly, we find that ML-MATCH shows greatest improvement for trifluoromethyl containing molecules when compared to ParamChem, by greater than 4 kcal/mol. ML-MATCH also produced a significantly greater Pearson R-value of 0.72 compared to ParamChem's 0.46 when compared to experimental free energy of hydration values. While ML-MATCH does not improve MUE or RMSE statistics for GBMV2 relative hydration free energies, it does significantly improve the ability to rank these compounds closer to the experimental ranking, as evidenced by the increase in the Spearman coefficient value for these compounds from 0.59 to 0.73.

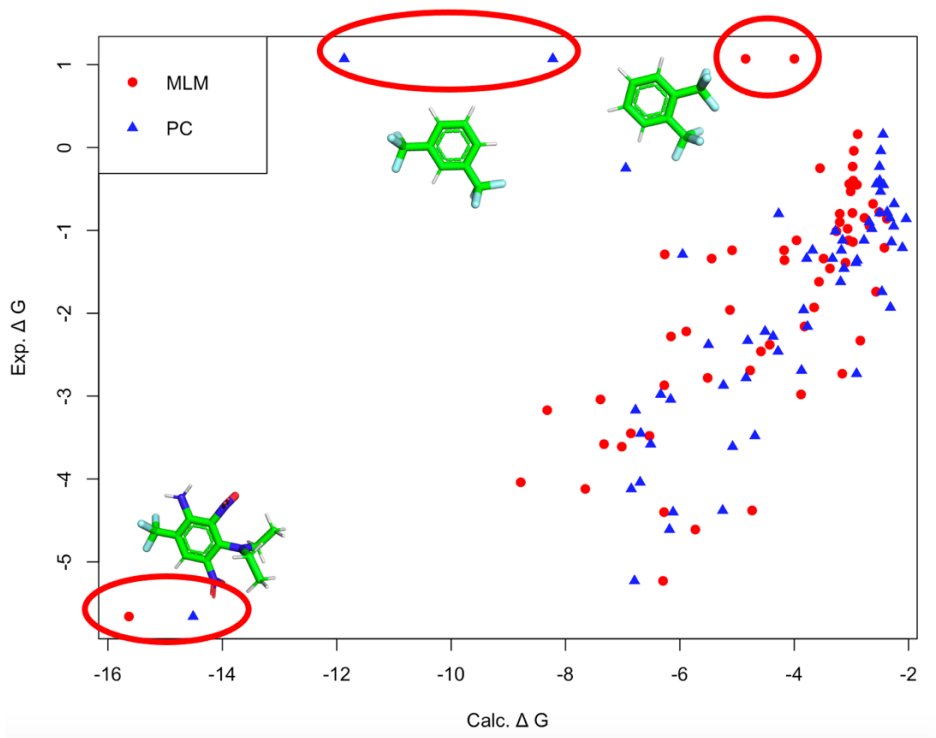


Figure 3.9: GBMV2 model free energy of hydration calculations result. Red circles are ML-MATCH vs Experimental free energy of hydration. Blue triangles are ParamChem vs Experimental free energy of hydration. Highlighted are the trifluoromethyl containing molecules for which ML-MATCH performs well.

As shown in Figure 3.9, one could consider the trifluoromethyl containing molecules to be outliers for which ML-MATCH performs considerably better over ParamChem. To dig deeper into these results, we extracted those molecules and recalculated the comparison metrics. Once the trifluoromethyl containing molecules are removed we see that ML-MATCH has much better agreement with ParamChem with ParamChem slightly outperforming ML-MATCH on average for these 57 molecules. This yielded a much more expected outcome for this comparison exercise. In the metrics of linearity and ranking, we see very comparable results while ParamChem outperforms ML-MATCH in RMSE.

	GBMV		
	MLM/Exp.	ParamChem/ Exp.	MLM/ ParamChem
Pearson	0.7759	0.8285	0.9260
MUE	1.0066	0.7442	0.5746
Spearman	0.8032	0.8346	0.8934
RMSD	1.4671	1.2081	0.8491

Table 3.3: Free energy of hydration results for GBMV2 with removal of trifluoromethyl containing molecules.

There are 5 out of 60 molecules for which ML-MATCH performs worst than ParamChem by greater than 1 kcal/mol compared to experiment. Structures are shown in Figure 3.10. To identify the source of this disparity with experiment the Pearson R-value and RMSE for the charges were calculated. We see that for molecules (a) and (c) we have very good correlation between ML-MATCH and ParamChem produced charges as shown in Figure 3.10. Thus, the difference in computed solvation free energies may be attributed to the dissimilarity in the bonded parameters defined by each parameterization scheme. In Figure 3.9, we see that the predicted atom types for molecules (a) and (c) have a one-to-one correlation between ML-MATCH and ParamChem. In Appendix II Figure 3, we see that the greatest deviation in parameters lie in the dihedrals of molecule (a), although not shown, we see the same for molecules (c). In Table 3.4 we see that while this deviation exists, the difference in the computed solvation free energy between the parametrization schemes lies around 1.2-1.3 kcal/mol which is very close to the acceptable 1 kcal/mol difference. For molecules (b), (d) and (e), all which contain a benzene bound to many chlorine atoms, we see very poor partial charge correlation as shown in Figure 3.11. This

may be due to the charge renormalization scheme the ML-MATCH uses to incorporate lone pairs. This is an area of ML-MATCH that needs further optimization.

	Pearson R Value	RMSE	Abs. Diff of $\Delta\Delta G$ (kcal/mol)
(a) trimethoxymethylbenzene	0.9818	0.0353	1.3782
(b) 1,2,3,4,5,6-hexachlorobenzene	0.5174	0.1451	2.1940
(c) diethoxymethoxybenzene	0.9698	0.0461	1.2083
(d) 1,2,3,4-tetrachloro-5-(2,3,4,6-tetrachlorophenyl)benzene	0.0271	0.1890	1.1606
(e) 1,2,4,5-tetrachloro-3-(3,4-dichlorophenyl)benzene	0.3687	0.1459	1.2247

Table 3.4: GBMV2 model results for ML-MATCH and ParamChem. ML-MATCH vs ParamChem charge comparison for those molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem with GBMV2 model.

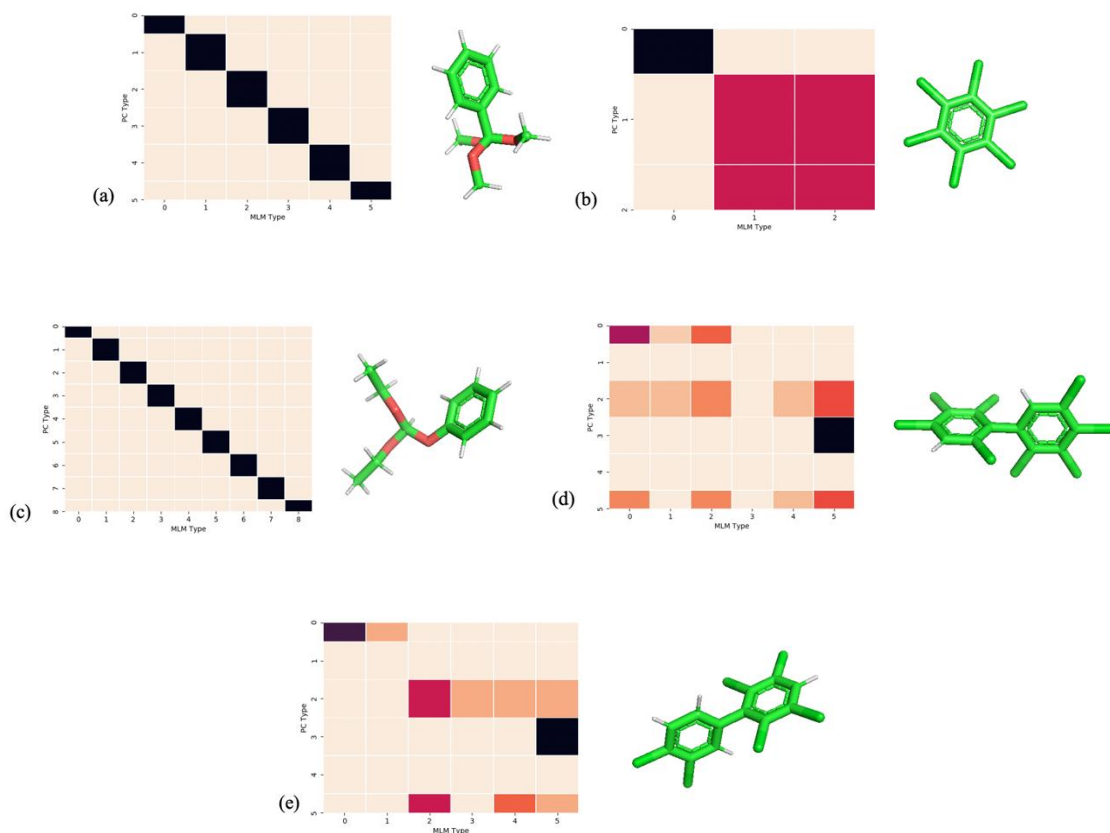


Figure 3.10: Atom types correlation matrix for FreeSolve molecules with differentially calculated FEHs by GBMV2. Molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem given by the GBMV2 model. The accuracy between the schemes for each molecule is (a) 1.00, (b) 0.666, (c) 1.00, (d) 0.40, (e) 0.50.

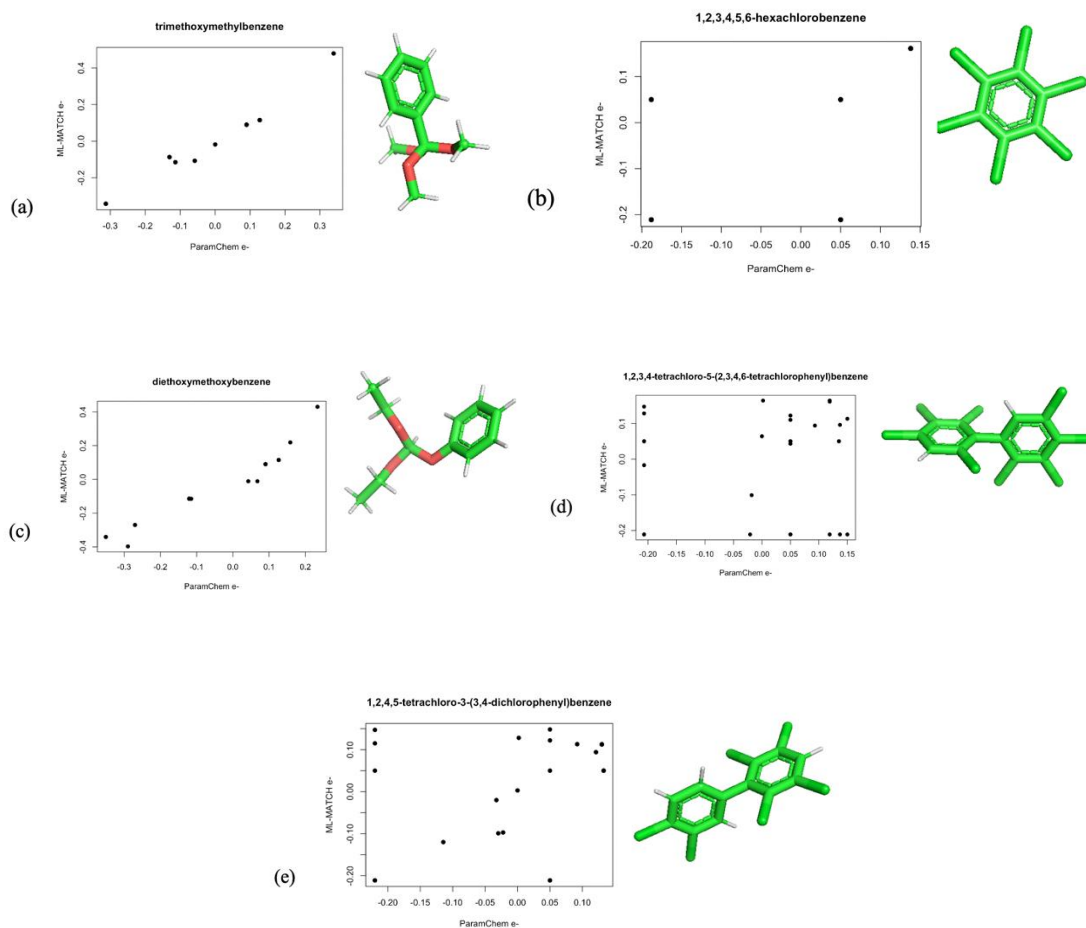


Figure 3.11: Partial charge comparison for FreeSolve molecules with differentially calculated FEHs by GBMV2. Partial charge comparison for those molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem as defined by the GBMV model.

For molecules (b), (d) and (e) we also see poor correlation in atom types between both schemes similarly shown for partial charges. We reason for the deviation between calculated solvation free energies is these compounding factors. It is important to note that within the CHARMM General Force Field these particular moieties are not well

represented. Thus, slightly poorer performance for such moieties is expected when comparing ML-MATCH to ParamChem, which is explicitly programmed to reproduce CGenFF parameters. However, we still see deviations very close to 1 kcal/mol for molecules (d) and (e). We see much further deviation for molecules (b).

3.4.2 FACTS Free Energy of Hydration Calculations Results

Using the FACTS model, we see similar results. Just as with GBMV, with the FACTS model we see better performance on trifluoromethyl containing molecules when compared to experiment. We also see significant improvement (greater than 1 kcal/mol) for ML-MATCH over ParamChem for multiple molecules with heavily chlorinated benzenes and benzyl bromide. Unlike with the GBMV2 results, ML-MATCH does improve results from those of ParamChem with respect to experiment by more than 1 kcal/mol for 7 compounds – those containing trifluoromethyl groups, benzyl bromide and two heavily chlorinated biphenyl derivatives which were predicted with worse accuracy for the GBMV2 calculations using ML-MATCH. This suggests that there are some intrinsic differences between the solvation models. However, it may also mean that incorporating the “lone pair” (a restrained point charge to halogens in aromatic rings) charge into their respective halogens may be introducing a form of systematic error for heavily chlorinated molecules that becomes more evident with these compounds, since GBMV2 simulations did not involve this charge redistribution. As shown in Table 3.2, the superior ranking ability of ML-MATCH over ParamChem that was observed for the GBMV2 results are retained for the FACTS calculations, where significant improvement in Spearman coefficients was

observed from 0.60 for ParamChem to 0.78 for ML-MATCH. Increased linearity compared to experiment was also observed, from 0.54 for ParamChem to 0.75 for ML-MATCH.

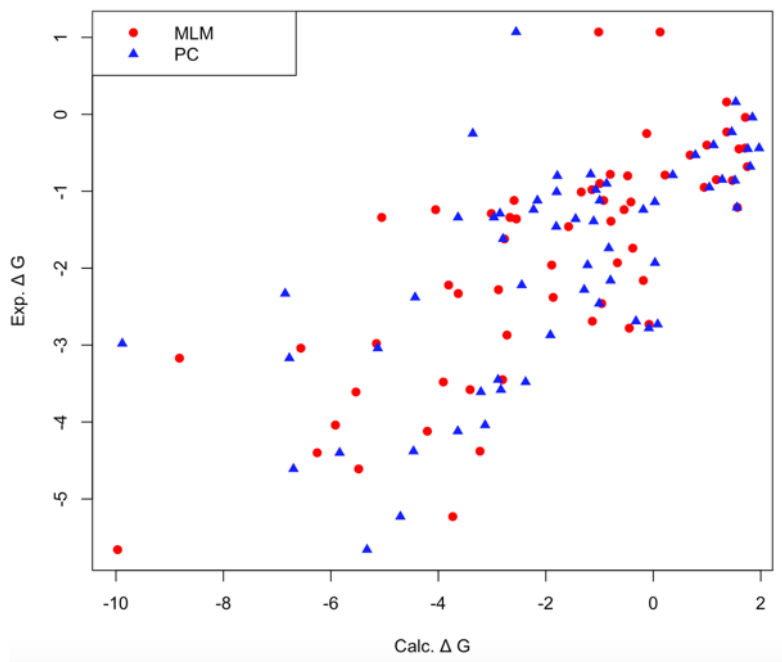


Figure 3.12: FACTS model calculations results. Red circles are ML-MATCH vs Experimental free energy of hydration. Blue triangles are ParamChem vs Experimental free energy of hydration.

Just as with the GBMV2 module results, we performed the exercise of extracting the trifluoromethyl containing molecules. These results are shown in Table 3.5. We again see that the extraction of these molecules causes greater correlation between ML-MATCH results and ParamChem. However, for the FACTS model we see that ML-MATCH outperforms ParamChem when compared to experiment for these 57 molecules.

	FACTS		
	MLM/Exp.	ParamChem/Exp.	MLM/ParamChem
Pearson	0.7814	0.7575	0.9331
MUE	2.2211	1.9817	0.8379
Spearman	0.7951	0.8148	0.9673
RMSD	2.7743	2.6970	1.3403

Table 3.5: Free energy of hydration results for FACTS with removal of trifluoromethyl containing molecules.

In addition to molecules (b) and (d) from the GBMV2 calculations, we have identified an additional 10 molecules whose calculated free energy of hydration negatively deviates from experimental values and whose error is 1 kcal/mol different than ParamChems's when compared to experiment. We see that although ML-MATCH performs better than ParamChem on average for these molecules when using the FACTS model, for those that it does poorly predict the deviation is larger than that of GBMV. We see the highest deviation in molecule (j) of 4.4256 kcal/mol and a low of 1.2662 kcal/mol for molecule (i).

	Pearson R Value	RMSE	Abs. Diff of $\Delta\Delta G$ (kcal/mol)
(f) 1,2,3,4-tetrachloro-5-(3,4,5-trichlorophenyl)benzene	0.9693	0.0340	2.0212
(g) 1,2,3-trichlorobenzene	0.9727	0.0319	1.8702
(h) 1,2,3,4-tetrachlorobenzene	0.9729	0.0332	1.4769
(i) fluorobenzene	0.9428	0.0494	1.2663
(j) N3,N3-diethyl-2,4-dinitro-6-(trifluoromethyl)benzene-1,3-diamine	0.8929	0.1416	4.4257
(k) 1,2,3,4-tetrachloro-5-(3,4-dichlorophenyl)benzene	0.9703	0.0326	1.7439
(l) 1,3-dichloro-2-(2,6-dichlorophenyl)benzene	0.9317	0.0457	1.7535
(m) 1,2,3-trichloro-5-(2,5-dichlorophenyl)benzene	0.9642	0.0344	2.3731
(n) 1,2,3,4-tetrachloro-5-phenyl-benzene	0.9756	0.0283	1.4138
(o) bromomethylbenzene	0.7219	0.0966	1.9995

Table 3.6: FACTS model results ML-MATCH and ParamChem. ML-MATCH vs ParamChem charge comparison for those molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem with FACTS model.

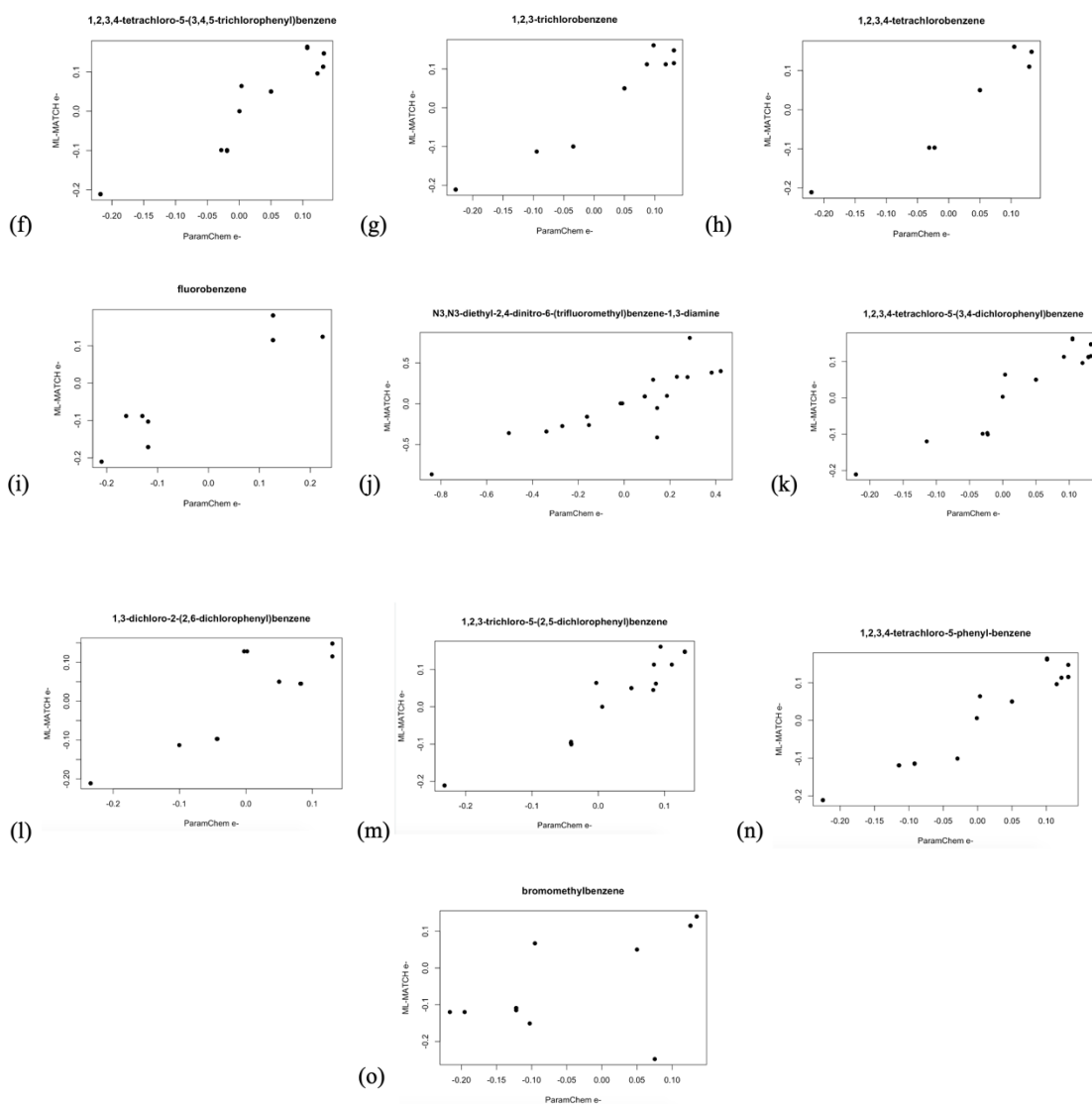


Figure 3.13: Partial charge comparison for FreeSolve molecules with differentially calculated FEHs by FACTS. Molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem as defined by the FACTS model.

Figure 3.14 shows that ML-MATCH and ParamChem offer similar decisions for atom types within this subset of molecules with the highest accuracy being in molecule (j) at

0.9714 and lowest being in molecules (m) at 0.7407. Just as with GBMV, this agreement in parameterization of atom types and partial charges between ML-MATCH and ParamChem highlights that the potential deviation stems from the bonded parameters. Further investigation is needed to thoroughly understand the impact that the dissimilar bonded parameters have in simulation.

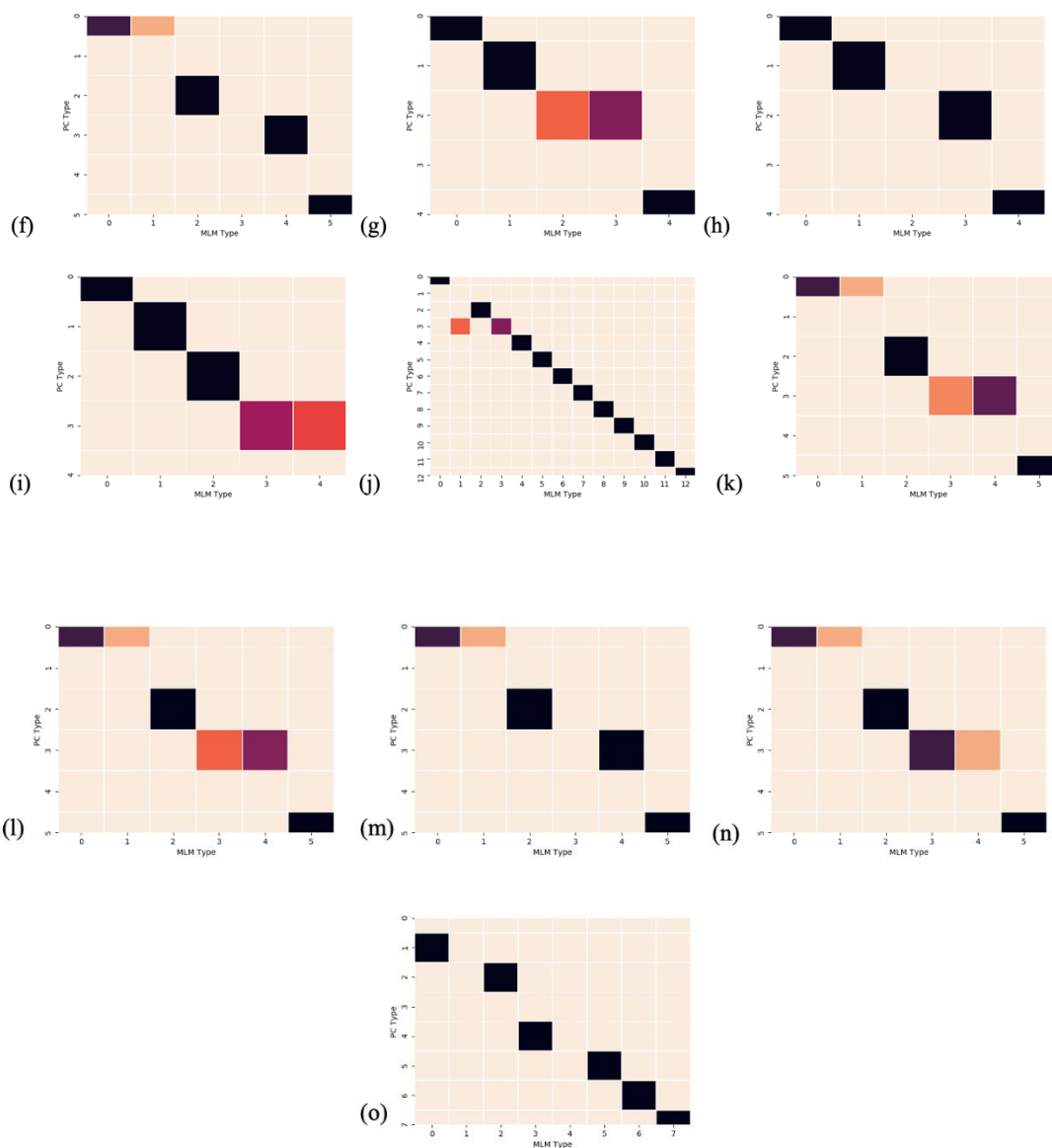


Figure 3.14: Atom types correlation matrices for FreeSolve molecules with differentially calculated FEHs by GBMV2. Molecules which have a difference of greater than 1 kcal/mol and has a greater deviation in comparison to experiment in ML-MATCH than ParamChem given by the FACTS model. The accuracy between the schemes for each molecule is (f) 0.8276, (g) 0.8667, (h) 0.8750, (i) 0.8333, (j) 0.9714, (k) 0.8214, (l) 0.7696, (m) 0.7407, (n) 0.8864 and (o) 0.8750.

3.5 Summary

In this Chapter, we provided the outcomes of the ML-MATCH framework. We offered the reasoning behind the selected underlying algorithm, Random Forest and implemented ML-MATCH parameterized molecules in simulation. We have shown through free energy of hydration simulations that ML-MATCH is both useful and accurate in simulation. This exercise highlighted areas in which ML-MATCH produces comparable or even better results when compared to ParamChem as well as chemical entities for which ML-MATCH parameters perform poorer in simulation when compared to experiment for this dataset. Overall statistics of solvation free energy predictions presented herein demonstrate that ML-MATCH is able to provide parameters for the studied molecules that yield relative free energy results that rank similarly and correlate linearly to experiment – all at the expense of little to no loss of accuracy when compared to other parameterization schemes. This is particularly significant in that ML-MATCH is not explicitly programmed to produce the parameters of a specific force field. With the implementation of a general representation of the local atomic environment, ML-MATCH is able to well capture the relationship between small molecule force field parameters and an atom's nearby surroundings. In Chapter 4,

we provide insight into the current efforts towards the optimization and implementation of ML-MATCH as well as future directions.

References

1. Rish, I., 2001, An empirical study of the naïve Bayes Classifier.
2. Bishop, C.M., & Tipping, M.E., 2003, *Advances in Learning Theory: Methods, Models and Applications*, Vol. 190, pp. 276-285.
3. Dudani, S.A., 1976, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC 6, pp. 325-327.
4. Breiman, L., 2001, Random Forests.
5. Murphy, K.P., 2006, *Naïve Bayes Classifiers*.
6. Tzanos, G., Kachris, C., & Soudris, D., 2019, 2019 8th International Conference on Modern Circuits and Systems Technologies.
7. Vanommeslaeghe, K. & MacKerell, Jr., A.D. 2012, *Journal of Chemical Engineering and Modeling*, Vol. 52, pp. 3144-3154.
8. J. Scheen, W. Wu, A. S. J. S. Mey, P. Tosco, M. Mackey, and J. Michel, 2020, *J. Chem. Inf. Model*.
9. G. Duarte Ramos Matos, D. Y. Kyu, H. H. Loeffler, J. D. Chodera, M. R. Shirts, and D. L. Mobley, 2017, *Journal of Chemical and Engineering Data*.
10. S. Luukkonen, L. Belloni, D. Borgis, and M. Levesque, 2020, *J. Chem. Inf. Model*.
11. J. L. Knight and C. L. Brooks, 2011, *J. Comput. Chem*.
12. B. R. Brooks et al., 2009, *J. Comput. Chem*.
13. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, 1983, *J. Comput. Chem*.
14. X. Ding, J. Z. Vilseck, and C. L. Brooks, 2019, *J. Chem. Theory Comput.*, 2019.
15. L. Wang et al., 2015, *J. Am. Chem. Soc.*, 2015.

Chapter 4

Implications and Future Directions

The overarching motivation for the work presented in this dissertation is Feynman's finding that if one has a sufficient number of examples of local atomic environment atomic forces, one can predict the force acting upon an atom based on the configuration of its neighbors. This body of work which builds the Machine Learning based Atom Typer for CHARMM (ML-MATCH), extrapolates this finding for the purpose of small molecule parametrization by looking at each molecule from an atom-centric viewpoint. From this work emerged the understanding of several themes for the field of small molecule parameterization. This chapter explains the implications of our findings and the future direction that we can take to further optimize and extrapolate this framework.

4.1 Well-defined atomic descriptors enable accurate force field parameter predictions

As one of the first steps of this work, we generated a new atomic fingerprint. The reasoning for this is that we wanted to more readily determine the characteristics that would be pertinent in distinguishing local atomic environments. With the use of OpenBabel, we were able to create a fingerprint of length 340 that encompassed environmental characteristics for each atom out to its second nearest covalently bonded neighbor. With the use of Random Forest algorithms implemented using sci-kit learn, we

were able to show that these newly developed fingerprints well depicted the atomic local environments and provide the necessary information to relate environment to atom type. In addition, we found that these descriptors also allow for accurate predictions of partial charges. Although we did not extend our work to using existing atomic fingerprints, we found our average regression RMSE of 0.028e to be very similar to published findings with an RMSE of 0.030e [1] which also used Random Forest for the prediction of partial charges. This suggests that this newly developed fingerprint performs as well for this particular task when compared to the widely use atomic fingerprint produced by the `GetHashedAtomPairFingerprintAsBitVec()` RDKit function. The next step in the further development of this atomic environment descriptor is comparison with other existing fingerprints in RDKit [2] and OpenBabel to further optimize and validate its usage.

Additionally, we have found that ML-MATCH not only provided good parameters when trained using CGenFF but we have also seen that ML-MATCH can be well trained using other force fields. We have trained ML-MATCH to predict AM1-BCC partial charges as an effort to depict this paradigm's ability to be extrapolated to additional force fields.

AM1-BCC charges were generated using ANTECHAMBER for all molecules contained in the CHARMM General Force Field. The machine learning algorithms were developed just as in Chapter 2. The preliminary findings are in the table below.

Element Grouping	R ²	RMSE	MAE
Br	0.629	3.00E-04	1.42E-02
C	0.974	1.00E-03	1.12E-02
Cl	0.861	3.00E-04	1.23E-02
F	0.989	2.59E-05	3.72E-03
H	0.957	3.78E-04	1.06E-02
I	-0.198	2.05E-04	1.44E-02
N	0.95	1.95E-04	7.17E-03
O	0.34	1.50E-02	4.12E-02
P	0.243	4.76E-01	3.23E-01
S	0.999	2.36E-04	9.15E-03

Table 4.1: Results for ML-MATCH trained on AM1-BCC. ML-MATCH partial charge regression models results trained using AM1-BCC charges defined by ANTECHAMBER.

We found the AM1-BCC trained Random Forest regression models perform very well overall with some element groupings [O,P,Br] needing further improvement. The iodine regression model performed poorly which is expected as there only exists one iodine sample in the training set. Further improvement of these models will come with identifying those atomic environments for which the model does not perform well and determining the reasoning. Optimization may come in the form of further model hyperparameterization procedures or the addition of molecules which contain atomic environments that are not well represented in the training set. These results show that the

newly developed atomic fingerprints can recapitulate varied charges which depicts the generalizability of the ML-MATCH framework.

4.2 The application of machine learning is well suited to force field development

The body of work adds to the existing applications of machine learning to force field development. Small molecule parameterization is increasingly complex as time goes on and it is important for us to be able to use the knowledge that we have gained in the past through QM calculations and other parameterization engines to inform how we handle this issue in the future. ML-MATCH is a step in the right direction as it takes advantage of a well-curated group of molecules and well-vetted parameterization to generate force field parameters for novel chemical moieties. We see the promising results of ML-MATCH in Chapter 3, and we hope that subsequent optimization will further improve the accuracy and precision of this methodology.

4.3 Further and more complex simulations are necessary for optimizing ML-MATCH

As an effort to determine how far we can push the parameters defined by ML-MATCH we ran Multisite – λ Dynamics (MSLD) [3] calculations to compute the binding free energy of a β -site amyloid precursor protein cleaving enzyme β -Secretase (BACE) and potential inhibitors. MSLD is thoroughly explained in Reference 3. It has been found that the accumulation of amyloid β ($A\beta$) oligomers in the brain is a pathological event of Alzheimer's disease [4]. The inhibition of BACE blocks the first step of $A\beta$ formation subsequently reducing build up. Figure 4.1 shows the structure of BACE while Figure

4.1(b) shows the common inhibitor substructure and the substituent which is placed at each site.

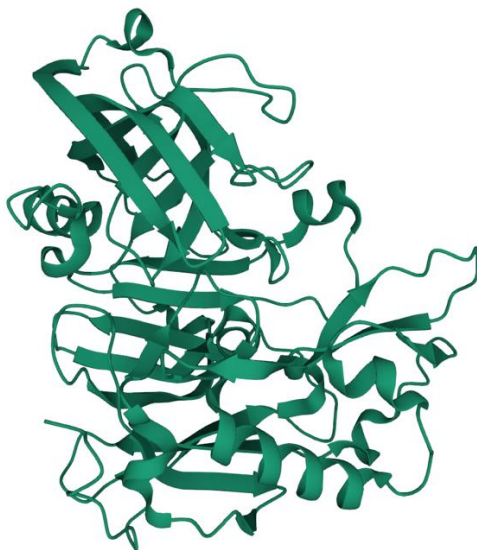


Figure 4.1: Crystal structure of β -Secretase PDB ID: 3SKF. [5]

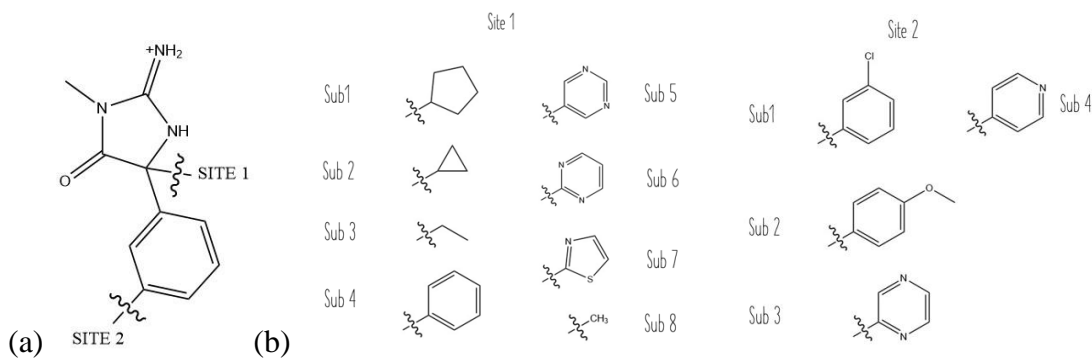


Figure 4.2: Collection of BACE inhibitors. (a) Depicts the common core substructure of BACE inhibitors which is a triazole moiety bound to a six-member aromatic ring. (b) All substituents which make up the potential inhibitors of BACE when bound to Site 1 and 2 on the common core substructure of the inhibitor.

To test ML-MATCH's accuracy in simulation we parametrized ten inhibitors using both ML-MATCH and ParamChem. These simulations specifically investigate the comparison of the charging schemes between models. MSLD was run using ParamChem and the charges were perturbed in simulation and transitioned to ML-MATCH to compare the goodness binding free energy approximations between models. This test was quite a presumptuous secondary test for ML-MATCH. The running of MSLD is quite complex and involves charge perturbations which may affect simulation performance and convergence. We first compare the predicted atom types and partial charges given by ML-MATCH and ParamChem. Figure 4.3 shows the correlation matrices for each molecule where ML-MATCH predicted molecules are on the x-axis and ParamChem on the y-axis. We found an average accuracy score of 79.9% between ML-MATCH and ParamChem predicted atom types. We find that for each molecule, the approximately 20% deviation comes from the atom types within the triazole ring. Which is to be expected as this particular moiety is not well represented in the CHARMM General Force Field.

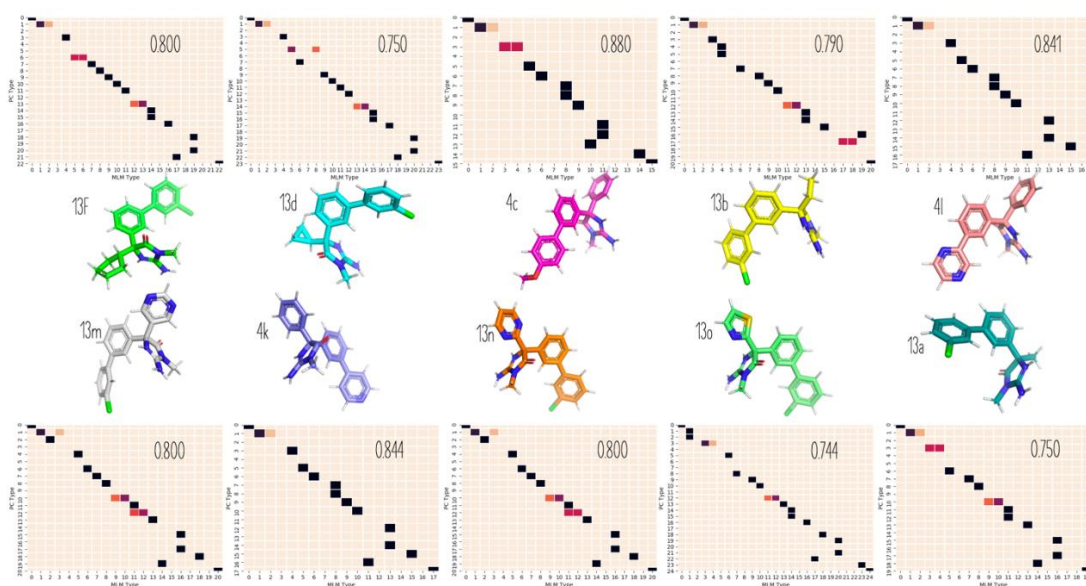


Figure 4.3: ML-MATCH vs ParamChem correlation matrices for atom type assignment.

Correlation matrices for the assignment of atom type for 10 tested BACE inhibitors.

Large deviations in atom types come from the unique triazole moiety within common core substructure of the molecule.

In addition, we compared the regression algorithms for the assignment of partial charges between models. Figure 4.4 shows relatively decent correlations constants with an average Pearson R-value of about 0.695. Again, we found that the deviation in partial charges between models stems of the triazole moiety.

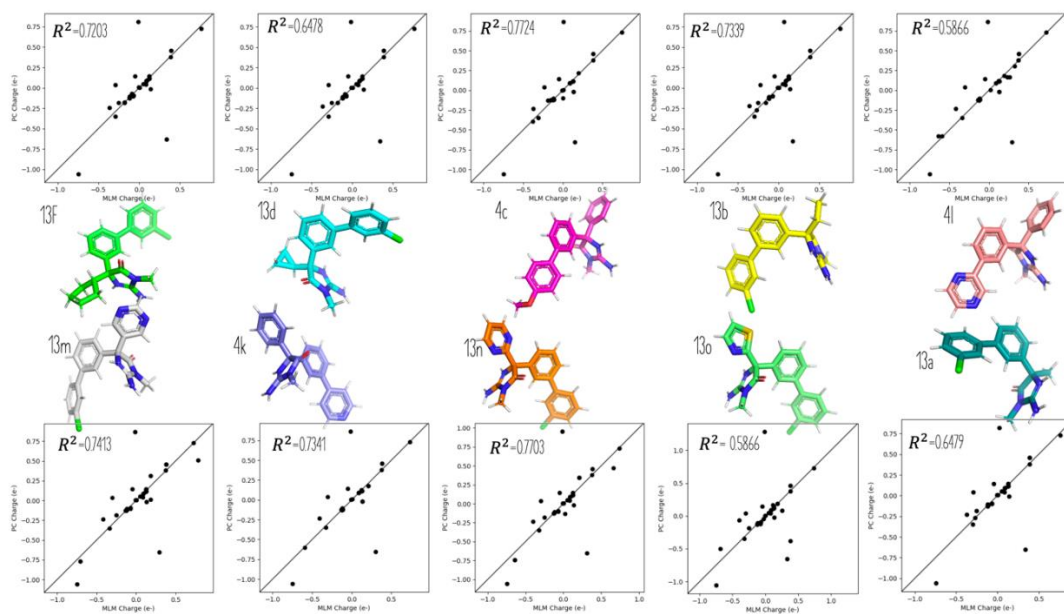


Figure 4.4: ML-MATCH vs ParamChem scatter plots for partial charge assignment.

Scatter plots comparing partial charges assignments for 10 tested BACE inhibitors. Large

deviations in atom types come from the unique triazole moiety within common core substructure of the molecule.

It was unsurprising to see varying performances across the molecules. Table 4.2 shows the breakdown for each substituent at each site and the corresponding experimental and calculated binding free energies. We found that ML-MATCH performed better than ParamChem for two molecules, very similarly for three molecules and poorer than ParamChem for 5 out of ten molecules. As a first in-depth test of this algorithm, these findings are promising. We must note that due to the rules-based nature of ParamChem as described in Chapter 1, generally when a common core substructure is within a group of chemical moieties, the charges of atom within that substructure will be the same across that subset of molecules. We see that with this group of molecules. As a result, the scheme is well suited for MSLD. However, the same cannot be said for ML-MATCH. Since the parameterization scheme of ML-MATCH is more general in nature and is based solely on an atom's local environment and not rules, we generally see slight charge deviations in the common substructure which then leads to increased charge renormalization for this method. This process may skew results. Further investigation is needed to determine this.

Ligand	Site 1	Site 2	Water dG	Protein dG	ddG	ParamChem UE	ML-MATCH UE	UE Diff
13f	1	1	6.866	6.501	-0.218	1.365	2.931	1.566
13d	2	1	6.601	7.58	0.526	1.807	0.985	-0.822
4c	3	2	18.275	18.12	1.266	1.192	1.274	0.082
13b	4	1	27.574	28.794	0.855	0.04	0.534	0.494
4l	3	3	30.029	30.888	0.93	0.593	0.175	-0.418
13m	5	1	145.001	148.244	3.9509	0.569	2.033	1.464
4k	3	4	17.488	18.021	1.168	1.569	1.749	0.18
13n	6	1	-54.212	-53.613	2.695	0.924	2.271	1.347
13o	7	1	27.225	31.245	3.325	1.369	3.269	1.9
13a	8	1	64.365	64.4814	-0.94	0.043	2.245	2.202

Table 4.2: MSLD binding free energy results for BACE inhibitors. Binding free energy results for charge perturbation simulation from ParamChem charges to ML-MATCH defined charges for 10 potential inhibitors of BACE.

4.4 Concluding Remarks

Machine learning based Atom Type for CHARMM is an atom parameterization engine that surpasses current atom parameterization schemes in its unique ability to understand and traverse the complex physical principles of existing small molecule force fields without having to be explicitly programmed to do so. ML-MATCH takes advantage of the power of machine learning to quantify the non-linear relationship between local atomic environment at force field parameters. We have found that with enough atomic environment to force field parameter examples, we can accurately and efficiently parameterize novel chemical entities. In this dissertation, we show that the ML-MATCH framework can be well applied to the CHARMM general force field and subsequently applied to various other small molecules force fields.

The work performed during this dissertation highlights the power of machine learning and its vast applications. From applying techniques normally used for text processing for the generation of atomic fingerprints to the generation of a brand-new atom parametrization paradigm, this dissertation provides a unique perspective for the application of machine learning to the prediction of small molecule force field parameters. We demonstrate the ML-MATCH is a paradigm that could be applied to a number of organic force fields and could eventually be developed to give the user their

choice of force field. It is our hope that ML-MATCH enables the advancement of molecular mechanics-based drug discovery through the use of machine learning.

References

1. Bleiziffer, P., et. al., 2018, Journal of Chemical Informatics and Modeling. Vol. 58, pp. 579–590.
2. RDKit: Open-source cheminformatics; <http://www.rdkit.org>
3. Knight, J., & Brooks, C.L. III., 2011, Journal of Chemical Theory and Computation. Vol 7, pp. 2728–2739.
4. Vassar, R., Alz Res Therapy, 2014, 6, 89.
5. Thompson, L.A., et.al. 2012, Bioorganic & Medicinal Chemistry Letters, Vol 21, pp. 6909-6915.

Appendix II

Appendix II Figure 1: ML-MATCH output for the simple benzaldehyde for which exact matches are found.

```
*Toppar stream files generated
*Machine Learning MATCH (ML-MATCH) v1.0.0
*Written by Murchtricia K Charles-Jones (murchkia(at)umich.edu)
*For use with CGenFF v4.1
*

read rtf card append
*Machine Learning MATCH (ML-MATCH) v1.0.0
*
36 1

!C.O.P -> Certainty of Prediction for Atom Type Classification
!stdev -> Standard Deviation of Partial Charge Prediction

RESI benzaldehyde      0.000
GROUP
ATOM  C1      CG2R61  -0.1151      !C.O.P: CG2R61  100.0% , stdev: 0.0
ATOM  C2      CG2R61  -0.1221      !C.O.P: CG2R61  100.0% , stdev: 0.002
ATOM  C3      CG2R61  -0.1221      !C.O.P: CG2R61  100.0% , stdev: 0.002
ATOM  C4      CG2R61  -0.1221      !C.O.P: CG2R61  100.0% , stdev: 0.002
ATOM  C5      CG2R61  -0.1151      !C.O.P: CG2R61  100.0% , stdev: 0.0
ATOM  C6      CG2R61  0.0862      !C.O.P: CG2R61  100.0% , stdev: 0.002
ATOM  C7      CG2O4    0.2381      !C.O.P: CG2O4    39.571% CG2O6  25.0% , stdev: 0.003
ATOM  H1      HGR52    0.0889      !C.O.P: HGR52    48.0%  HGA4    27.0% , stdev: 0.008
ATOM  O1      OG2D1   -0.4472      !C.O.P: OG2D1   83.0%  OG2D5   17.0% , stdev: 0.008
ATOM  H2      HGR61    0.1261      !C.O.P: HGR61   48.0%  HGA4    27.0% , stdev: 0.0
ATOM  H3      HGR61    0.1261      !C.O.P: HGR61   48.0%  HGA4    27.0% , stdev: 0.0
ATOM  H4      HGR61    0.1261      !C.O.P: HGR61   48.0%  HGA4    27.0% , stdev: 0.0
ATOM  H5      HGR61    0.1261      !C.O.P: HGR61   48.0%  HGA4    27.0% , stdev: 0.0
ATOM  H6      HGR61    0.1261      !C.O.P: HGR61   48.0%  HGA4    27.0% , stdev: 0.0

read param card flex append

BONDS
CG2R61  CG2R61    305.00    1.3750
CG2R61  CG2O4    300.00    1.4798
CG2O4   HGR52    330.00    1.1100
CG2O4   OG2D1    700.00    1.2150
CG2R61  HGR61    340.00    1.0800

ANGLES
CG2R61  CG2O4   HGR52    15.00    116.00
CG2R61  CG2O4   OG2D1    75.00    126.00
CG2R61  CG2R61  CG2O4    45.00    119.80
CG2R61  CG2R61  CG2R61    40.00    120.00
CG2R61  CG2R61  HGR61    30.00    120.00
HGR52   CG2O4   OG2D1    65.00    118.00

DIHEDRALS
CG2R61  CG2R61  CG2R61  CG2O4    3.1000    2    180.00
HGR52   CG2O4  CG2R61  CG2R61    1.0800    2    180.00
OG2D1   CG2O4  CG2R61  CG2R61    1.0800    2    180.00
CG2O4   CG2R61  CG2R61  CG2R61    3.1000    2    180.00
CG2R61  CG2R61  CG2R61  CG2R61    3.1000    2    180.00
HGR61   CG2R61  CG2R61  CG2R61    4.2000    2    180.00
CG2O4   CG2R61  CG2R61  HGR61    2.4000    2    180.00
CG2R61  CG2R61  CG2R61  HGR61    4.2000    2    180.00
HGR61   CG2R61  CG2R61  HGR61    2.4000    2    180.00

IMPROPERS

END
RETURN
```

Appendix II Figure 2: ML-MATCH output for [(1S)-1-methylpropyl]benzene for which there are missing angles and dihedrals as defined by the predicted atom types.

```

*Toppar stream files generated
*Machine Learning MATCH (ML-MATCH) v1.0.0
*Written by Murchtricia K Charles-Jones (murchkia(at)umich.edu)
*For use with CGenFF v4.1
*

read rtf card append
*Machine Learning MATCH (ML-MATCH) v1.0.0
*
36 1

!C.O.P -> Certainty of Prediction for Atom Type Classification
!stdev -> Standard Deviation of Partial Charge Prediction

RESI mobley_103    0.000
GROUP
ATOM  C1      CG331    -0.2700      !C.O.P: CG331    100.0% , stdev: 0.0
ATOM  C2      CG321    -0.1758      !C.O.P: CG321    85.0%  CG301    15.0% , stdev: 0.003
ATOM  C3      CG311    -0.0900      !C.O.P: CG311    85.0%  CG301    15.0% , stdev: 0.0
ATOM  H1      HGA1      0.0900      !C.O.P: HGA1     37.0%  HGPAM1   20.0% , stdev: 0.0
ATOM  C4      CG331    -0.2700      !C.O.P: CG331    100.0% , stdev: 0.0
ATOM  C5      CG2R61   -0.0000      !C.O.P: CG2R61   100.0% , stdev: 0.0
ATOM  C6      CG2R61   -0.1409      !C.O.P: CG2R61   100.0% , stdev: 0.027
ATOM  C7      CG2R61   -0.1187      !C.O.P: CG2R61   100.0% , stdev: 0.002
ATOM  C8      CG2R61   -0.1187      !C.O.P: CG2R61   100.0% , stdev: 0.002
ATOM  C9      CG2R61   -0.1187      !C.O.P: CG2R61   100.0% , stdev: 0.002
ATOM  C10     CG2R61   -0.1409      !C.O.P: CG2R61   100.0% , stdev: 0.027
ATOM  H2      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H3      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H4      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H5      HGA2      0.0900      !C.O.P: HGA2     37.0%  HGPAM1   20.0% , stdev: 0.0
ATOM  H6      HGA2      0.0900      !C.O.P: HGA2     37.0%  HGPAM1   20.0% , stdev: 0.0
ATOM  H7      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H8      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H9      HGA3      0.0900      !C.O.P: HGA3     31.0%  HGA5     30.0% , stdev: 0.0
ATOM  H10     HGR61     0.1268      !C.O.P: HGR61    48.0%  HGA4     27.0% , stdev: 0.0
ATOM  H11     HGR61     0.1268      !C.O.P: HGR61    48.0%  HGA4     27.0% , stdev: 0.0
ATOM  H12     HGR61     0.1267      !C.O.P: HGR61    48.0%  HGA4     27.0% , stdev: 0.0
ATOM  H13     HGR61     0.1267      !C.O.P: HGR61    48.0%  HGA4     27.0% , stdev: 0.0
ATOM  H14     HGR61     0.1267      !C.O.P: HGR61    48.0%  HGA4     27.0% , stdev: 0.0

```

read param card flex append

BONDS

CG321	CG331	222.50	1.5280
CG311	CG321	222.50	1.5380
CG311	HGA1	309.00	1.1110
CG311	CG331	222.50	1.5380
CG2R61	CG311	230.00	1.4900
CG2R61	CG2R61	305.00	1.3750
CG331	HGA3	322.00	1.1110
CG321	HGA2	309.00	1.1110
CG2R61	HGR61	340.00	1.0800

ANGLES

CG2R61	CG2R61	CG2R61	40.00	120.00
CG2R61	CG311	CG331	51.80	107.50
CG311	CG2R61	CG2R61	45.80	120.00
CG311	CG331	HGA3	33.43	110.10
CG321	CG311	CG2R61	47.00	125.20 ! from: CG2D2 CG2D1 CG331, deviation : 8.4216
CG321	CG311	CG331	53.35	114.00
CG321	CG311	HGA1	34.50	110.10
CG331	CG311	HGA1	34.50	110.10
CG311	CG321	CG331	58.35	113.50
CG331	CG321	HGA2	34.60	110.10
CG2R61	CG311	HGA1	43.00	111.00
CG311	CG321	HGA2	33.43	110.10
HGA2	CG321	HGA2	35.50	109.00
CG321	CG331	HGA3	34.60	110.10
HGA3	CG331	HGA3	35.50	108.40
CG2R61	CG2R61	HGR61	30.00	120.00

DIHEDRALS

CG2R61	CG2R61	CG2R61	CG2R61	3.1000	2	180.00
CG311	CG2R61	CG2R61	CG2R61	3.1000	2	180.00
HGR61	CG2R61	CG2R61	CG2R61	4.2000	2	180.00
CG331	CG311	CG2R61	CG2R61	0.2300	2	180.00
CG2R61	CG311	CG321	HGA2	3.2000	2	180.00 ! from: CG2DC3 CG2DC1 CG204 HGR52, deviation : 7.8544
HGA3	CG331	CG311	CG2R61	0.0400	3	0.00
CG2R61	CG2R61	CG2R61	CG311	3.1000	2	180.00
HGR61	CG2R61	CG2R61	CG311	2.4000	2	180.00
HGA3	CG331	CG321	CG311	0.1600	3	0.00
CG321	CG311	CG2R61	CG2R61	2.0000	2	0.00 ! from: CG2DC1 CG2DC1 CG2DC2 CG2DC3, deviation : 9.3753
HGA3	CG331	CG311	CG321	0.2000	3	0.00
CG2R61	CG2R61	CG311	CG331	0.2300	2	180.00
HGA2	CG321	CG311	CG331	0.2000	3	0.00
CG331	CG321	CG311	CG2R61	2.0000	2	0.00 ! from: CG2DC1 CG2DC1 CG2DC2 CG2DC3, deviation : 7.9096
CG331	CG321	CG311	CG331	8.5000	2	180.00 ! from: CG331 CG2D1 CG2D1 CG331, deviation : 6.5172
CG331	CG321	CG311	HGA1	3.2000	2	180.00 ! from: CG2DC3 CG2DC1 CG204 HGR52, deviation : 6.6056
HGA1	CG311	CG2R61	CG2R61	3.2000	2	180.00 ! from: CG2DC3 CG2DC1 CG204 HGR52, deviation : 9.1569
HGA3	CG331	CG311	HGA1	0.1950	3	0.00
HGA2	CG321	CG311	CG2R61	3.2000	2	180.00 ! from: CG2DC3 CG2DC1 CG204 HGR52, deviation : 7.8544
CG331	CG311	CG321	HGA2	0.2000	3	0.00
HGA1	CG311	CG321	HGA2	0.1950	3	0.00
HGA3	CG331	CG321	HGA2	0.1600	3	0.00
HGA1	CG311	CG331	HGA3	0.1950	3	0.00
CG311	CG321	CG331	HGA3	0.1600	3	0.00
HGA2	CG321	CG331	HGA3	0.1600	3	0.00
CG2R61	CG2R61	CG2R61	HGR61	4.2000	2	180.00
HGR61	CG2R61	CG2R61	HGR61	2.4000	2	180.00

IMPROPERS

END
RETURN

Appendix II Figure 3 (a): Molecule (a) trimethoxymethylbenzene parameterization by ML-MATCH

```
*Toppar stream files generated
*Machine Learning MATCH (ML-MATCH) v1.0.0
*Written by Murchtricia K Charles-Jones (murchkia(at)umich.edu)
*For use with CGenFF v4.1
*

read rtf card append
*Machine Learning MATCH (ML-MATCH) v1.0.0
*
36 1

!C.O.P -> Certainty of Prediction for Atom Type Classification
!stdev -> Standard Deviation of Partial Charge Prediction

RESI mobley_207    0.000
GROUP
ATOM  C1      CG331    -0.1306      !C.O.P: CG331  100.0% , stdev: 0.003
ATOM  O1      OG301    -0.3128      !C.O.P: OG301  100.0% , stdev: 0.004
ATOM  C2      CG301     0.3392      !C.O.P: CG301  100.0% , stdev: 0.018
ATOM  C3      CG2R61    0.0002      !C.O.P: CG2R61 100.0% , stdev: 0.0
ATOM  C4      CG2R61   -0.0583      !C.O.P: CG2R61 100.0% , stdev: 0.027
ATOM  C5      CG2R61   -0.1137      !C.O.P: CG2R61 100.0% , stdev: 0.002
ATOM  C6      CG2R61   -0.1137      !C.O.P: CG2R61 100.0% , stdev: 0.002
ATOM  C7      CG2R61   -0.1137      !C.O.P: CG2R61 100.0% , stdev: 0.002
ATOM  C8      CG2R61   -0.0583      !C.O.P: CG2R61 100.0% , stdev: 0.027
ATOM  O2      OG301    -0.3128      !C.O.P: OG301  100.0% , stdev: 0.004
ATOM  C9      CG331    -0.1306      !C.O.P: CG331  100.0% , stdev: 0.003
ATOM  O3      OG301    -0.3128      !C.O.P: OG301  100.0% , stdev: 0.004
ATOM  C10     CG331    -0.1306      !C.O.P: CG331  100.0% , stdev: 0.003
ATOM  H1      HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H2      HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H3      HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H4      HGR61    0.1277      !C.O.P: HGR61  48.0% HGA4   27.0% , stdev: 0.0
ATOM  H5      HGR61    0.1277      !C.O.P: HGR61  48.0% HGA4   27.0% , stdev: 0.0
ATOM  H6      HGR61    0.1277      !C.O.P: HGR61  48.0% HGA4   27.0% , stdev: 0.0
ATOM  H7      HGR61    0.1277      !C.O.P: HGR61  48.0% HGA4   27.0% , stdev: 0.0
ATOM  H8      HGR61    0.1277      !C.O.P: HGR61  48.0% HGA4   27.0% , stdev: 0.0
ATOM  H9      HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H10     HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H11     HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H12     HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H13     HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
ATOM  H14     HGA3     0.0900      !C.O.P: HGA3   31.0% HGA5   30.0% , stdev: 0.0
```

read param card flex append

BONDS

CG331	OG301	360.00	1.4150	
CG301	OG301	360.00	1.4150	
CG301	CG2R61	280.00	1.5000	! from: CG2N2 CG331 , deviation : 10.1541
CG2R61	CG2R61	305.00	1.3750	
CG331	HGA3	322.00	1.1110	
CG2R61	HGR61	340.00	1.0800	

ANGLES

CG2R61	CG2R61	CG2R61	40.00	120.00	
CG2R61	CG301	OG301	75.00	126.00	! from: CG2DC1 CG204 OG2D1, deviation : 9.072
CG301	CG2R61	CG2R61	40.00	119.00	! from: CG2DC3 CG2DC1 CG203, deviation : 10.089
CG301	OG301	CG331	95.00	109.70	
HGA3	CG331	HGA3	35.50	108.40	
CG2R61	CG2R61	HGR61	30.00	120.00	
OG301	CG301	OG301	70.49	107.00	
HGA3	CG331	OG301	45.90	108.89	

DIHEDRALS

CG2R61	CG2R61	CG2R61	CG2R61	3.1000	2	180.00	
HGR61	CG2R61	CG2R61	CG2R61	4.2000	2	180.00	
CG2R61	CG2R61	CG301	OG301	1.0000	2	180.00	! from: CG2DC3 CG2DC1 CG204 OG2D1, deviation : 9.5278
CG2R61	CG301	OG301	CG331	3.1000	2	180.00	! from: CG2D2 CG2D10 OG301 CG331, deviation : 7.44
CG301	CG2R61	CG2R61	CG2R61	2.0000	2	0.00	! from: CG2DC1 CG2DC1 CG2DC2 CG2DC3, deviation : 10.8291
CG301	CG2R61	CG2R61	HGR61	5.0000	2	180.00	! from: CG2DC1 CG2DC2 CG2DC3 HGA5, deviation : 9.5663
HGA3	CG331	OG301	CG301	0.2840	3	0.00	
CG331	OG301	CG301	CG2R61	3.1000	2	180.00	! from: CG2D2 CG2D10 OG301 CG331, deviation : 7.44
OG301	CG301	OG301	CG331	0.6700	2	0.00	
CG301	OG301	CG331	HGA3	0.2840	3	0.00	
CG2R61	CG2R61	CG2R61	HGR61	4.2000	2	180.00	
HGR61	CG2R61	CG2R61	HGR61	2.4000	2	180.00	
OG301	CG301	CG2R61	CG2R61	1.0000	2	180.00	! from: CG2DC3 CG2DC1 CG204 OG2D1, deviation : 9.5278
CG331	OG301	CG301	OG301	0.6700	2	0.00	

Appendix II Figure 3 (b): Molecule (a) trimethoxymethylbenzene parameterization by ParamChem.

```
* Toppar stream file generated by
* CHARMM General Force Field (CGenFF) program version 2.3.0
* For use with CGenFF version 4.3
*
read rtf card append
* Topologies generated by
* CHARMM General Force Field (CGenFF) program version 2.3.0
*
36 1

! "penalty" is the highest penalty score of the associated parameters.
! Penalties lower than 10 indicate the analogy is fair; penalties between 10
! and 50 mean some basic validation is recommended; penalties higher than
! 50 indicate poor analogy and mandate extensive validation/optimization.

RESI trimetho      0.000 ! param penalty= 34.000 ; charge penalty= 35.372
GROUP              ! CHARGE  CH_PENALTY
ATOM C1            CG331  -0.087 ! 14.871
ATOM O1            OG301  -0.342 ! 25.854
ATOM C2            CG301   0.479 ! 31.922
ATOM C3            CG2R61 -0.018 ! 35.372
ATOM C4            CG2R61 -0.107 ! 20.297
ATOM C5            CG2R61 -0.115 !  0.379
ATOM C6            CG2R61 -0.115 !  0.000
ATOM C7            CG2R61 -0.115 !  0.379
ATOM C8            CG2R61 -0.107 ! 20.297
ATOM O2            OG301  -0.342 ! 25.854
ATOM C9            CG331  -0.087 ! 14.871
ATOM O3            OG301  -0.342 ! 25.854
ATOM C10           CG331  -0.087 ! 14.871
ATOM H1            HGA3   0.090 !  0.000
ATOM H2            HGA3   0.090 !  0.000
ATOM H3            HGA3   0.090 !  0.000
ATOM H4            HGR61  0.115 !  0.060
ATOM H5            HGR61  0.115 !  0.000
ATOM H6            HGR61  0.115 !  0.000
ATOM H7            HGR61  0.115 !  0.000
ATOM H8            HGR61  0.115 !  0.060
ATOM H9            HGA3   0.090 !  0.000
ATOM H10           HGA3   0.090 !  0.000
ATOM H11           HGA3   0.090 !  0.000
ATOM H12           HGA3   0.090 !  0.000
ATOM H13           HGA3   0.090 !  0.000
ATOM H14           HGA3   0.090 !  0.000

read param card flex append
* Parameters generated by analogy by
* CHARMM General Force Field (CGenFF) program version 2.3.0
*

! Penalties lower than 10 indicate the analogy is fair; penalties between 10
! and 50 mean some basic validation is recommended; penalties higher than
! 50 indicate poor analogy and mandate extensive validation/optimization.

BONDS
CG2R61 CG301      230.00      1.4900 ! trimetho , from CG2R61 CG311, penalty= 8

ANGLES
CG2R61 CG2R61 CG301      45.80      120.00 ! trimetho , from CG2R61 CG2R61 CG311, penalty= 1.2
CG2R61 CG301  OG301      75.70      110.10 ! trimetho , from CG2R61 CG321 OG302, penalty= 12.5

DIHEDRALS
CG2R61 CG2R61 CG2R61 CG301      3.1000  2      180.00 ! trimetho , from CG2R61 CG2R61 CG2R61 CG311, penalty= 1.2
CG301  CG2R61 CG2R61 HGR61      2.4000  2      180.00 ! trimetho , from CG311 CG2R61 CG2R61 HGR61, penalty= 1.2
CG2R61 CG2R61 CG301  OG301      0.0000  2          0.00 ! trimetho , from CG2R61 CG2R61 CG321 OG302, penalty= 12.5
CG2R61 CG301  OG301  CG331      0.2000  3          0.00 ! trimetho , from CG203 CG301 OG301 CG331, penalty= 34

IMPROPERS

END
RETURN
```