PREDICTION OF MASS SPECTRA FOR NATURAL PRODUCTS USING AN AB INITIO APPROACH

A thesis submitted in partial fulfillment of the requirements for the degree

of

MASTER OF SCIENCE IN BIOINFORMATICS

of

RHODES UNIVERSITY, SOUTH AFRICA

Department of biochemistry and microbiology

Faculty of science

by

YOLANDA NOVOKOZA

19N8294

RHODES UNIVERSITY
Where leaders learn

TABLE OF CONTENTS

ABSTRACT

Mass spectrometry (MS) is a technique that measures the fragmentation of molecules, dependent on the molecule's chemical composition and structure, by first introducing a charge on the molecules. The instrument records the mass to charge ratio, but the energy from the ionization process causes the molecule to fragment. The resultant mass spectrum is highly indicative of not only the molecule analyzed, but also its chemical composition. MS is used in research and industry for both routine and research purposes.

One such way to ionize molecules for MS is by bombarding the molecule with electrons which is the basis of electron impact mass spectrometry (EIMS). Although EIMS is widely used, prediction of electron impact mass spectra from first principles is a challenging problem due to a need to accurately determine the probability of different fragmentation pathways of a molecule. *Ab initio* molecular dynamics based methods are able to explore in an automatic fashion the energetically available fragmentation paths thus give reaction mechanisms in an unbiased way.

The mass spectra of five molecules have been explored in work-flows leading to the prediction of mass spectra. These molecules include three natural products *alpha*-hispanolol, PFB oxime derivative and boronolide (for which experimental mass spectra were not available) and two compounds from the NIST database (for which experimental mass spectra were available).

For each of these systems many random conformations were generated using the RDKit library. To all conformations random velocities were applied to each atom. *Ab initio* molecular dynamics was performed on each conformer, using these initial random velocities using CP2K software, at DFTB+ level at a variety of highly raised temperatures (to accelerate the formation of fragments)

Fragmentation was monitored by iterating through all bonds, and identifying bond breakages during dynamics. Graph theoretical packages were used then to track distinct fragments generated. For each of these fragments, charges were determined from Mulliken analysis for all atoms on the fragment from the QM calculations and sum of atomic spin densities per fragment was also plotted. The fragment with the greatest charge (corresponding to the formation of a cation fragment) was taken for plotting on the mass spectrum. Finally, from the mass of the fragment and its elemental composition, the isotopic distribution for the fragment was determined, and this distribution was included by addition in to the mass spectrum.

For all trajectories, the sum of all isotopic distributions determined the final mass spectrum.

ABBREVIATIONS

Ab initio molecular dynamics	(AIMD)
Alpha hispanolol	(α-H)
Artificial Neuron Network	(ANN)
Born-Oppenheimer ab initio molecular dynamics	(BO-AIMD)
Density functional theory	(DFT)
Density functional tight binding	(DFTB)
Electron impact	(EI)
Electron volt	(eV)
Fragment-based drug discovery	(FBDD)
Gas chromatography-mass spectrometry	(GC-MS)
High performance computing	(HPC)
Human immunodeficiency virus	(HIV)
Impact excess energy	(IEE)
Mass spectrometry	(MS)
Molecular dynamics	(MD)
Molecular mechanics	(MM)
Natural Products	(NP)
Periodic boundary conditions	(PBCs)
Plasmodium falciparum DXR	(PfDXR)
Potential energy surface	(PES)
Protein data bank	(PDB)
Quantum chemical electron ionization mass spectrometry	(QCEIMS)
Quantum mechanics	(QM)
Ribonucleic acid	(RNA)
Root mean square deviation	(RMSD)
Root mean square fluctuation	(RMSF)

Simplified molecular input line entry system	(SMILES)
South African National compounds database	(SANCDB)
The national institute of standards and technology	(NIST)
Traditional chinese medicine	(TCM)
Ultra performance liquid chromatography	(UPLC)
Visual molecular dynamics	(VMD)
3-Dimensional	(3D)

ACKNOWLEDGEMENT

I would like to first thank God of Mount Zion for giving me the courage to be able to do and finish my project as it was not an easy road but his guidance has given me strength up until this far

I would also like to pass my greatest gratitude to Prof Kevin Lobb for being my supervisor and helping me through the entire project, and always be available whenever I need his assistance. I will forever be grateful to him.

I also thank all the RUBI members for the assistance and also my Masters class for all the support that they have given me throughout the year

Last but not least I thank my grandmother and all the family members who were there for me, praying with me through all the good and bad times, they have shown me support and love

FUNDING ACKNOWLEDGEMENT

I would like to give my sincere thanks to Johnson Matthey company for funding this research by paying for my fees and making sure that I do not go to bed on an empty stomach, I would like to thank them for their generosity, if it was not for them I wouldn't be here.

DEDICATION

I would like to dedicate this project to my late mother Andiswa Novokoza and late grandfather. They would be proud of me right now as they always encouraged me to never fear anything and that I can do anything I put my mind in.

1. INTRODUCTION

1.1 Mass Spectrometry

Mass spectrometry (MS) is a technique that measures the characteristics of a sample or a molecule, that is its chemical composition and structure, by converting the material to charged molecules in order to measure their mass to charge ratio. It is used as an analytical tool in industry and academia for both routine and research purposes. There are a wide range of applications of this analytical technique which include pharmaceutical (e.g. drug discovery, combinatorial chemistry, pharmacokinetics and drug metabolism), clinical (e.g. neonatal screening, haemoglobin analysis and drug testing), environmental (for example to check water quality and food contamination), biotechnology (in protein and peptides analysis) and geological (for example, oil composition) applications (Hassan, 2012).

Mass spectrometry is a highly sensitive technique that is able to analyze even minute quantities of the molecule. It does not only elucidate the structure of the compounds, it also gives molecular formula and the isotopic abundance of particular molecular formula, (Baghel *el al*, 2017). Mass spectrometry, at its simplest, enables the identification of the molecular weight of a compound. In most cases, mass spectrometry is based on the generation of positive ions in the vapour phase; these pass through an analyzer.

The mass spectrometer consists of the inlet system, ion generation chamber, analyzer tube, Ion collector and data collection system (Figure 1.1). The analyzer differentiates ion trajectories based on their mass to charge ratio, from which they may be identified by mass by the ion detector.

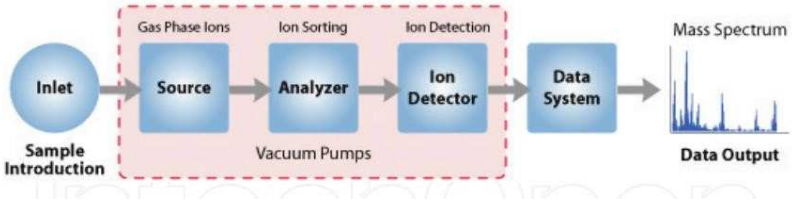


Figure 1.1: Schematic for Mass Spectrometry

In analytical chemistry compounds are often purified prior to mass spectrometry; this often involves chromatography (either liquid phase chromatography or gas chromatography), thus the compounds can first be separated by gas chromatography and electron ionization mass spectrometry can be used to fragment and identify the fragments, this can all be done using gas chromatography/mass spectrometry(Stein *et al.,* 1994).

1.2 Electron Impact Ionization in Mass Spectrometry

Prior to differentiation of ions in perhaps a magnetic sector analyzer, for electron impact ionization the sample that is investigated is converted into vapour phase and bombarded with electrons that have enough energy to knock out one electron from the molecule (<10eV). This generates a positively charged ion called molecular ion or parent ion that is denoted by M^+. This has some consequences in mass spectrometry.

When the molecule loses an electron (becomes positively charged) it normally becomes unstable. Since the molecule is less stable when it is positively charged, when the energy increases from 10eV to 70eV on the mass spectrometry the molecule will break into smaller pieces creating fragments. This energy is sufficient to overcome

bond strength for many types of bonds and the molecules break into smaller portions called fragments or daughter ions which are denoted by M_1^+. The generated ions are separated in the analyzer under the influence of electric and/or magnetic fields, following which, recording, detection and mass spectrum plotting occurs. When a parent ion fragments it is common that it breaks into two different parts; for example, it can fragment giving a positive ion together with an uncharged free radical, or a radical cation together with a neutral species. On the mass spectra only the positively charged particles will appear, the uncharged free radicals or uncharged species do not follow trajectories determined by the analyzer and thus get lost in the machine, subsequently being removed by the vacuum pump.

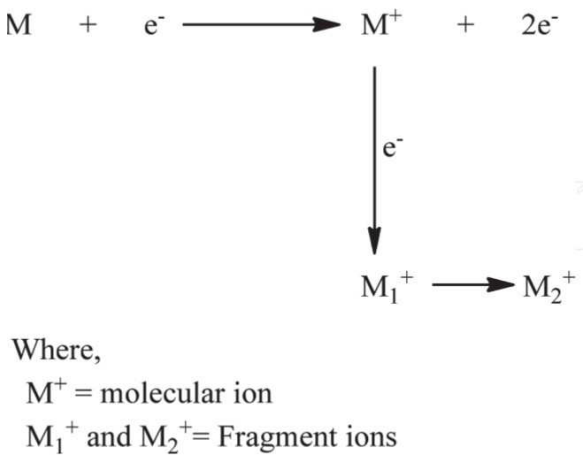


Figure 1.2: Schematic of ionization of a molecule by electron bombardment.

Electron impact (EI) is the major ionization method used in mass spectrometry and gas chromatography-mass spectrometry (GC-MS). EI provides a uniform semi-quantitative and high sensitivity response to all molecules and atoms. The observed fragmentation patterns may be matched with existing rich libraries of 70eV EI mass spectra, resulting in extensive molecular identification and structural information capability for this method (Dagan & Amirav, 1995).

1.3 Molecular spectra.

While computational prediction of spectra may appear to be relatively straightforward for properties such as infrared (IR) spectra (based on vibrational analysis) and UV-visible spectra (dependent on calculation of the excited states of a molecule), more recently in the literature strategies have been undertaken to explore accurate spectral prediction, including prediction of spectra that are not directly available from electronic structure theory, but where molecular conformation and other factors influence the spectra. Mass spectral prediction falls into this category, but it is worth exploring the literature in both complex and simple cases with respect to example Nuclear Magnetic Resonance (NMR) and vibrational spectral prediction.

1.3.1 Prediction of NMR spectra.

The shielding of the nuclei from the external magnetic field is calculated (in all directions - hence tensors are calculated). NMR shielding tensors may be computed using continuous set Gauge transformations (CSGT) method or be computed using Gauge-independent atomic orbital (GIAO) method. The CSGT method requires large se basis in order to obtain accurate results. Computing NMR spectra can be done using either Hartree-fock or DFT method (http://gaussian.com/nmr/).

Rychnovsky conducted a study in 2006 on predicting NMR spectra by computational methods

(Rychnovsky, 2006). This study had as purpose the structural revision of the natural product Hexacyclinol. As background Hexacyclinol had been extracted in 2002 from *panus rudis* strain HKI0254 and a polycyclic structure had been proposed on the basis of extensive 1D and 2D NMR data analysis (Schlegel *et al.*, 2002). Rychnovsky evaluated the use of a Density Functional Theoretical (DFT) method, performing calculations on several known highly oxygenated terpenes in order to determine whether these methods could also be used or if they would be applicable to confirming the structure from NMR of Hexacyclinol. Like Hexacyclinol, these three terpenes were highly oxygenated with a relatively high degree of unsaturation. The reference three diterpene natural products were Elisapterosin B, Elisabetein A and Maoecrystal V., The structural assignment for each of these three molecules was confirmed using X-ray analysis (appropriate since each of these

molecules was conformationally rigid). There were 3 steps used in the analysis, the global minimum was identified with a Monte-Carlo conformational search using the MMFF force field, the minimum was calculated with HF/3-21G method which gave more accurate structure with moderate computational cost. Finally the NMR chemical shifts were calculated with Gauge Included Atomic Orbitals (GIAO) using the mpw1pw91/6-31G(d,p) DFT method. For the Hartree-Fock calculations Spartan04 was used but the GIAO calculations were performed using Gaussian03. The chemical shift prediction provided good results; in particular the accuracy of these ^{13}C chemical shift predictions with highly oxygenated terpenes supported the use of this analysis to evaluate the proposed natural product structure for hexacyclinol.

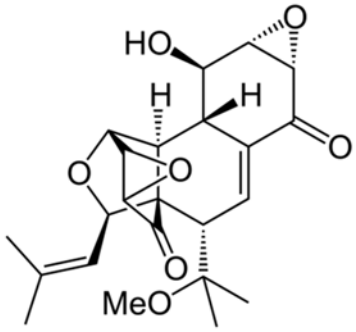


Figure 1.3: Structure of hexacyclinol

1.3.2 Prediction of Vibrational Spectra

Vibrational analysis allows for prediction of vibrational spectra, but also it allows for estimation of free energies of molecules and also characterizes the potential energy surface (real frequencies for minima, imaginary for transition states).

Vibrational analysis calculations are an example of spectral prediction where the initial spectra reproducibly do not match experimental spectra, but there is a requirement for frequencies to be scaled for this match to occur. Some studies have been performed by Palafox *et al.* on scaling factors for the prediction of vibrational spectra of several systems. For this study semi-empirical methods were also used (since they are potentially attractive for the fast computation of vibrational frequencies due to their low computational cost). Among the semi-empirical methods, they used MNDO, AM1, and

PM3 methods that are included in both the Gaussian94 and AMPAC packages. They used Gaussian94 for optimization and also to calculate the frequencies. In terms of DFT methods, SLYP and SVWN with no exchange, together with Becke's exchange functional (B88) and Becke's three parameter exchange functional (B3LYP) with correlation and exchange were all utilised. In all cases, they obtained optimum geometry and the harmonic frequencies were determined from the second derivative of the energy with respect to the nuclear displacement. It is interesting to note that it was the PM3 method that provided overall the most accurate frequency results in the C-H stretch region. However, for prediction of the vibrational spectrum as a whole, the B3LYP method at the 6-31G level or higher provided the best results. Further it is also of note that the DFT methods provided more reliable predictions for the calculated frequencies than with HF or MP2 methods. For some benzene derivatives there is a dependence of the scaling procedure on the size of the organic molecule and the accuracy required for the predicted frequencies. If the aromatic derivatives have less than 20-30 atoms, DFT methods with the 6-31G basis set is adequate to calculate the frequencies. If the derivatives are larger than 30 atoms then semi-empirical and DFT methods with small basis set were judged to be appropriate in terms of computational cost to calculate the frequencies (Palafox 2000, Palafox *et al.* 2005)

1.3.3 Informatics and Mass Spectroscopy

As an example, in lipids research mass spectrometry is a key tool used globally in the analysis of lipids, and this analysis is often through the combination of liquid chromatography and mass spectrometry (LC/MS) (Hermansson *et al.*, 2005; Houjou *et al.,* 2005). In the identification of lipids it is easy to use fragments observed within mass spectra as a characterization method, due to the structural characteristics of lipids. There have been computational strategies developed to utilise MS based approaches together with database searches in order to identify specific classes of lipids. However, there is still development in this area required for lipid analytics. This remains one of the biggest challenges in the elucidation of biological phenomena behind the large amounts of lipidomics data that is currently available (Katajamaa *et al.,* 2006; Stolt *et al.,* 2006).

Yetukuri *et al.*, 2007 conducted a study on bioinformatics strategies for lipidomics analysis: in the characterization of obesity related hepatic steatosis. This involved several stages of work. Starting with a generic simplified molecular input line entry system (SMILES) template for the glycerophospholipid class, they used the corresponding systematic names against fatty acid seed SMILES to generate names. They converted these SMILES systems into canonical SMILES, and from this derived formula and calculated molecular weight from the SMILES. From the atoms in the SMILES and lastly they calculated the isotopic distribution of each compound and fitted it to the resolution of the mass spectrometer. The Lipid profiling platform was based on the non-targeted analysis of total lipid extracts using ultra performance liquid chromatography (UPLC) coupled to quadrupole time of flight mass spectroscopy. The lipidDB database of lipids was constructed using SMILES, and the internal library has platform-specific information of the internal standards and lipid species identified using UPLC/MS/MS. For its use in mass spectrometry in the identification of lipids, the isotopic distribution for all molecular species held in the database is useful. While actual isotope patterns are kept in the database, the patterns are corrected for the resolution of the mass spectrometer when matching with spectral data.

1.3.4 Prediction of mass spectra in proteomics and metabolomics

In the field of proteomics, tandem mass spectrometry (MS/MS) is a necessary technique for highthroughput peptide identification and characterization (Aebersold & Mann, 2003). A challenge in this field is the need for accurate theoretical spectrum prediction, not only in terms of m/z peaks but also intensities of possible occurring ions. This is a requirement for both database search and *de novo* identification approaches. It is still challenging to accurately predict the theoretical spectrum because of poor understanding of the complex physical-chemical peptide fragmentation processes that occur during MS/MS experiment (Wang *et al.,* 2015).

To mediate these problems, Wang *et al.* conducted a study creating the openMS-simulator: an opensource software for theoretical tandem mass spectrum prediction in which the aim was to predict the intensity ratio of every adjacent *y*-ions (determined by near-neighbouring amino acids as well as remote amino acids). The open source package has four functions which are theoretical spectrum prediction, peptide spectrum match (PSM) re-ranking, false discovery rate (FDR) analysis and spectrum visualization. It takes the sequence of the peptide as input and then gives the results of the predicted theoretical spectra as the output. The open-source simulation uses a statistical model for the intensity production of possible ions and also has an extension to support higher-energy collisional dissociation (HCD) spectrum prediction; this tool can take many fragmentation pathways into consideration. It can also be used to accurately predict theoretical spectrum of a peptide sequence which helps to improve the identification of peptides (Wang *et al.*, 2015).

Figure 1.4 shows the comparison between the experimental and predicted spectra of a peptide.

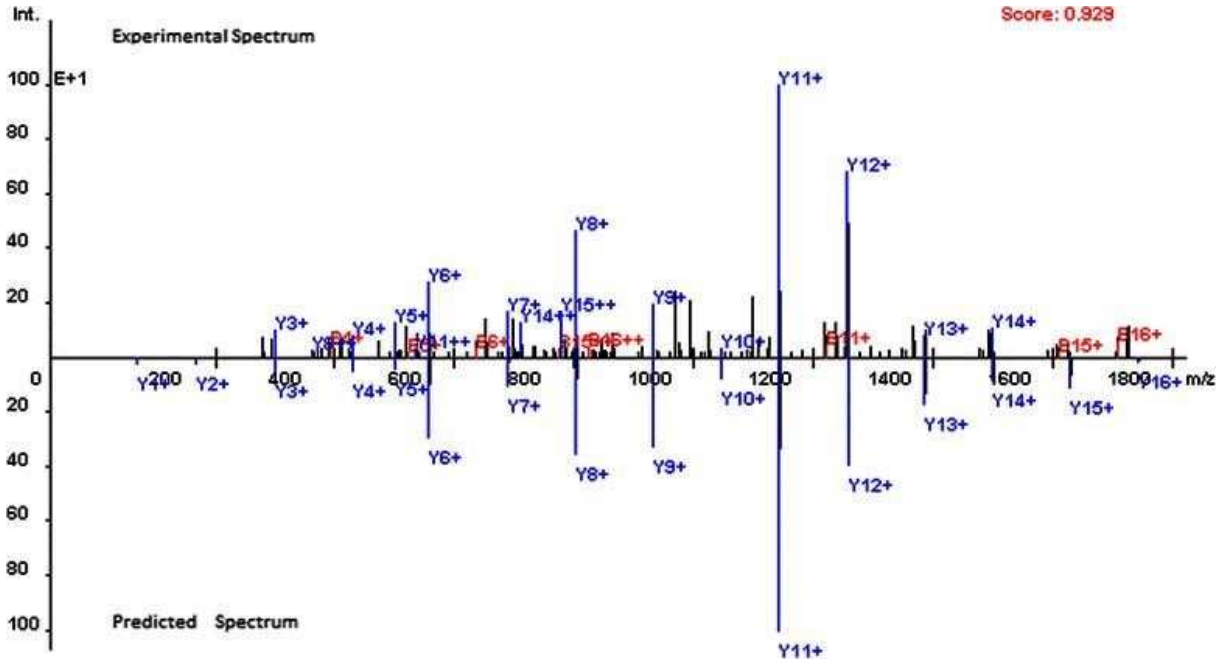


Figure 1.4 Experimental and predicted spectra of the peptide EIELEDPLENMGAQMVK using openMS-simulator (Wang *et al.*, 2015).

Metabolomics is a field of omics science which describes metabolics using high throughput technologies. Metabolites are low molecular weight (<1500Da) chemicals that are in the cells, tissues and biofluids (1,2). Electrospray tandem mass spectrometry (ESI-MS/MS) is mostly used in metabolomics experiments through the use of this technique is both time-consuming and tedious.

In 2014 Allen *et al.* conducted a study resulting in the generation of CFM-ID (Competitive Fragmentation Modeling for Metabolite Identification), a web server for annotations, spectrum prediction and metabolite identification from tandem mass spectra. The aim of the study was to help experimentalists by automating some of the more time-consuming tasks in the interpretation of mass spectrometry data, since this tool aids the analysis of MS/MS spectra of unknown compounds. It allows the user to identify a list of candidate molecules (by querying public chemical repository for molecules of correct mass). Upon identification of component compounds, it is designed with the aim of providing other analytical information, including information from other techniques such as NMR or information about the analyte compound source, e.g. from human blood.

In order to perform spectral prediction in CFM-ID, the input chemical structure can be provided in SMILES or InChi format and must be a neutral molecule. Prediction of the MS/MS spectrum is possible for a range of collision energies, including at 10V, 20V and 40V collision energy. In the published study they reported a ten-fold cross validation for positive mode spectrum prediction and compound identification tasks involving 1491 non-peptide metabolites from METLIN (Metabolite and Chemical Entity Database). In the spectrum prediction their method predicted the peak locations with measured and documented recall and precision values where the prediction was for non-peptide metabolites. The match for spectral peak intensity has also been quantified, providing confidence levels associated with expected predicted spectra from this server. For example, the match in intensity between predicted and experimental spectra provided Pearson correlation coefficients of 0.7 (for low energy spectra), 0.6 (for medium energy) and 0.45 for high energy spectra. As such the CFM-ID web server provides experimentalists with a new set of tools, with known performance characteristics, for the

interpretation of mass spectrometry data. A final point to note is that this particular server provides graphical views right the way through from spectrum prediction, peak assignment and compound identification (Allen *et al.*, 2014)

1.3.5 Prediction of mass spectra for general compounds

In a separate study, Allen *et al.* also constructed a software tool CFM-EI (Competitive Fragment Modeling for Electron Ionization) (Allen *et al.*, 2016). The purpose of this study was to assist in GC/EIMS compound identification, to extend the CFM-ESI (mentioned earlier) to EI spectra and to support existing databases where data is sparse by providing further reference compounds.

The change between CFM-ESI and CFM-EI (to make CFM applicable to EI-MS) involved modifications to accommodate odd electron ions, although some improvements were included regarding isotope distributions. The CFM-EI tool incorporated within its working an artificial neural networks, these artificial neural networks were the composition of so-called transition functions. These transition functions defined the formation of successive child fragments and were based on training sets where a break tendency was defined based on chemical features surrounding a bond. A trained CFM-EI model was used to predict spectra, and the quality of predicted spectra was evaluated not only on the peak positions, but also on their intensities. Again this provides a framework within which the tool is used, where the performance of the tool has been quantified. The CFM-EI tools is a stand-alone program, not a web server as is the case for CFM-ESI (Allen *et al.*, 2016).

Jalali-Heravi & Fatein conducted a study in 2000 on stimulation of mass spectra of noncyclic alkanes and alkenes using artificial neural networks. They conducted this study in order to evaluate the feasibility and resultant performance of using artificial neural network (ANN) for the simulation of mass spectra of organic compounds. To this end they constructed and trained the ANN for the simulation of mass spectra of a variety of organic compounds. They identified 262 organic compounds for this data set, including 117 noncyclic alkanes and 145 non-cyclic alkenes. All of these compounds are present in

The National Institute of Standards and Technology (NIST) mass spectral database (NIST, Gaitherburg, MD). All the spectra within the data set have been collected under similar conditions with an ion source energy of 70eV. They used topological descriptors for MS spectra simulations, 75 topological descriptors were calculated for each compound. The number of inputs in the ANN was equal to the number of descriptors and the number of outputs from the ANN corresponded to the number of m/z positions, for prediction of intensity at each position. Again in this study correlation coefficients for the ANN and multiple linear regression (MLR) simulated mass spectra were 0.87 and 0.79, supporting the relative performance of the ANN system.

1.3.6 *De novo* prediction of mass spectra

With the advent of modern approximate quantum chemistry techniques (also known as QM) for calculation of molecular properties, it has become possible to predict and compute a wide variety of spectra for reasonably sized chemical compounds (Grunenberg *et al.,* 2010). As previously mentioned, mass spectrometry (MS), more specifically EI-MS, is a vital analytical method in organic chemistry (Gross, 2011). Recently, the Quantum Chemical Electron Ionization Mass Spectrometry (QCIEMS) prediction method (Bauer & Grimme., 2014; Bauer and Grimme, 2016) has appeared in the literature as a successful new method in the theoretical prediction of mass spectra.

The QCEIMS is a fully automated procedure that uses AIMD (with appropriate stochastic and statistical elements) in order to accurately predict EI mass spectra (EI-MS). The QCEIMS works with several third-party electronic structure programs such as MOPAC, ORCA and TURBOMOLE; these allow the atomic forces required by the QCEIMS internal MD procedure to be calculated with various semi empirical methods (Asgeirsson *et al*., 2017)

A study by Bauer & Grimme in 2014 explored the use of this method for a set of organic drug molecules, and compared the QCEIMS spectra of these compounds to experimental

spectra. The resultant correlation of the first principles spectra provided evidence towards this method being workable for medicinally and pharmaceutically relevant organic compounds. In short, the QCEIMS as applied is based on BornOppenheimer *ab initio molecular* dynamics (BO-AIMD), and uses this to compute the fragmentation pathways of a molecule that is energy-rich at an elevated temperature closely modeling the fragmentation of raised energy ions that are observed after ionization that is caused by bombarding molecules with electrons in the gas phase. Molecules were optimized using dispersion-corrected density functional theory (DFT) at the TPSS-D3/def2-tzvp level as implemented in the Turbomole software. The nature of the stationary point on the potential energy scans was confirmed to be a local minimum by calculating the harmonic vibrational frequencies. The QEIMS program was used on these structures with an impact energy of 70eV. The IEE distribution was computed according to a Poisson energy distribution with the greatest possible IEE being 70eV (Bauer & Grimme, 2014).

1.4 Computational Techniques

1.4.1 Molecular Mechanics

Conformation is important in the preparation of molecular systems in this work, and conformations may be very rapidly generated and compared using a fast computational method called Molecular Mechanics.

Molecular mechanics (MM) is a classical description of molecular and supra-molecular systems, and may be used across many orders of magnitude of size of systems, from low molecular weight molecules (e.g. hydrocarbons) to large complexes (e.g. proteins, nucleic acids and also membrane fragments) or material assemblies with a large number of atoms (Poltev, 2015).

MM force fields are used in many cases to approximate the quantum mechanical potential energy surface, and in so doing decrease the computational cost on large system by orders of magnitude (Vanommeslaeghe *et al.*, 2015). One advantage that the MM potential energy function has over standard QM functions is that MM provides a more accurate representation of dispersion interaction (which QM calculations can only address at the expensive MP2 and higher levels of theory (Hobza & Sponer, 1999), although this is starting to change with the emergence of dispersion-corrected functionals such as ωB97-XD (Chai & Head-Gordon, 2008)

MM is sometimes used in experimental studies (e.g. in the protein databank, the PDB) (Berman *et al.*, 2003) and Nucleic acid databank (Berman *et al*, 1992); for both of these experimental data is refined using MM methods). As mentioned, MM-like semi-empirical terms are now used in some quantum mechanics models, from simple addition of a dispersion term applied to all pairs of atoms in DFT-D and DFT-D2 methods to more complex arrangements where all triples of atoms are concerned DFTD3 (Grimme 2004; Jurecka *et al.*, 2007). MM excludes electrons explicitly, and therefore cannot be used to calculate properties of molecules that are directly related to electrons in orbitals.

Quantitative estimations for molecular properties through simple atom level classical mechanics representations began in the early 1940s, and these estimations were concerned with the conformations of organic molecules. The mathematical expressions of potential energy were suggested in the work of Hill in 1946 (Polten, 2015) and these expressions already included information present in modern molecular mechanics force fields such as stretch and bend terms for bonded interactions and Lennard-Jones terms for non-bonded atom interactions. In the 1960s there was an increase in the MM methods; this increase in use of MM was driven by the introduction of computers into all branches of natural science (Polten, 2015).

MM methods (also referred to as empirical force field methods) depend on the idea that the conformation of the molecules, and the energies of those conformations, can be described reliably with a simple model which draws on concepts from classical mechanics (Boeyens, 1885; Boeyens, 2001). MM focuses on the motion of the nuclei while the molecule is treated like a set of interacting atoms; the molecular potential energy surface is calculated from a series of terms that depend on each atom's position in relation to other atoms in the molecule.

A force field for MM calculations contains a set of potential energy functions that describes all the bonded and non-bonded interactions between atoms of a molecule, and also gives parameters that define the atom interactions depending on the atoms interacting (thus a C-C bond has different parameters in terms of an equilibrium distance and a bond spring constant that is different to, say a C-O bond or a C=C bond). The total energy of a typical force-field is the sum of bond length deformation, angle and torsion angle deformation, non-bonded interactions, electrostatic interactions and out-of-plane deformation which are represented in the following equation 1 (Bowen ***et al.,*** 1991):

$$\cup_T = \sum_{i=i}^{m} \cup_B + \sum_{i=1}^{n} \cup_\theta + \sum_{i=1}^{p} \cup_{NB} + \sum_{i=1}^{r} \cup_q + \sum_{i=1}^{w} \cup_{\theta OPP} \qquad \text{Equation 1}$$

There are several force fields available for performing molecular mechanics calculations, each with a different set of parameters and with slight changes to the total energy term. The CHARMM software program (chemistry at HARVARD using molecular mechanics), was initially developed in the 1980s. At that time CHARMM force field that was used contained parameters but not for explicit hydrogen atoms. The CHARMM19 force field, developed around 1985, started to include explicit hydrogens, but only in situations where hydrogen bonding was important; i.e. hydrogens bonded to nitrogen and oxygen (Reiher, 1985; Neria *et al.,* 1996). The parameters are constantly undergoing revision, so the updated CHARMM22 force field was released with the 1992 version of the CHARMM software (Ponder & Case, 2003). The most recent versions of the CHARMM force field include parameters for all hydrogen atoms. Parameterization of this force field is dependent on the use of approximate quantum mechanics calculations to determine the form of the potential energy surface, and explicit water in these calculations is necessary for the development of accurate parameters. In terms of the most recent versions of the CHARMM force field, equation 2 shows the exact form of the energy function used within the latest CHARMM36 force field.

$$E_{total} = \sum_{bonds} k_b (r - r_{0,b})^2 + \sum_{angles} k_a (\theta - \theta_{0,a})^2 + \sum_{dihedrals} k_{d,n} [1 + \cos(n\chi - \delta_{d,n})] + \sum_{non-bonded} \epsilon \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right] + \sum_{non-bonded} \frac{q_i q_j}{\epsilon_1 r_{ij}}$$

Equation 2

In equation 2 for the CHARMM36 energy calculation, the first three summations describe the form of the energy for bond stretching, angle bending and changes in torsion, and these calculations are summed over all bonds, angles and torsions in a molecule. The respective *k*'s are the force constants, while the remainder of each term describes the deviation from an ideal equilibrium value. The final two summations are for the non-bonded terms, namely the van der Waals and electrostatic interactions, and these are

dependent on the interatomic distance r_{ij} between pairs of atoms *i* and *j* (Moses *et al.*, 2017).

A second example of a force field is the GROMOS force field, and this has gone through several revisions starting with the 26C1 GROMOS force field (1981) extending through to version 54A8 (2012). In the latest version of the GROMOS force field, parameters are categorized according to the kind of (sub)molecule the specific parameter applies to. It is interesting to note that the first GROMOS force field (26C1) included only 26 atom types, and this force field could only accommodate the simulation of proteins (GROMOS volume 3, 2017).

1.5 Quantum Mechanics

Quantum mechanics calculations (QM), also referred to as quantum chemistry refers to approximate methods to solve for the energy of a molecule based on solution of the Schrödinger Wave Equation (SWE). Through the solution to the SWE, it is possible to calculate properties of an isolated physical system at any instant in time (Oliveira, 2007). QM methods may be used to determine the physical properties systems like atoms, molecules, and condensed phases and is a more appropriate method for determining properties of systems on the scale of atoms and molecules than classical mechanics (Yehuda & Yshai 2013). One of the strengths of QM is that after calculation, there is detailed information available with regard to the electronic structure of chemical compounds; it is this electronic structure that enables understanding of the compound in terms of its reactivity and kinetics thermodynamics and molecular properties (Gupta, 2016). QM is far more general than MM, and has been applied in an approximate manner to fields such as particles, condensed matter, nuclear and atomic physics.

The approximate QM method that is mostly used in applications related to biological systems or large molecular complexes is density functional theory (DFT) (Dreizler & Gross 1990). DFT is a tool used to complement experimental investigations and also to

predict (with reasonable accuracy, and not hugely expensive computational cost) many molecular properties such as geometries, reaction pathways, and spectroscopic and mechanistic properties (Wawrzyniat *et al.*, 2008; Alia *et al*., 2009).

The interactions between electrons determines the structure and properties of matter from molecules to solids. In order to describe the interacting electrons, the three-dimensional electronic DFT (Kohn, 1965; Parr & Yang, 1989) in many cases removes the need for the calculation of a complex manydimensional wave function. Kohn noted that DFT has been very useful for systems with many electrons where wave function methods are too computationally expensive (Kohn, 199).

However, there is still a computational cost associated with DFT. Sturniolo *et al.,* conducted a study to compare DFT to a more approximate method known as density functional tight binding (DFTB) (DFTB+ was used) approach and these two approaches were observed to give similar/same results (Sturniolo *et al.,* 2018). DFTB is an electronic structure method that makes use of the Kohn-Sham approximation to solve the quantum many body problems for electron just like DFT. However, DFTB approximates it in such a way that only electrons are represented. The calculation does not give information about the full spatial wave-function, and this is the source of its relative computational efficiency as compared to DFT. The DFTB method is not strictly an *ab initio* technique since it uses empirical parameterizations that are computed from pure DFT calculations. These parameterizations specifically describe interactions between pairs of chemical species (Porezag *et al.,* 1995). DFTB+ calculations are much computationally cheaper compared to DFT calculation and they hold the potential for treating large organic systems like proteins or polymers (Sturniolo *et al.,* 2018) DFTB+ is also the name of a fast and efficient quantum mechanical simulation software package that implements the DFTB method. It offers this approximate DFT based quantum simulation with functionalities similar to *ab initio* quantum mechanical packages, but with the advantage of speed of calculation. The DFTB+ software is capable of optimization of structures of molecules and solids, the calculation of one electron spectra, and even the calculation of electron transport under nonequilibrium conditions (Aradi *et al.,* 2007; https://www.dftbplus.org/about-dftb/).

1.6 Molecular Dynamics

Molecular dynamics (MD) is a simulation technique where atoms have kinetic energy appropriate to a molecule at a particular temperature. The velocities and accelerations of all atoms follow Newton's laws of motion. The calculation of forces may be at the QM or the MM level. In any case MD simulations are dependent on an initial set of conditions, a good model to represent the forces acting between the particles (either from electronic-structure calculations or using the empirical force fields i.e. QM or MM), and in many cases are also dependent on sets of boundary conditions. When these conditions are met, the main task remaining during MD is the solution to the classical equations of motion.

1.6.1 Molecular Mechanic based Molecular Dynamics

When the forces in molecular dynamics are calculated at the molecular mechanics level, simulations of extremely large systems becomes possible.

To illustrate the size of the systems accommodated, one has to simply look at molecular dynamics studies of proteins. The study and prediction of protein-ligand and protein-protein interactions has become very important molecular biology and the protein three dimensional (3D) structures are also important for structure based drug design. Since proteins also nucleic acids are highly flexible, molecular dynamics can provide insight into their functionality (Adam *et al.,* 2015). Since molecular dynamics (MD) is a physics-based modeling method that provides detailed information about the conformational and fluctuation changes of atoms and molecules in a system (Fadeel *et al.,* 2012), MD simulations can be used to describe the strength, properties and patterns of protein behaviour, drugreceptor interactions, the nature of solvation of molecules and also conformational changes of proteins or molecules during the simulation (Vlachakis *et al.,* 2014). In terms of conformation, during MD the energy surface is traversed through the solution to Newton's laws of motion for the system (Leach, 2007).

MD methods were initially developed in the late 1950s by Alder and Wainwright, where the study treated interactions as those of hard spheres. In 1964 Rahman made the first simulation using a realistic potential for liquid argon (Rahma, 1964) and the first MD simulation of a molecule in a solution of water was first performed in 1974 (Stillinger & Rahman, 1974). The first simulation of a protein was first done in 1977 using bovine pancreatic trypsin inhibitor (BPTI) (McCammon *et al.,* 1977; Warshel Levitt, 1976) and the use of MD simulations has advanced from simulating hundreds of atoms to systems that have biological relevance which includes proteins in solution (Roccatano *et al.,* 2007).

MD simulations based on MM can be easily performed on systems with up to 100 000 atoms, and this capability increases to systems with 500 000 atoms when appropriate computer facilities are used, such as high performance computing (HPC). When running MD simulations, the initial model of the system is obtained from either experimental structure or comparative modeling data, and the simulated system may be represented at different levels of detail. Many systems may be represented at the atomistic level (which is as close to reality as possible), but for extremely large systems or long simulations, coarse-grained representations can be used, where groups of atoms are treated as single entities during simulation (Orozco *et al.,* 2011; Lazaridis & Karplus 1999).

Initial conditions for MD include the initial velocities of every atom in the system; although some information on atom velocities may be deduced from crystallographic files, it is more general that velocities of atoms are randomly selected. In MD the velocity for the particles in a system are normally randomly attributed to the Maxwellian distribution centered at a desired temperature and adjusted such that the system has zero angular momentum (González, 2011). At each point during simulation, forces acting on each of the atoms in the system must be obtained (in this case from forcefields), where the potential energy is deduced from the current atomic positions (Hermans *et al.,* 1964).

For many systems it is necessary during simulation to define periodic boundary conditions (PBCs). With PBCs the simulation box is surrounded by multiple replicas of itself, only

atoms inside the main cell are considered but molecules and atoms are free to move beyond the limits of the cell; any atom leaving the cell results in an image particles entering the cell on the opposite side of the cell to replace it as illustrated in Figure 1.4.

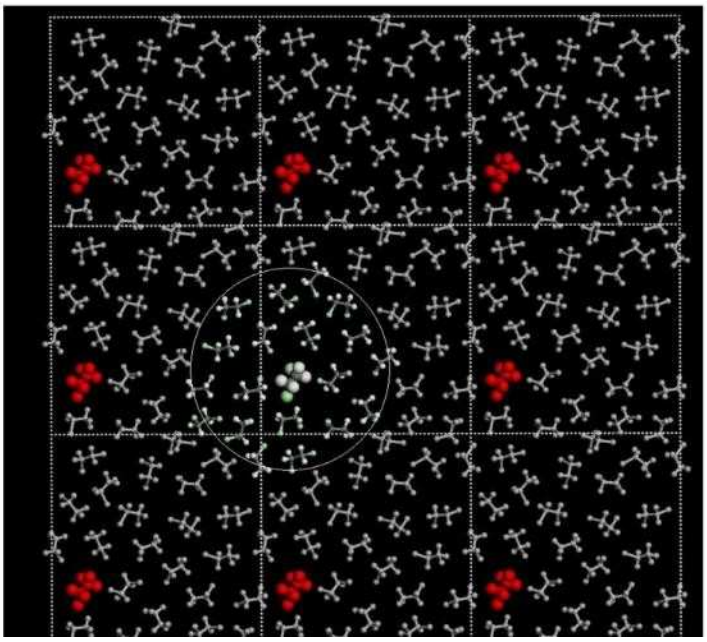


Figure 1.4: Schematic representation of PBCs in 2D

After setting initial conditions, all forms of molecular dynamics (including *ab initio* molecular dynamics mentioned below) follow the same set of procedures. After setup, the velocities of the ensemble are increased until the simulation reaches the desired temperature (heating phase). Next molecular dynamics is allowed to proceed at the desired temperature but properties are not recorded during this phase (the equilibration phase). It is best practice to equilibrate a system since the initial configuration will include artefacts of construction, such as close contacts upon solvation. Following this molecular dynamics continues (after the system has reached a steady state in terms of properties) into the production phase of dynamics. It is this production phase that is used for analysis and visualization. A useful tool for molecular dynamics visualization is the program visual molecular dynamics (VMD). Some of the properties of macromolecules that are typically measured during production dynamics are the root mean square deviation

(RMSD), root mean square fluctuation (RMSF) and the radius of gyration. (González, 2011).

RMSD is the most used quantitative measure to check the similarity between the atomic coordinates of two superimposed conformations of a single protein, for example. It is possible to calculate the RMSD for a subset of atoms; for instance with proteins it is typical to consider the Cα atoms throughout, Cα atoms residues in a specific subset, all heavy atoms of a specific subset of residues, or all heavy atoms in a small-molecule ligand present during simulation (Kufareva & Abagyan 2012).
RMSD is an important check in the analysis of the time-dependent motions of the structure. It provides information regarding the stability of a structure during the time-scale of the simulations. In most cases if RMSD increases during production dynamics, it is an indication that the equilibration step has not proceeded for sufficient time, and the system is therefore not equilibrated.

RMSF is the measure of displacement of an atom or a group of atoms relative to the reference structure averaged over a number of atoms. When the simulation is equilibrated, the fluctuations of each subset of the structure relative to average structure of the simulation the RMSF are computed (Martinez, 2015)

Radius of gyration it is a parameter used to describe the equilibrium conformation of an entire system and for large systems such as proteins can be used to explore their compactness or folding during simulation (Lobanov *et al.*, 2008).

1.6.2 *Ab Initio* Molecular Dynamics

Ab initio molecular dynamics (AIMD) is different from molecular dynamics using molecular mechanics since all calculations of forces come from QM calculations, although propagations of velocities will be based on Newton's equations. As such AIMD is based on real physical potentials whereas MD is based on semi-empirical effective potentials that approximate the real QM potentials (Paquet & Viktor, 2018). AIMD

calculations assume (for the QM calculations) that the system is composed of N nuclei and N_e electrons (as is usual for QM), that the Born-Oppenheimer approximation holds, and that the nuclei dynamics can be treated classically (using classical dynamics) on the ground state electronic surface (Radu *et al.,* 2005). During AIMD, dynamical trajectories are produced using forces that are obtained directly from these electronic structure calculations, done at each point or frame of the simulation, meaning that AIMD accounts for electronic polarization effects, but further, allows chemical bond breaking and formation to take place (Marx & Hutter, 2000; Tuckerman 2002). In the current study we have conducted AIMD calculations on five different compounds including alpha hispanolol, boronolide and PFB-oxime, following bond breaking and forming, and identifying fragments produced. As will be seen later in Tables 2.1, 2.2 and 2.3 AIMD of different conformations of the same molecule resulted in different fragmentations, and this was used to build up theoretical mass spectra.

1.7 Natural Product Databases

Natural products (NP) have a rich source of high chemical diversity that provides a basis for the identification of novel scaffold structures in many situations, such as for rational drug design (Koehn & Carter 2005). Many natural product-based drugs have been developed over the years; approximately 47% of drugs used in cancer treatment are either natural products or their derivatives (Newman & Cragg, 2012). About 19 natural product-based drugs were approved between the year 2005 and 2010 (Mishra & Tiwari, 2011).

Sources of natural product database are historically based. For example, over several thousand years, the traditional chinese medicine (TCM) has been used to treat and prevent a number of diseases (Li *et al.,* 2018). As such natural compound databases including compounds extracted from traditional chinese herbs and medicines have been created to assist with tasks such as *in silico* drug discovery (Hatherley *et al*., 2015). Several databases specializing in traditional chinese medicine (TCM) are now available and these include Hou's CTM (Hou *et al.,* 2001) database that maintains a full set of 3D structures

of compounds isolated from TCM, together with properties that include drug-likeness and clinical effects (Shen, 2012). Another database, database@Taiwan (Chen, 2011), contains 24 000 structures of compounds isolated from 453 chinese herbs. There are also other chinese databases such as HIT database, TCMID database and also TCMSP database (Li *et al.,* 2018).

Of relevance is the study of natural NP in Africa and the African NP databases that are available. These include ConMedNP which has the compounds extracted from central Africa (Ntie-Kang *et al.,* 2014), SANCDB (Hatherley *et al*., 2015) and Afrocancer which consists of compounds extracted from South Africa (Zeng *et al.,* 2018). At its inceptions in 2015, the SANCDB database contained 600 compounds extracted from 143 different South African source organisms (Hatherley *et al*., 2015). In this work, the diversity of chemical structure in the SANCDB provides a source of compounds from which to test mass spectral prediction for a range of chemical space. Further, an advantage of successful spectral mass prediction for compounds such as are in these databases, is the aid of this to natural product discovery.

1.7.1 α-Hispanolol

One of the main compounds studied in this thesis in terms of theoretical mass spectra was α-hispanolol (Figure 1.5). Since a large physical sample was available, the reasoning was that experimental EI mass spectra could be obtained for this compound under a wide variety of experimental conditions; this in turn would inform the conditions of calculation of predicted mass spectra. Hispanolol contains aromatic (furan) moieties and cyclic aliphatic systems which make prediction of mass spectra interesting, since bond cleavage will not always produce separate fragments. In addition to this reasoning, the closely related hispanolone is also present in the SANCDB. Hispanolone was first extracted from *Ballota hispanica* and has been identified in other *lamiacaea* species which include the endemic Southern African medicinal plant *B. africana* (Van Wyk, 2005). In terms of usage, hispanolone has been found to have low cytotoxicity and anti-inflammatory activity in the *in vivo* model of the 12-O-tetradecanylphorbol-13-acetate (TPA)-induced ear edema assay (Nieto-Mendoza *et al.*, 2005). Hispanolone derivatives have been

reported to have anti-inflammatory effects via the activation of nuclear factor-kB (NF-kB) inhibition in 2PS-activated RAW 264.7 macrophages and inhibiting myeloperoxidase activity, an index of neutrophil infiltration (Giron *et al*., 2008). Some of the derivatives have antitumoral effects by activating apoptotic cell death machinery (Traves *et al.*, 2013). α-hispanolol itself has been found to lower cell viability in some tumor cells, having no effect on normal cells.

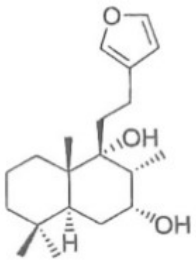


Figure 1.5: Molecular structure of α-hispanolol.

1.7.2 Boronolide

Boronolide is a natural product that was extracted from the bark and branches of *Tetradenia fruticosa* and also the leaves of *Tetradenia barberae* that has been in South Africa and Madagascar as a traditional medicine. Boronolide has a polyacetylated side chain and an α,β-unsaturated-δ-lactone moiety (Lin *et al.,* 2010). Di-deacetylated boronolide have been extracted from *Tetradenia riparia* which is found in Central Africa and has been used by the Zulu people as an emetic that is an infusion of the leaf has been reported to be effective against malaria (Naidu *et al.,* 2005).

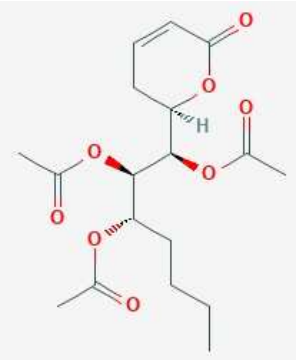


Figure 1.6: Chemical structure of boronolide.

1.7.3 PFB oxime

In a study of secondary metabolites which are produced from South African marine algae, a collection of *Plocamium corallorhiza* was investigated from the Southeast coast of South Africa and this collection had produced a number of unstable halogenated monoterpene aldehydes which were not detected in the west coast collection. A major aldehyde metabolite had been isolated as an unstable, optically active oil that quickly degraded when exposed to air and room temperature. When this compound reacted with pentafluorobenzylhydroxylamine hydrochloride, it produced PFB-oxime derivative (Figure 1.7) (Mann *et al.,* 2007). This compound was interesting to this study for mass spectral prediction in terms of the presence of unsaturation, and the high level of halogenation.

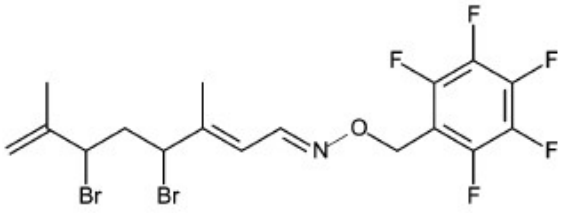


Figure 1.7: PFB oxime.

1.8 The NIST mass spectral databases

The NIST mass spectral database contains experimental mass spectral data and also provides software tools that help in compound identification. Within this database are reference mass spectra for small molecule and peptide compounds from GC/MS (by electron ionization) and LC-MS/MS (by tandem MS). In terms of the software tools, specifically these are data analysis tools including the Automated Mass Spectral Deconvolution and identification system (AMDIS) for GC/MS. There is also a fully functional version of NIST MS program Version 2.3 with a small demonstration library available (https://chemdata.nist.gov)

1.8.1 Compounds Used from NIST in this study

Two compounds were studied from NIST database, these compounds are ethyl 3-bromo-4,6-dimethyl-2H-pyran-5carboxylate (Figure 1.8) and 5-amino-1-(phenylmethyl)-[1,2,3]triazole-4-Carboxamide (Figure 1.9). The diverse chemical functionality made these attractive targets for mass spectral prediction, together with ready availability of experimental mass spectral data.

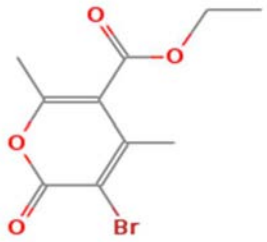


Figure 1.8: Ethyl 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate.

Figure 1.9: 5-amino-1-(phenylmethyl)-[1,2,3]triazole-4-carboxamide

1.9 Molecular Docking

Given that natural products may be too large to be used for targeting particular disease targets, mass spectral fragmentation provides a convenient access to smaller systems. As an aside, in this project fragments will be docked against two disease targets; focus is not on the search for new lead compounds, but rather as a proof of concept of how the impact of theoretical mass spectra can be extended.

Molecular docking is a computational technique used to predict the atomic coordinates of a protein-ligand complex (Brooijmans & Kuntz, 2003; Rognan, 2017). The molecular docking approach is used extensively in the search for new lead compounds in drug discovery (Chen *et al.,* 2013).

The main aim of the molecular docking technique is to predict the best matching binding mode of a ligand (docking pose) to a macromolecule (e.g. proteins). Generally there will be many poses generated which will then be scored to identify the best pose, together with an indication of the interaction energy. For docking calculations it is necessary for there to be a 3-dimensional model of the molecular target (Salmaso, 2018).

There are therefore two stages in molecular docking, sampling and scoring. The sampling process should thoroughly search the conformational space, while the scoring function must be able to accurately predict the binding energy for the interaction.

1.10. Aims of the Project

The aim of this project is to predict the mass spectra of SANCDB and NIST compounds using a modified QCEIMS method. In order to achieve this the following aims must be met:

1. A workflow, controlled by a python script, must be able to generate a series of different ab initio molecular dynamics trajectories from a single input structure. This script at a minimum must produce input files for CP2K in order for CP2K calculations to produce the different trajectories. There must be some control within the script regarding the number of trajectories to be run, and also aspects of the trajectory such as temperature of simulation and initial atomic velocities.
2. From the trajectory outputs, an analysis script must be able to do several things
    a. Identify where bond breakages occur
    b. Where bond breakages do occur, the script must be able to separate the formed fragments and specify which atoms from the original molecule are in each of the fragments.
    c. Given knowledge of the fragments from the molecular dynamics trajectory, the script must parse the log file for details of charge and spin density calculations, to determine the fragments that will appear on the mass spectrum.
    d. Also, given knowledge of the fragments and the proportions of elements in the fragments, the script must predict the isotopic distributions from fragments that should appear in the mass spectrum.
    e. Finally the script should collate all of this information from all trajectories to produce a final mass spectrum.
3. As an aside, it will be of use to see if fragments produced by fragmentation may be used in drug discovery workflows.

1.11 Objectives of the Project

To obtain the theoretical mass spectra data of several products in a workflow that builds up to a QCEIMS implementation, using Born-Oppenheimer molecular dynamics in the context of the CP2k software at the DFTB+ level of theory.

To compare the theoretical spectra obtained with experimental mass spectra, where this data is available.

To test the implementation in terms of the quality of spectral generation, computation time and robustness in terms of initial conditions.

CHAPTER 2

2 Mass spectra prediction

2.1 Literature version of QCEIMS

The QCEIMS method was constructed based on the observation that although the capabilities of modern quantum chemistry, particularly in the context of DFT, are remarkable in terms of predicting properties and spectroscopy of molecules, there was still a gap in the prediction of mass spectra (Grimme, 2013). The prediction of mass spectra using QCEIMS BOMD approach method occurs through the following approach. Firstly molecular dynamics is performed on a neutral molecule in the ground state, and this provides starting points for the molecular dynamics of the ionized species. From this trajectory, an ensemble of instantly ionized species across a range of randomly chosen conformations is generated. Ultimately ionization leads to species with an "impact excess energy", which is typically up to 50 eV, and this ensemble will have a distribution of energies. For reproducible spectra, 100s of trajectories are required for a total simulation time of about 10ps, and this will involve 10^6-10^8 separate steps of calculation. From the trajectories charged fragments are identified and recorded, isotopic distributions are calculated and the spectrum is thus generated. Figure 2.1 summarizes this process. (Bauer & Grimme, 2014).

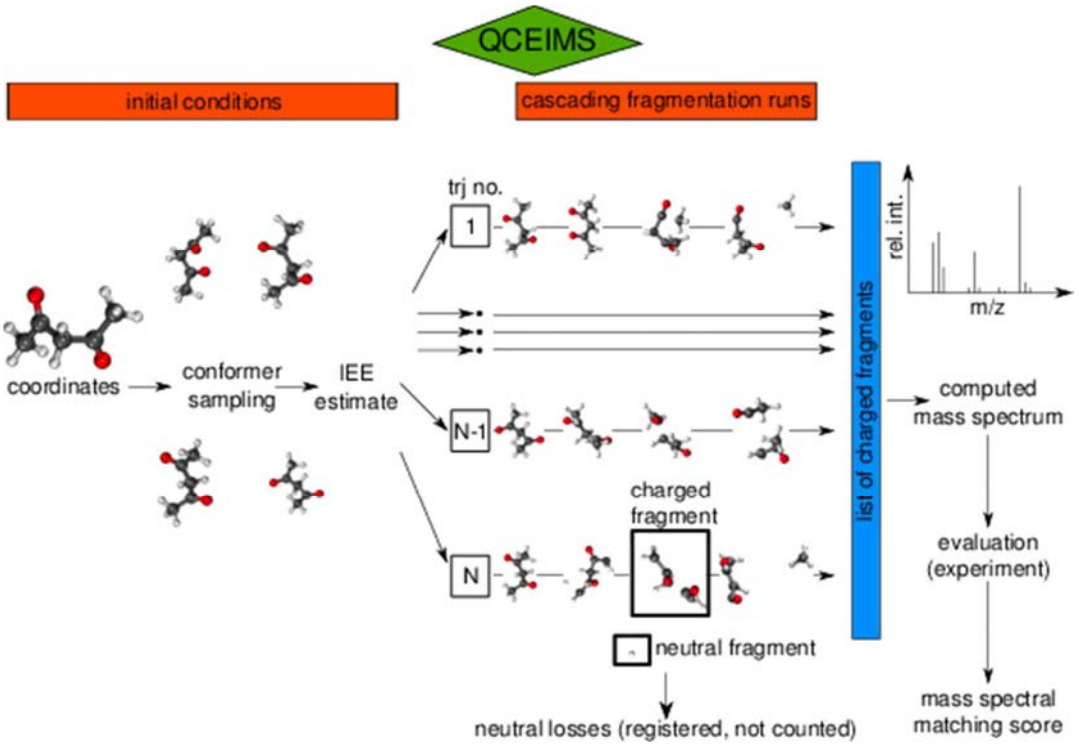


Figure 2.1: An overview of how a mass spectra is predicted using the QCEIMS procedure

In Figure 2.2, a different diagrammatic perspective on the approach is provided, together with experimental and theoretical spectra overlay, illustrating the success of this approach.

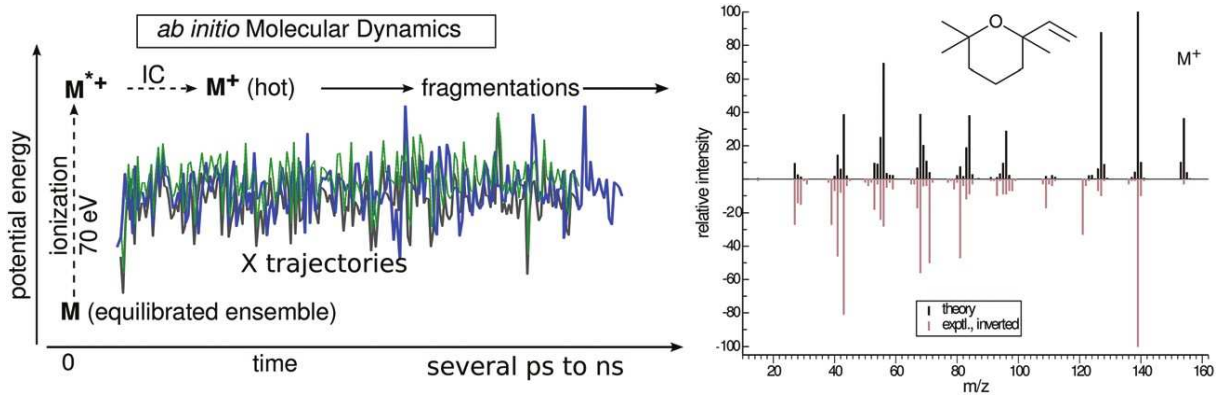


Figure 2.2: Description of the QCEIMS procedure with mass spectra result for a small molecule (both the theoretical and experimental mass spectra are illustrated) (Bauer & Grimme *et al.*, 2014)

2.2 Methodology

2.2.1 Software and python libraries used in this study

RDKit provides a wide range of chemoinformatic library capabilityIt is an open source toolkit for chemoinformatics, easily used in this project within python. RDKit itself is constructed in C++. Some of the advertised capabilities include python, java and C# wrappers, allowing RDKit to be used beyond C++, a wide range of descriptor generation tools, 2D and 3D molecular operations, and chemoinformatics nodes for use within KNIME software. RDKit was developed in the early 2000s for use in rational drug discovery and for building models to predict ADME, TOX and biological activity. Development is active on this open source project, that comes with a BSD license (Landrum 2019). RDKit integrates with many other open-source projects (KNIME, postgreSQL, Ipython, Pandas, and Lucene).

In order to be able to simulate systems at the molecular and atomic level, CP2K is an appropriate simulation tool for this purpose (http://www.cp2k.org/, (2012)). CP2K can perform a wide range of QM-type calculations including semi-empirical approaches based on the NDDO Hamiltonian (AM1, MNDO, MNDO/d, PM3, PM6) and DFT methods (Porezag *et al.,* 1995; Murdachaew *et al.,* 2011). This is in addition to the wide range of force field calculations that are accessible to this software (Hutter *et al.,* 2014). In terms of classical force fields it is compatible with CHARMM, AMBER and GROMOS force fields and is able to read the corresponding topology files (Foiles *et al.,* 1986). Further to this CP2K is able to perform mixed QM/MM simulations (where different parts of the system are treated at different levels) and also adaptive partitioning QM/MM based dynamics, where parts of the system that may be under movement are allowed to change from a QM based description to an MM based description, depending upon their localization within the system. The adaptive partitioning is particularly useful in the case of solvent. There are a very broad range of types of problems that CP2K can

solve, beyond simple energy, force and electronic structure calculations. In the context of this project, CP2K is capable of performing MD simulations at many levels, including at QM, MM and mixed QM/MM levels.

Networkx (https://networkx.github.io/) is a powerful tool for dealing with networks in particular. In future work, where pathways between fragmentation are to be mapped and explored, the networkx python libraries are anticipated to be of incalculable use. In this study networkx was used in the context of graph theory, since by setting up a simple map of a molecule and defining which bonds were to be broken, networkx is quickly able to provide a list of the separate fragments, detailing the nodes belonging to each fragment graph.

2.2.1 Approach used in this study, deviation from literature

Due to the complexity of setting up a distribution of ion energies, as a starting point this study does not generate conformations through a molecular dynamics approach on the unionized molecule, but starts with molecular dynamics on the ions. The omission of this step at this stage will not affect the subsequent analysis scripting when this step is fully included. What it does mean is that at present the trajectories are initiated from the instantaneously ionized system, and that heating is still required to bring the ions to temperature. All trajectories are brought to the same temperature (so no ensemble has yet been included in this approach), but the collation of charged fragments from the trajectories and generation of the mass spectra are followed exactly. Due to time constraints, the evaluation metric for the fit between theoretical and experimental spectra was also not included.

2.3 Detailed scripting, methods and results

2.3.1 Initial construction of molecules and determination of trajectory conditions

All structures in this study were constructed and cleaned as 3-dimensional models, with all hydrogen atoms modeled explicity, using Discovery Studio Visualizer.
Initial testing involved simple scripts that took in a mol2 structure and created a working CP2K input script that could be passed on to CP2K for calculation. For this purpose a rudimentary python library ("cp2k") was written to simply print out default CP2K parameters to a CP2K input file, while transferring across appropriately the atomic coordinates.

As a test with hispanolol, BOMD was performed at 2000K for 10000 steps of dynamics in duplicate, and the RMSD of each frame between trajectories was calculated. The result of this was that both trajectories were identical in energy and geometry for each of the 10000 steps. The BOMD as implemented in CP2K is not truly stochastic, so multiple trajectories with identical starting conformations will produce identical fragmentations. The introduction of random initial velocities upon writing the CP2K input file did allow for stochasticity, and when the desired initial set of conformations was used, fragmentation no longer was reproducible between pairs of runs.

Most molecules exists with multiple conformational isomers that have the same chemical bonds (and stereochemistry) although their 3D geometry is different since they hold different torsion angles (Crippen & Havel, 1988). The conformation of a molecule has an effect on its chemical reactivity, molecular binding and biological activity. The stability of different conformations of a molecule is not the same because they experience different steric, electrostatic and solute-solvent interactions (Gaalswyk & Rowley 2016). The RDKit cheminformatics tool can generate conformers, including finding the lowest energy conformers that are structurally similar to the experimentally determined structures, very efficiently (Jerome *et al.*, 2016). In the current study we used RDKit to

generate random conformations of our molecules prior to running *ab initio* MD (AIMD) simulations using these conformations.

We could now test more generally the production of BOMD trajectories. To this end ten random geometries of α-hispanolol were generated using RDKit on the local departmental servers. From these ten geometries ten CP2K input files were generated for all of these using a python script for AIMD simulation using random velocities that were between 0-0.0001 bohr/time in atomic units (au.time). The DFTB+ level of theory was chosen and the temperature was set to 2000 K for 10000 steps of dynamics. 2000 K was an arbitrary starting point, but one that assured generation of fragments and therefore allowed for python code that was dependent on this fragmentation to be written. In terms of feasibility of such a temperature, one can consider a molecule with, for example an excess energy of 2 eV following ionization. This translates to energy of 3.2×10^{-18}J per molecule; if all of this energy is kinetic energy it translates to a temperature of about 1.5×10^{4} K. The time step for all the MD runs that we used was 0.2fs for 10 000 steps. Each trajectory was performed using a single node (with 8 cores on the local cluster, 24 cores at the CHPC). Figure 2.3 shows the code used to set this up, where cp2k.write_cp2k is the library routine created to write the set of directives for the CP2K input file together with atomic coordinates and random velocities.

```
import cp2k
from rdkit.Chem import
AllChem import os
current_dir =
os.getcwd() filelist =
os.listdir(current_dir)

number_of_confs=10

sanccompound =
AllChem.MolFromMol2File('alpha_hisp.mol2',False,False)
ids=AllChem.EmbedMultipleConfs(sanccompound,numConfs=number_of_con
fs)

coun
t=0
for
id
in
ids:
     file_name =
"alpha_hisp_00"+str(count).zfill(3)+".cp2k"
project_name="Traj_16_"+str(count).zfill(3)
file4 = open(file_name, 'w')

cp2k.write_cp2k(sanccompound,id,file4,project_name,2000,10
000)      file4.close()      count=count+1
```

Figure 2.3; Code used to setup CP2K input files for 10 trajectories on 10 conformations.

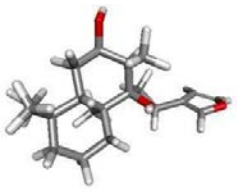
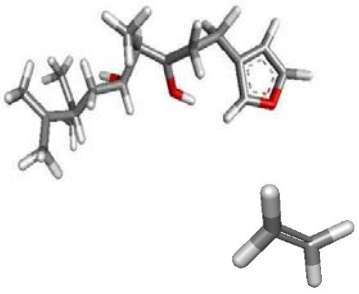
It was worth exploring fragmentation on a small number of systems with limited numbers of trajectories, prior to automating the collection of fragments. As such all of these trajectories were viewed using VMD; to illustrate the fragmentation the coordinates for the last frame of the trajectory was saved as a single pdb structure which was viewed within Discovery Studio. Given that Discovery Studio constructs bonds based on atomic distances, this was a visual way to immediately see the separate fragment structures as a result of the BOMD calculation. The fragmentations of α-hispanolol, boronolide and PFB were very different given the structurally and chemically different starting materials. This provided the basis from which we could construct a python script to analyze all the trajectories from different molecules. With the help of RDkit tools we were able to use the script to see the generated fragments using SMILES. We used RDKit to predict the masses of the fragments that were generated. In order to get the coordinates for the mass

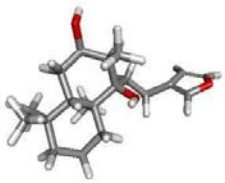
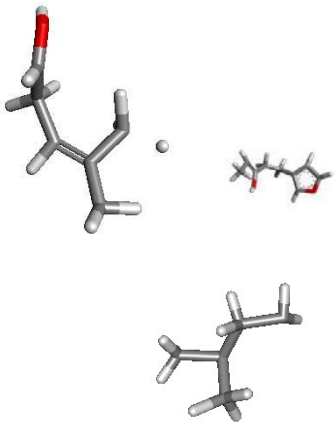
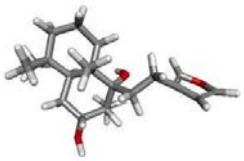
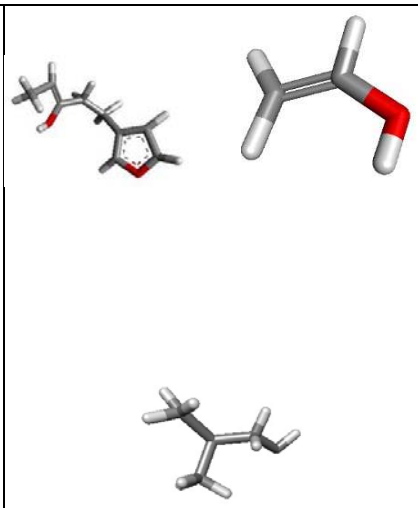
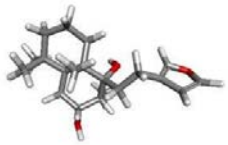
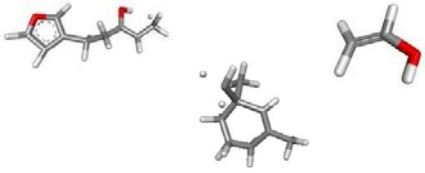
spectra data, we used a counter and looped through the list of the fragment masses and whenever there were same masses we incremented by one. For mass spectra graph we ran one hundred trajectories (meaning we generated 100 conformations) for each molecule so that the produced mass spectra are statistically convergent with respect to the observed fragments.

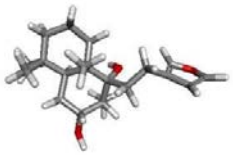
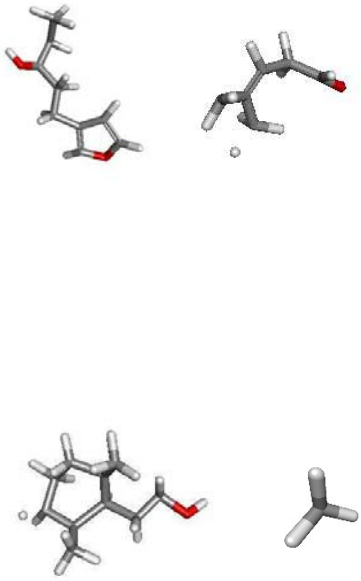
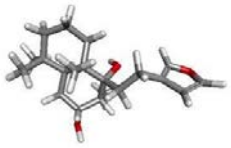
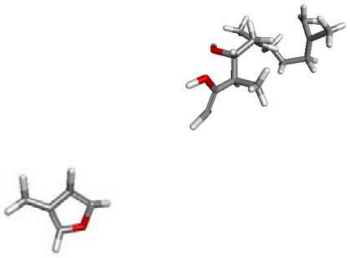
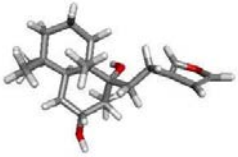
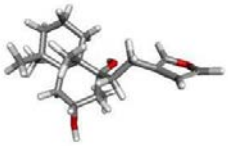
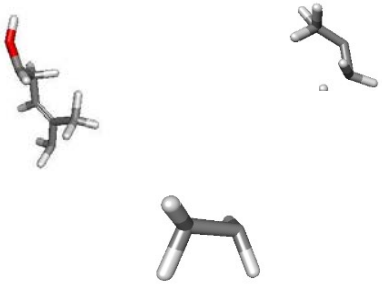
2.3.2 Fragmentation

2.3.2.1 Alpha hispanolol fragmentation

The following table (Table 2.1) details the fragmentation observed from within Discovery Studio Visualizer. In the table the trajectory number, together with the initial conformation (from RDKit) is provided, and the fragments are provided. This is compared to the list of fragments that was generated within our script (SMILES). A discussion on how this list of SMILES fragments was generated appears later, since it was informed by the visual analysis of fragmentation.

Trajectory	Conformation	Fragments Observed (Discovery Studio Visualizer)	List of fragments from RDKit SMILES
0			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']

1			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']
2			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']
3			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']

4			['[H]cc(c([H])c([H])o)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]']
5			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']
6		No fragmentation was observed	['[H]cc([H])c(c[H])C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']
7			['[H]', '[H]', '[H]OC1([H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C2(C([H])([H])[H])C(O[H])(C([H])([H])C([H])([H])[H])C1([H])C([H])([H])[H]', '[H]c', 'c', 'c', 'c', 'o']

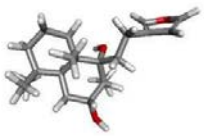
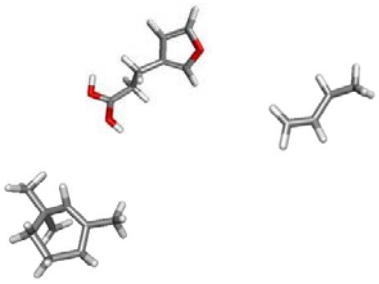
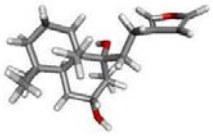
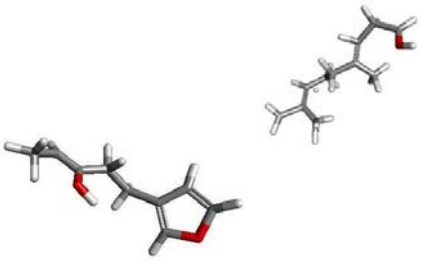
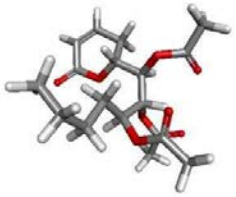
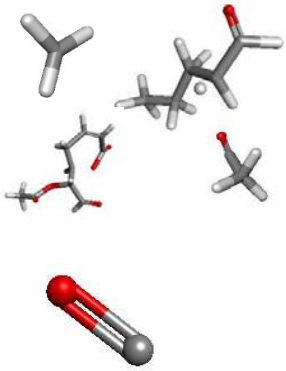
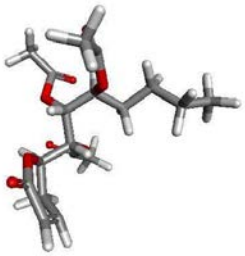
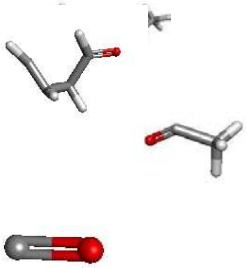
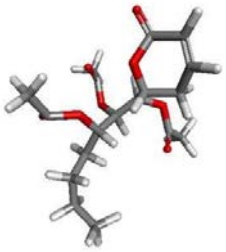
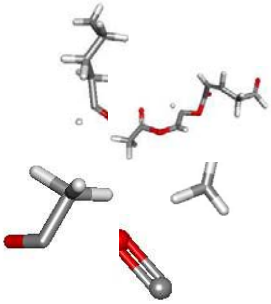
8			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']
9			['[H]', '[H]cc(c([H])c)C([H])([H])C([H])([H])C1(O[H])C([H])(C([H])([H])[H])C([H])(O[H])C([H])([H])C2([H])C(C([H])([H])[H])(C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C21C([H])([H])[H]', 'o']

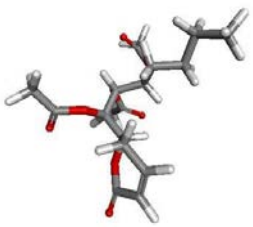
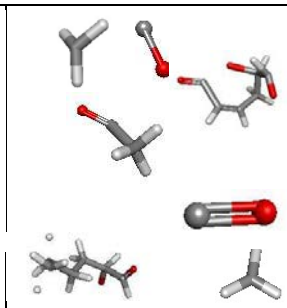
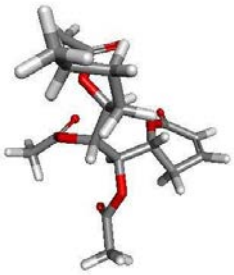
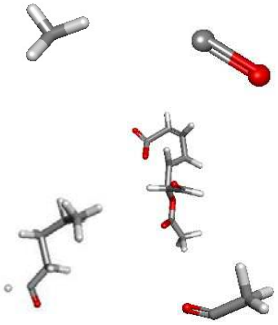
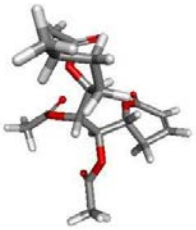
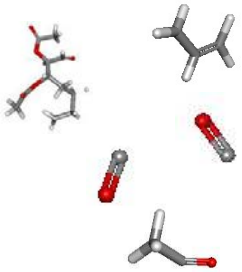
Table 2.1: The different conformations of α-hispanolol together with the fragments observed after *AIMD*. Fragments obtained using RDKit smiles are also presented

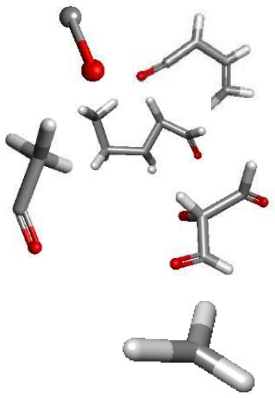
It is interesting to note that in one trajectory (trajectory 6) at 2000 K no fragmentation is observed, and that species expected to have stability are commonly produced, such as ethene (trajectory 0) or the 3-methylenefuran (analagous to the benzilic) and allylic carbocations, observed in trajectories 5 and 8 respectively. In trajectories 1 and 3, for example, hydrogen atoms (which we are considering still bonded to their respective fragments) within Discovery Studio are shown as not being bonded due to an extended bond length associated with dynamics at 2000 K.

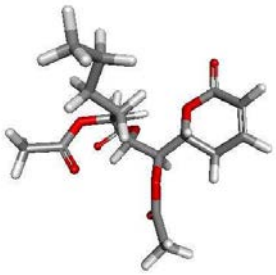
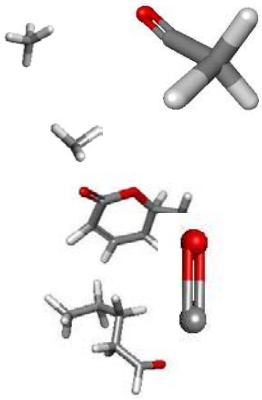
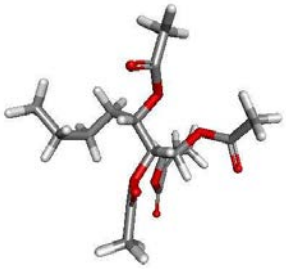
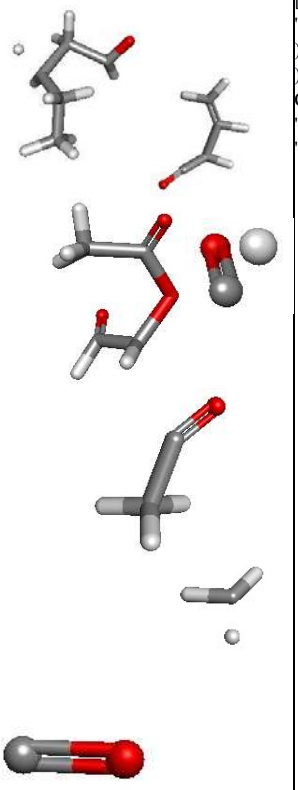
2.3.2.2 Results for Boronolide

The boronolide fragmentation is interesting in comparison, as this is a highly oxygenated system by comparison to hispanolol. This particular system appears to produce a greater number of fragments.

Trajectory	Conformation	Fragments from DS	SIMLES fragments
0			['C=O', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C1([H])OC(=O)C([H])=C([H])C1([H])[H]', '[H]C(O)C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C([H])[H]']
1			['C=O', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])(OC(=O)C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C=C([H])C([H])([H])C([H])O']
2			['C=O', '[H]C(O)C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C(O)C([H])([H])C([H])=C([H])C=O', '[H]C(O)C([H])OC(=O)C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C([H])[H']]

3			['C=O', 'C=O', '[H]', '[H]', '[H]C', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C(O)C1([H])OC(=O)C([H])=C([H])C1([H])[H]', '[H]C([H])[H]']
4			['C=O', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C1([H])OC(=O)C([H])=C([H])C1([H])[H]', '[H]C(O)C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C([H])[H]']
5			['C=O', 'CO', '[H]', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])(OC(=O)C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C=C([H])C([H])[H]']

6			['C=O', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])O', '[H]C(O)C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])([H])C=O', '[H]C([H])C([H])=C([H])C=O', '[H]C([H])[H]']

7			['C=O', 'C=O', 'CO', '[H]', '[H]C(O)C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C(O)C1([H])OC(=O)C([H])=C([H])C1([H])[H]', '[H]C([H])([H])C=O', '[H]C([H])[H]', '[H]C([H])[H]']
8			['C=O', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])(OC(=O)C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C([H])C([H])=C([H])C=O', '[H]C([H])[H]', '[H]CO']

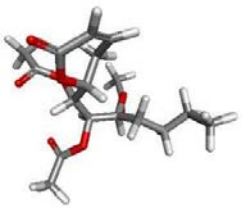
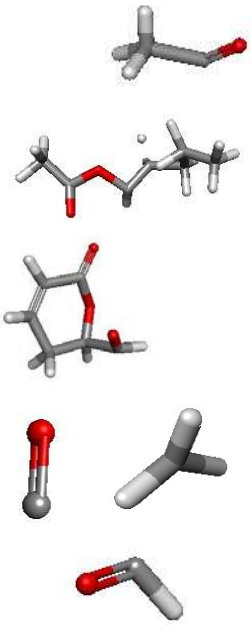
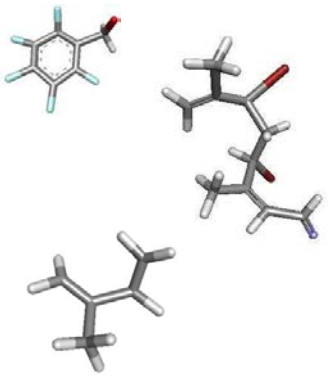
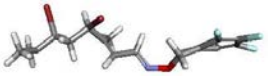
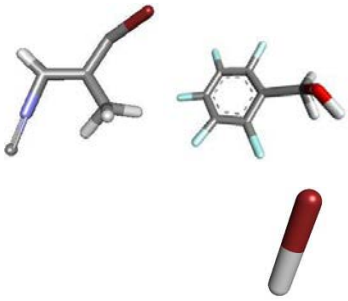
9			['C', 'C', 'O', '[H]', '[H]', '[H]', '[H]C(O)C([H])(OC(=O)C([H])([H])[H])C([H])([H])C([H])([H])C([H])([H])C([H])([H])[H]', '[H]C(O)C1([H])OC(=O)C([H])=C([H])C1([H])[H]', '[H]C([H])([H])C=O']

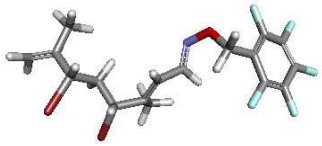
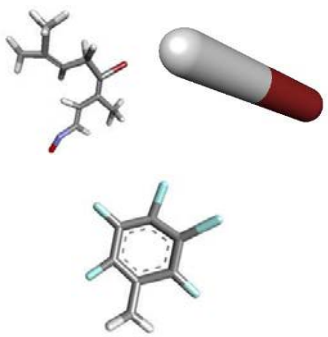
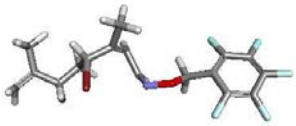
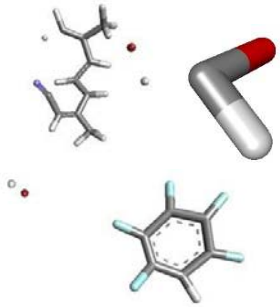
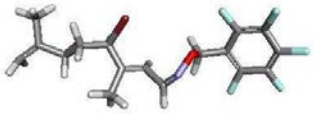
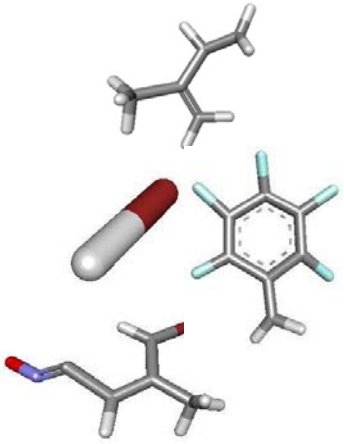
Table 2.2: Different conformations of boronolide generated from rdkit together with fragments that were formed during AIMD and RDKit SMILES fragments.

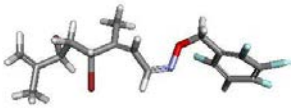
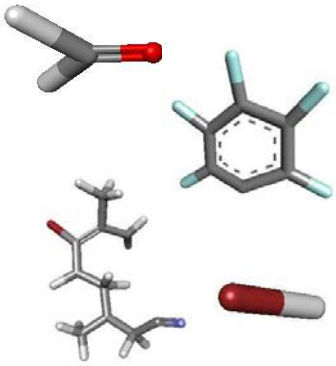
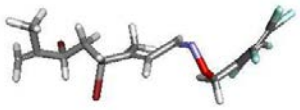
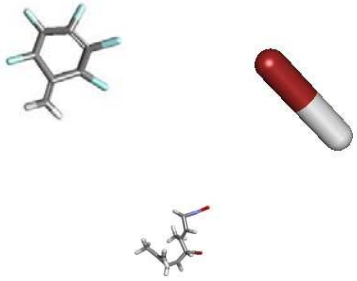
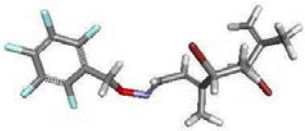
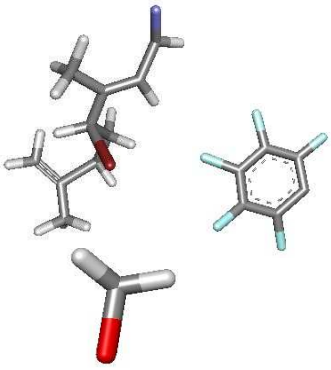
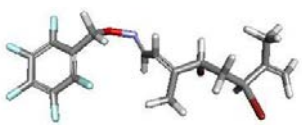
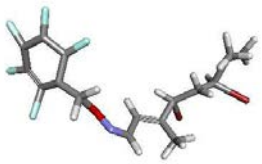
It is interesting to note that carbon monoxide as a stable small neutral molecule is often formed during fragmentation of Boronolide. There is only one trajectory where this is not formed.

2.3.2.3 Results for PFB-oxime

Of interest in addition was how different halogens would behave during AIMD, particularly given the difference in strength of C-F and C-Br bonds. Table 2.3 details the fragmentation of 10 BOMD trajectories of the PFB-oxime.

Trajectory	Initial Conformation	Fragments	SMILES Fragments
0			RDKit did not produce fragments
1			['Br', '[H]', '[H]', '[H]C(C=N)=C(CBr)C([H])([H])[H]', '[H]C([H])(O)c1c(F)c(F)c(F)c(F)c1F', '[H]C([H])C([H])C(=C([H])[H])C([H])([H])[H]']

2			['Br', '[H]', '[H]C(C(=C([H])[H])C([H])([H])[H])C([H])([H])C(Br)C(=C([H])C([H])=NOC([H])([H])c1c(F)c(F)c(F)c1F)C([H])([H])[H]']
3			['Br', 'Br', 'C', 'Fc1cc(F)c(F)c(F)c1F', 'O', '[H]', '[H]', '[H]', '[H]', '[H]C(C([H])C(=C([H])[H])C([H])([H])[H])C([H])C(=C([H])C=N)C([H])([H])[H]']
4			['Br', '[H]', '[H]', '[H]C(C=NO)=C(CBr)C([H])([H])[H]', '[H]C([H])C([H])C(=C([H])[H])C([H])([H])[H]', '[H]C([H])c1c(F)c(F)c(F)c(F)c1F']

5			['Br', 'C', 'Fc1cc(F)c(F)c(F)c1F', 'O', '[H]', '[H]', '[H]', '[H]C(C(=C([H])C([H])=N)C([H])([H])[H])C([H])([H])C(Br)C(=C([H])[H])C([H])([H])[H]']
6			['Br', '[H]', '[H]C(=NO)C([H])=C(C([H])([H])[H])C([H])(Br)C([H])([H])CC(=C([H])[H])C([H])([H])[H]', '[H]C([H])c1c(F)c(F)c(F)c(F)c1F']
7			RDKit did not produce fragments
8			RDKit did not produce fragments

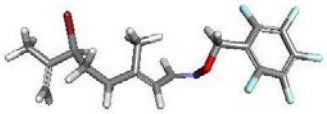
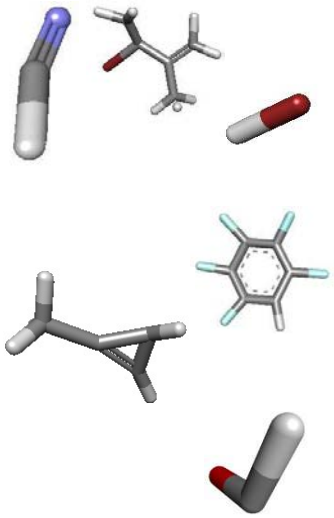
9			['Br', 'CO', 'Fc1cc(F)c(F)c(F)c1F', '[H]', '[H]', '[H]', '[H]C([H])C(Br)C(=C([H])[H])C([H])([H])[H]', '[H]C=N', '[H]CC(=C[H])C([H])([H])[H]']

Table 2.3: The fragments of PFB oxime that were generated during AIMD.

In the fragmentation of the PFB oxime, no C-F bond is cleaved at any stage, but HBr is formed regularly from the two Br atoms present in the original molecule. The species HCBr (with indeterminate charge/splin multiplicity) also appears to form regularly.

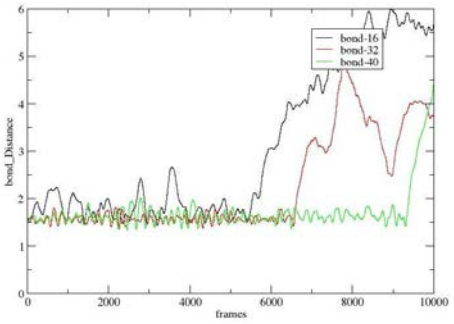
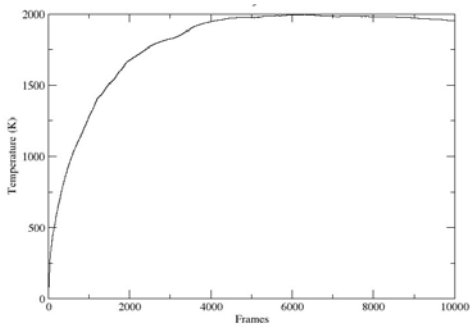
2.3.3 Monitoring of bond breakages during AIMD

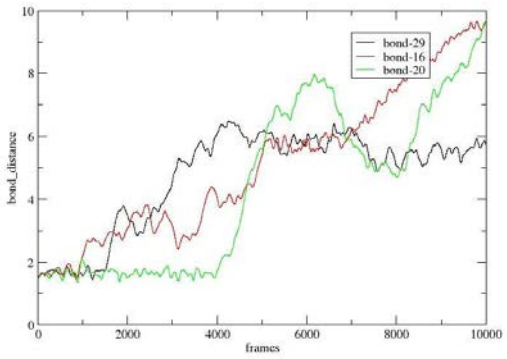
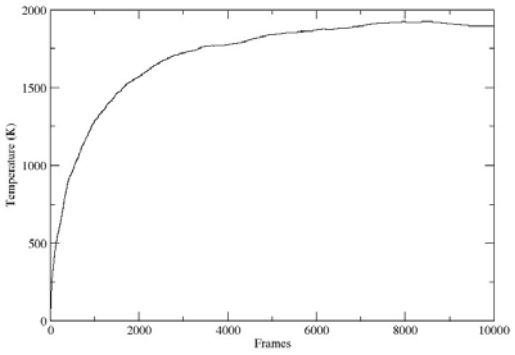
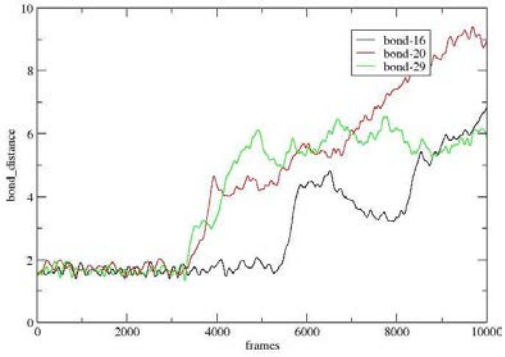
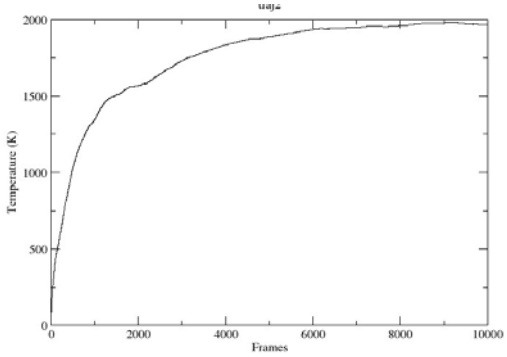
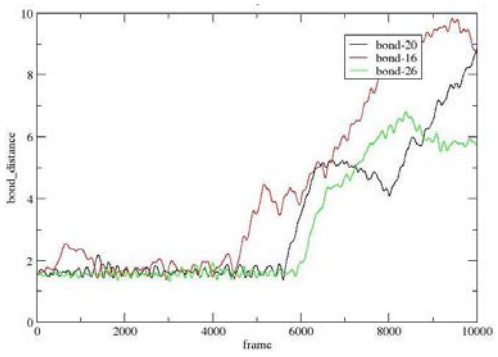
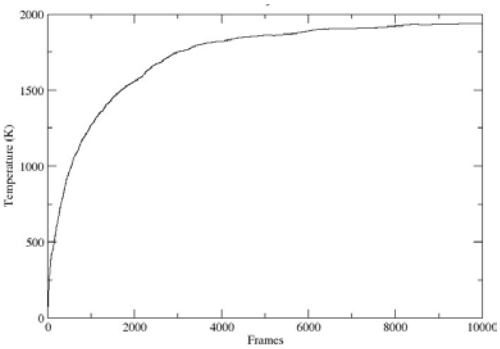
For the ten geometries of each of α-hispanolol, boronolide and PFB oxime, AIMD was performed on all the generated conformations, and during the simulation of bond breaking and formation that took place. A typical instance where bond forming takes place is observed in the formation of HBr where not only are C-Br and O-H bonds broken but a Br-H bond is formed. We plotted the graphs for all the bonds that were observed to break in each trajectory, to try to get a sense of how to recognize bond breaking events during the course of AIMD. Bond formation is an even more difficult scenario to follow and will require monitoring of the full atom distance matrix of the molecule during AIMD, with knowledge of the different bond lengths for all different pairs of atoms; further care must then be taken to rule out a forced proximity of two atoms to be registered as a bond forming event.

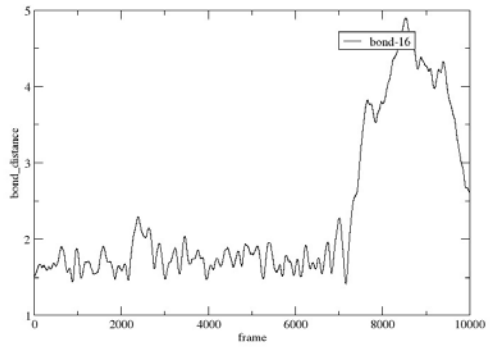
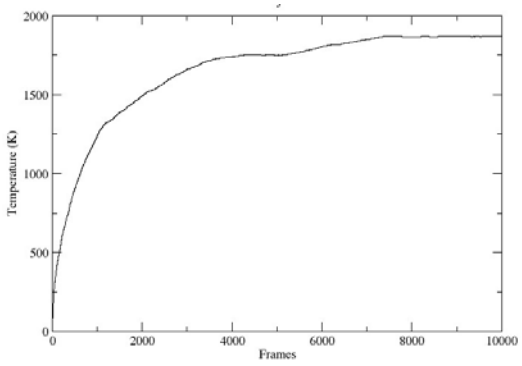
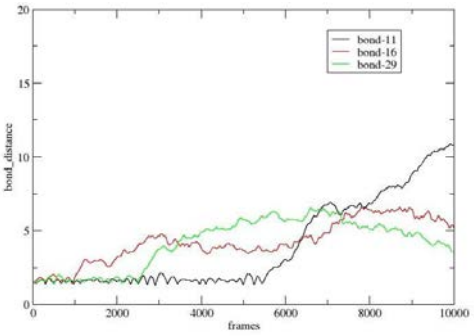
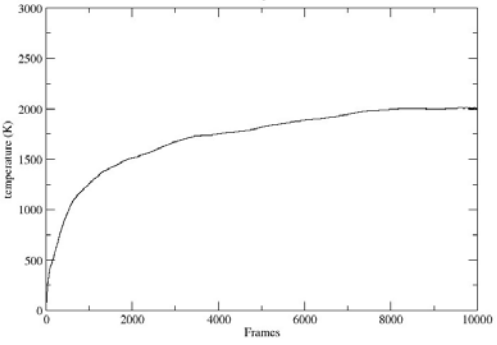
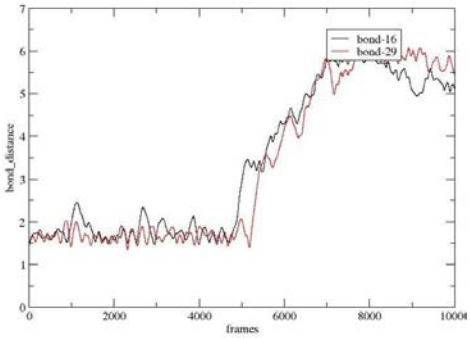
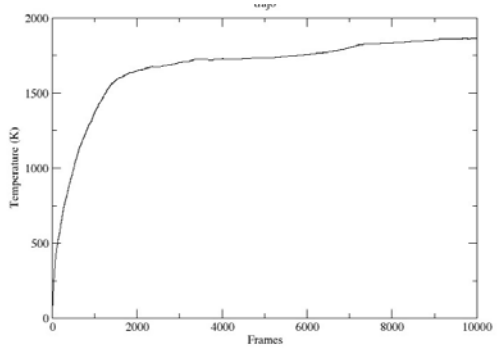
Bond formation monitoring will be explored in future work.

Tables 2.4, 2.5 and 2.6 monitor the bonds that are broken, and in some cases bonds that remain intact. For instance for hispanolol, in trajectory 7, the increase in vibrational energy in bonds that do not break is apparent after bond breakage events. Given the range of bond breakages across all systems none of these bonds then reformed. In the study the assumption was then made that bond breakages are irreversible. In terms of software requirements, it was observed that bond distances at 2000 K rapidly increase after bond breakage, and since a distance of 3 Å is well beyond the length of even an I-I bond, this was chosen as a cut-off for all bond types. So, if by monitoring all bonds identified by RDKit in the original molecule, if any one of them increases in bond distance beyond 3 Å it identifies a fragmentation pathway.

The following graphs in Table 2.4 shows the bond distances for bonds that were broken during the MD production run for all the ten different geometries of hispanolol.

	Selected bond lengths during AIMD	Temperature during AIMD
0		

1	 bond-29 bond-16 bond-20 bond_distance frames	 Temperature (K) Frames
2	 bond-16 bond-20 bond-29 bond_distance frames	 Temperature (K) Frames
3	 bond-20 bond-16 bond-26 bond_distance frame	 Temperature (K) Frames

4	 bond-16 bond_distance frame 0 2000 4000 6000 8000 10000 1 2 3 4 5	 Temperature (K) Frames 0 500 1000 1500 2000 2000 4000 6000 8000 10000
5	 bond-11 bond-16 bond-29 bond_distance frames 0 5 10 15 20 2000 4000 6000 8000 10000	 temperature (K) Frames 0 500 1000 1500 2000 2500 3000 2000 4000 6000 8000 10000
6	 bond-16 bond-29 bond_distance frames 0 1 2 3 4 5 6 7 2000 4000 6000 8000 10000	 Temperature (K) Frames 0 500 1000 1500 2000 2000 4000 6000 8000 10000

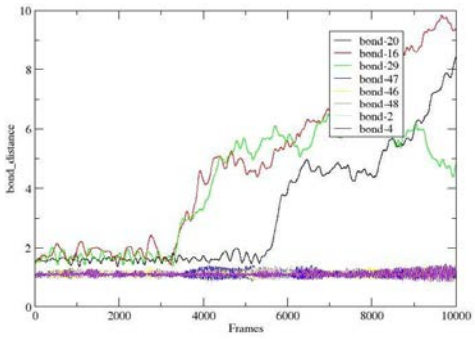
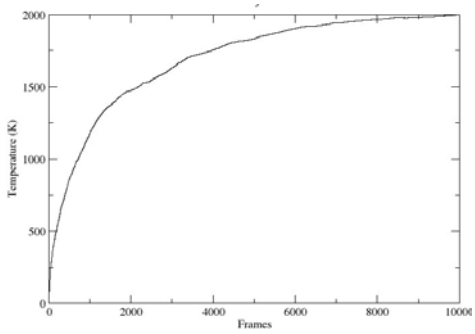
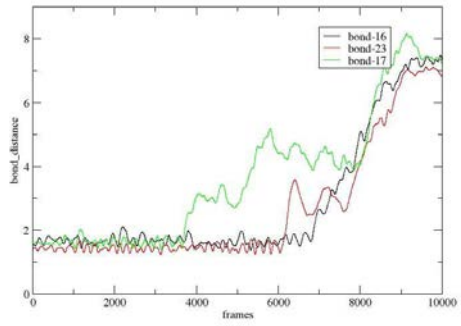
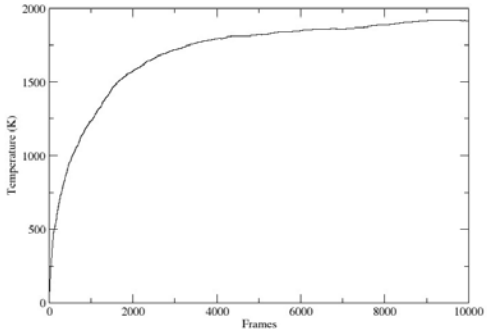
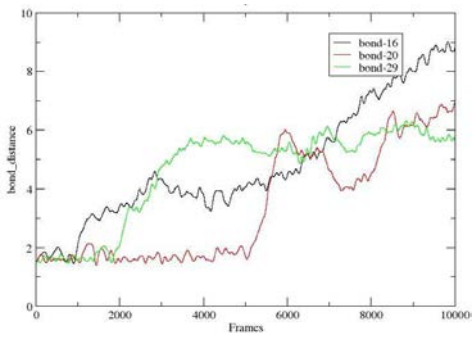
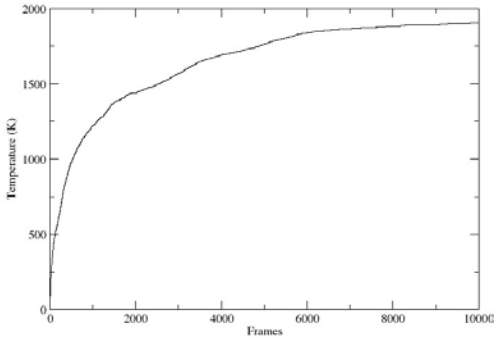
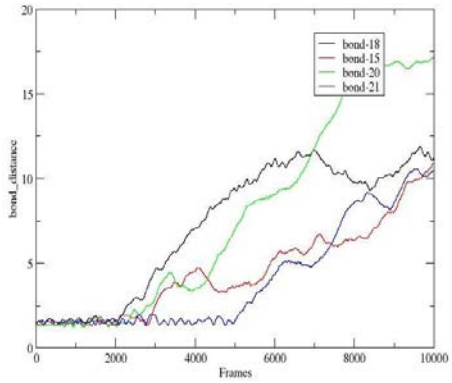
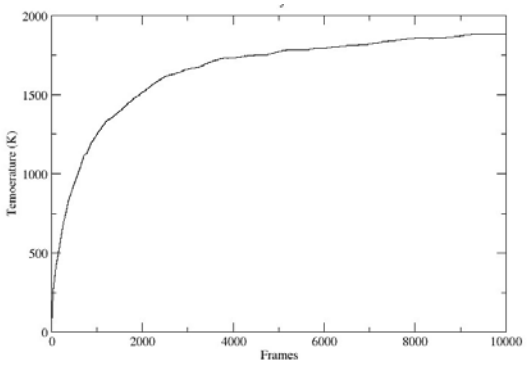
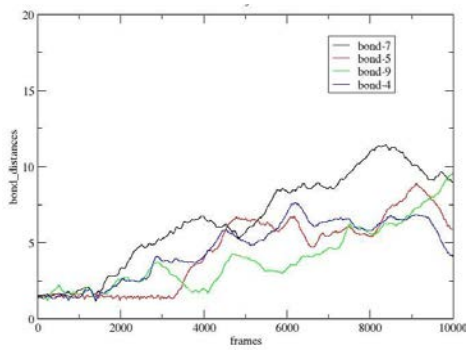
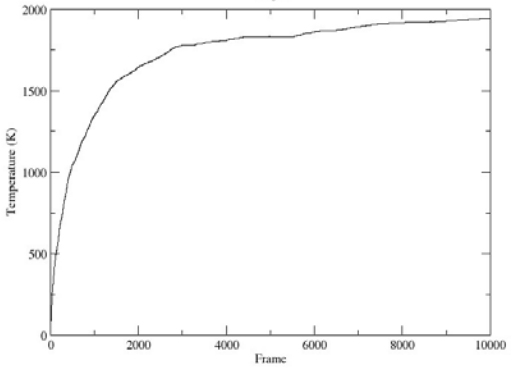
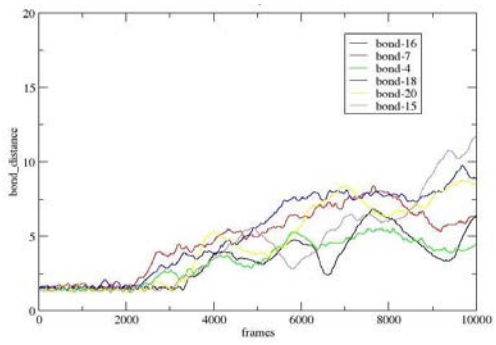
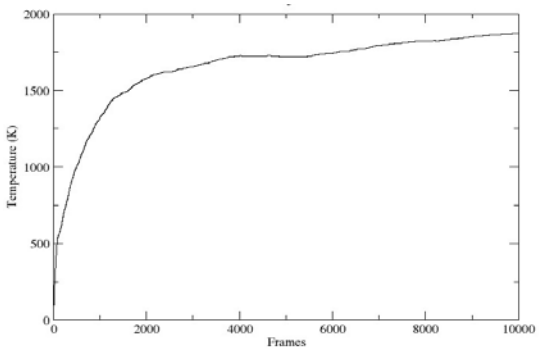
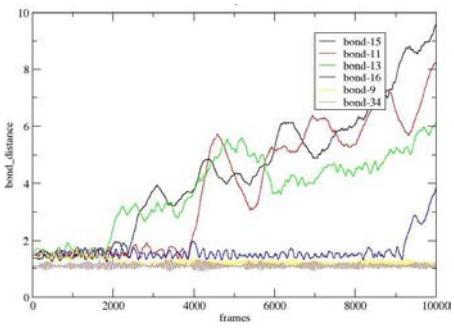
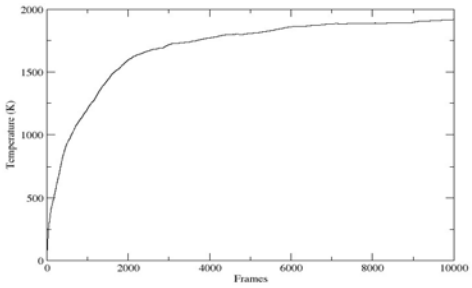
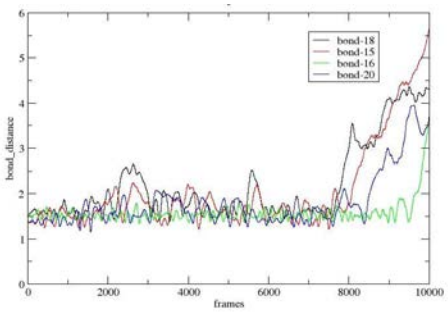
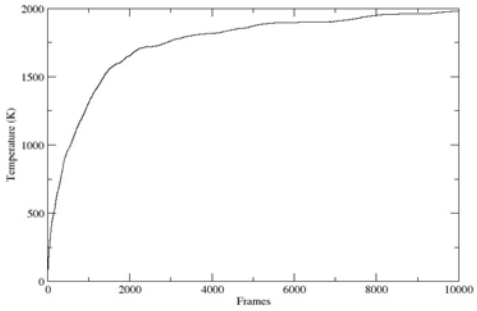
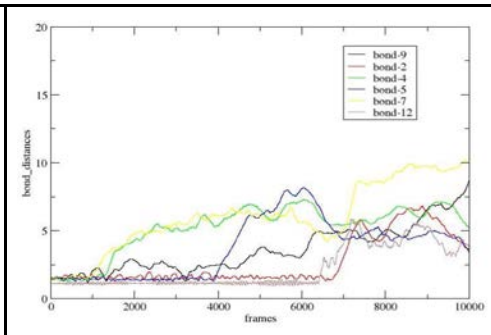
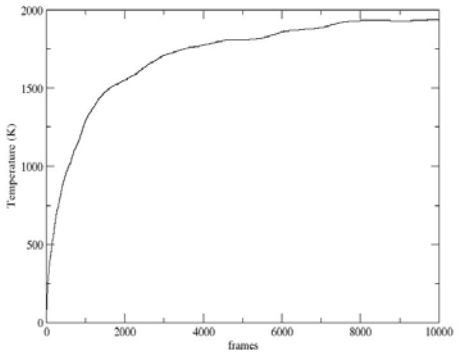
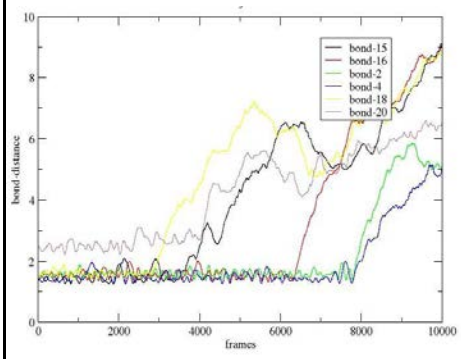
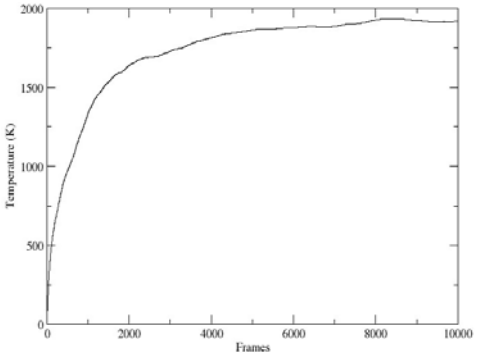
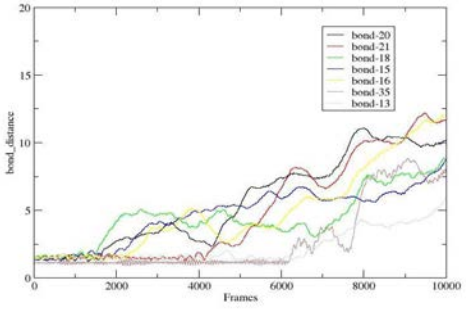
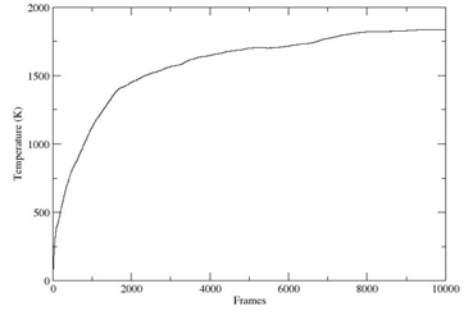
7		
8		
9		

Table 2.4: Temperature during MD of α-hispanolol at 2000K and the bonds broken. The temperature reaches 2000K after the 4000th frame. It is evident that fragmentation is observed after the temperature reaches 2000K.

In a similar manner the following graphs in Table 2.5 shows the bond distances for bonds that were broken during the MD production run for all the ten different geometries of Boronolide.

	Selected bond lengths during AIMD	Temperature during AIMD
0		
1		

2	 bond-16 bond-7 bond-4 bond-18 bond-20 bond-15 bond_distance frames	 Temperature (K) Frames
3	 bond-15 bond-11 bond-13 bond-16 bond-9 bond-34 bond_distance frames	 Temperature (K) Frames
4	 bond-18 bond-15 bond-16 bond-20 bond_distance frames	 Temperature (K) Frames

5	 bond-9 bond-2 bond-4 bond-5 bond-7 bond-12 bond_distances frames	 Temperature (K) frames
6	 bond-15 bond-16 bond-2 bond-4 bond-18 bond-20 bond-distance frames	 Temperature (K) Frames
7	 bond-20 bond-21 bond-18 bond-15 bond-16 bond-35 bond-13 bond_distance Frames	 Temperature (K) Frames

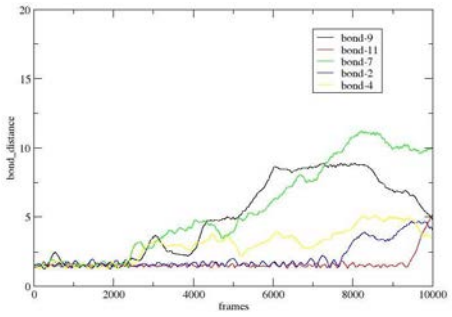
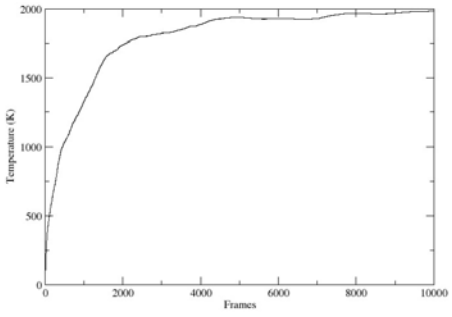
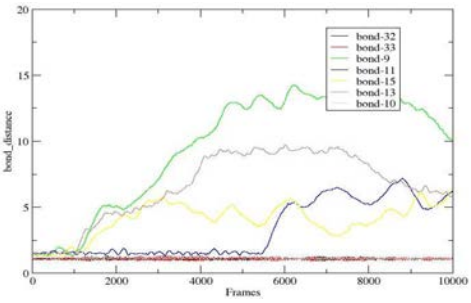
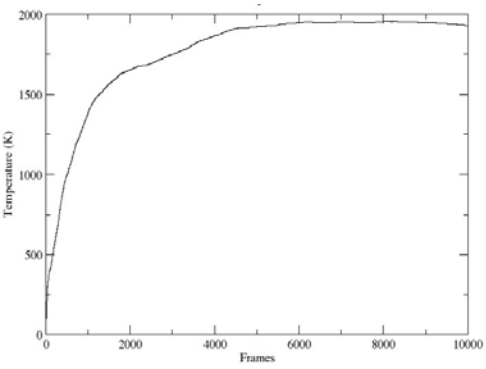
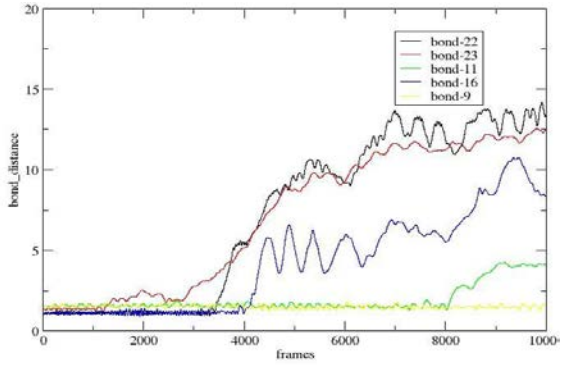
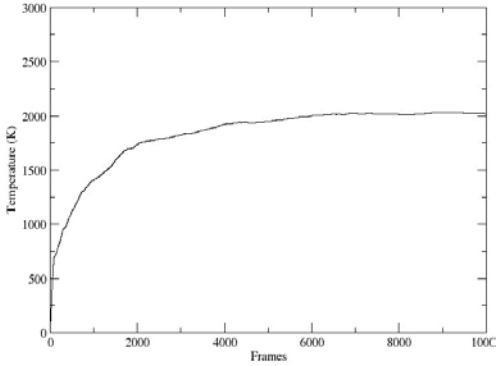
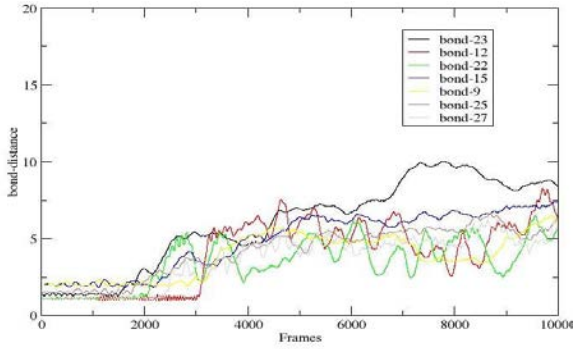
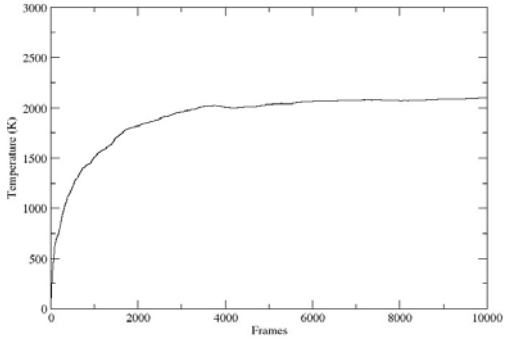
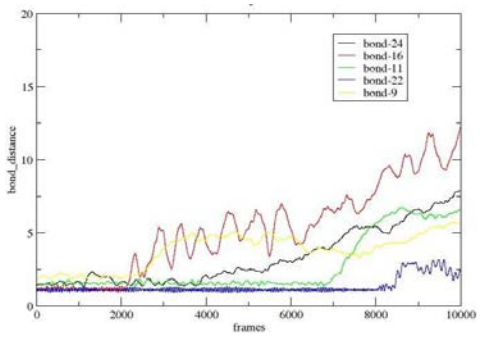
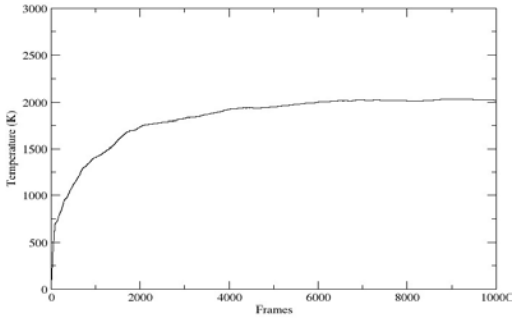
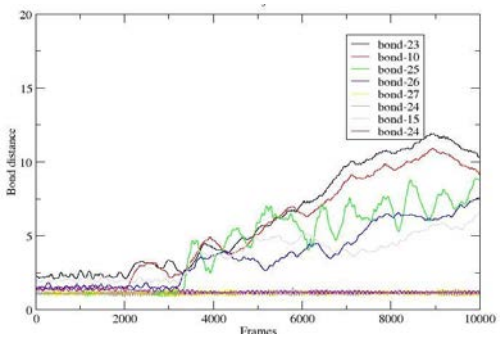
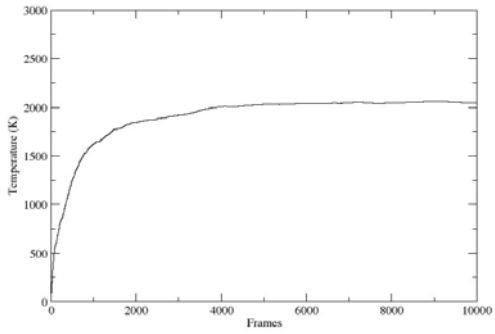
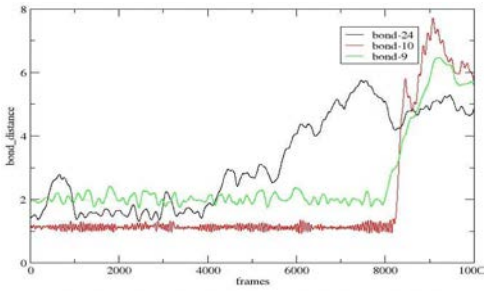
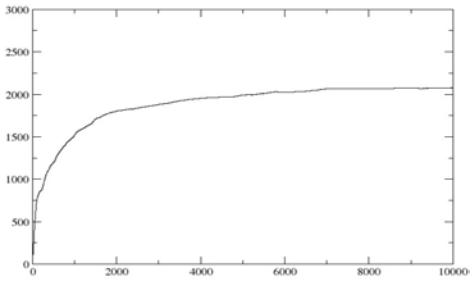
8		
9		

Table 2.5: Monitoring of bond breakages during 10 trajectories of molecular dynamics of boronolide, together with temperature during dynamics..

The graphs in Table 2.6 shows the bond distances for bonds that were broken during the MD production run for some of the ten different geometries of the PFB oxime derivative, in a similar manner.

	Selected bond lengths during AIMD	Temperature during AIMD
1		
3		

4	 bond-24 bond-16 bond-11 bond-22 bond-9 bond_distance frames	 Temperature (K) Frames
5	 bond-23 bond-10 bond-25 bond-26 bond-27 bond-24 bond-15 bond-24 Bond distance Frames	 Temperature (K) Frames
6	 bond-24 bond-10 bond-9 bond_distance frames	

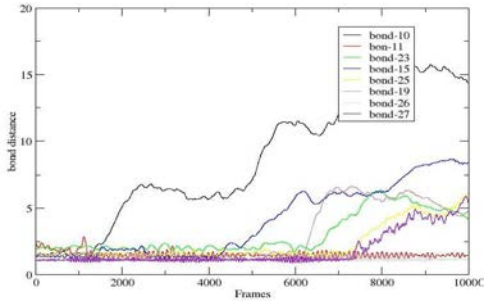
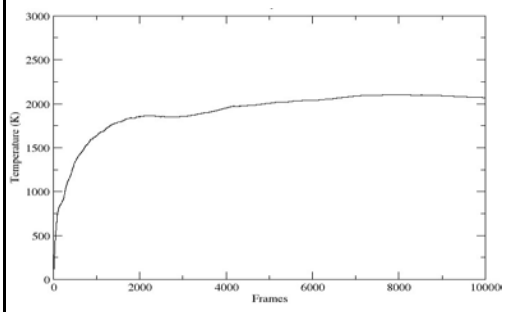
9		

Table 2.6: The distribution of temperature during molecular dynamics of PFB oxime and the distances for each of the bonds broken monitored in each trajectory

Although there are interesting features to these graphs with respect to, for instance, vibrational amplitude in a bond before cleavage, the focus was on defining simply whether a bond had broken or not.

As such we developed within the analysis python script routines to check the bonds that were broken. To this end we generated a list of bonds from the original molecule prior to reading in each trajectory using RDKit (Figure 2.4). From this we created a list of atom pairs which are the two atoms which define the bond. For each of the bonds we created a counter (matching bond_list and bond_broken) for the number of times the bond is broken in the trajectory. The first and the end atom from the

RDKit bond (having the start and end atom indexes) were combined to a pair and were appended to the bond_list. For each bond in bond_broken we count the number of frames for which the bond is greater than 3 Å, although a single frame where this occurs is enough for us to decide that the bond has broken.

```
def
        atom_pairs(sanccompou
        nd): #get list of
        rdkit bonds
        rdk_bonds=sanccompoun
        d.GetBonds()
        #this is our list of atom pairs
        as bond_list bond_list=[]
        bond_broken=[]
        #this is our counter (matching bond_list) for the number of times the bond is
        broken in the trajectory for rdk_bond in rdk_bonds:


                start_atom_idx=rdk_bond.GetBeginAtomIdx()#extract first atom index from this bond
                start_atom=sanccompound.GetAtomWithIdx(start_atom_idx)
                start_atom_symbol=start_atom.GetSymbol()


                end_atom_idx = rdk_bond.GetEndAtomIdx()#extract last atom index
                from this bond end_atom =sanccompound.GetAtomWithIdx(end_atom_idx)
                end_atom_symbol = end_atom.GetSymbol()


                #now we have a pair of atoms start_atom_idx, end_atom_idx)
                atom_pair=[start_atom_idx,end_atom_idx]
                bond_list.append(atom_pair)#add a pair of start/end atoms to list [7,12]
                bond_broken.append(0)
                #for each bond we have a table counting breakages. initialize to 0 here


        return bond_list,bond_broken
```

Figure 2.4: Setting up of bond lists for exploration of bond breaking.

A while loop was set up to read the trajectory files produced frame by frame. Initially the number of atoms line was read, the comment line (time, energy, and i, according to the xyz format of the trajectory) and finally the block of x,y,z coordinates of the atoms for the frame. For each particular frame a loop was set up on the bond list (containing start and end atom indexes) to check if there were any breakages on each particular bond in the molecule. The Cartesian distance between the start and end atoms was calculated, based on the respective x,y and z coordinates. Figure 2.5 shows a code fragment implementing this procedure.

```
BOND_BREAK=3.0
 for bond in
bond_list:

      #at this point must extract x1 y1 z1 from string
      atom_coordlist[bond[0]] atom_coords = atom_coordlist[bond[0]]

      x1 =
      atom_coords[12:2
      4] y1 =
      atom_coords[31:4
      4] z1 =
      atom_coords[50:6
      4]

      #at this point must extract x2 y2 z2 from string
      atom_coordlist[bond[1]] atom_coords2 =atom_coordlist[bond[1]]

      x2 =
      atom_coords2[12:2
      4] y2 =
      atom_coords2[31:4
      4] z2 =
      atom_coords2[50:6
      4]

      #need to calculate
      distance using dx =
      float(x1)-float(x2) dy
      = float(y1)-float(y2)
      dz = float(z1)-
      float(z2)
      #print(x2,y2,z2)
      distance = math.sqrt((dx)**2 + (dy)**2 + (dz)**2 )

      if (distance>BOND_BREAK):
            bond_broken[counter]=bond_broken[counter]+1
      counter=counter+1
      #still to keep track of the bond number
```

Figure 2.5: Portion of code monitoring bond breakage

A list of bonds broken (bond_broken_list) was generated from inspection of the bond_broken set of counters. RDKit was then used to produce a SMILES string by fragmenting the original molecule with the bonds in the list (Figure 2.6).

```
if bond_broken_list:
broken_molecule=

AllChem.FragmentOnBonds(sanccompound,bond_broken_list,False)
molecules_for_docking=

AllChem.FragmentOnBonds(sanccompound,bond_broken_list,True)
else:
    broken_molecule=sanccompound
    molecules_for_docking=sanccompound

fragment_smiles=AllChem.MolToSmiles(broken_molecule)
docking_smiles=AllChem.MolToSmiles(molecules_for_docking)
docking_smiles=re.sub("\*","H",docking_smiles)
```

Figure 2.6: Code using RDKit to identify fragments

In Figure 2.6, code is presented for creating whole sanitized molecules from the fragments (capping with H's). The fragments that were generated from this method, using RDKit to FragmentOnBonds and generate SMILES, as shown in Tables 2.1, 2.2 and 2.3 are different in comparison to the visual representation of fragmentation (as viewed using Discovery Studio). This could be because Discovery Studio used different parameters than what we used to view the fragments in RDKit (to consider a bond to be broken the bond distance should be greater than 3Å). In RDkit for Table 2.3 for the PFB-Oxime, Trajectory 4 shows that there are no fragments produced during this simulation whereas when we viewed the last frame of the trajectory in Discovery Studio, three fragments were formed, though for trajectory 6 when we viewed the coordinates from the last frame in discovery studio there were no fragments formed even though in rdkit there were fragments produced. Again, the RDKit was created in the context of Drug Discovery, and sanitizing unusual fragments of molecules may well be beyond its intended capability.

An improved way, therefore to use our determined bond breakages and determine fragments was to use network maps to identify fragments that were generated during AIMD. In addition to this an advantage would be to be able to identify exactly which atoms composed a fragment for all different fragments that were produced. The way this was done was using the networkx python library. Figure 2.7 shows a fragment of code

that was used to create a graph view of the molecule, and Figure 2.8 shows the effect of using this code on a single trajectory that formed two fragments.

```
import networkx as nx
 def
create_map(bond_list_l,bonds_broken_l,file
name):

      molecule_map=n
      x.Graph()
      bond_number=0
      for bond in
      bond_list_l:
            if not (bond_number in bonds_broken_l):
                  print("adding bond ",bond," bond no ",bond_number," to map")
                  molecule_map.add_edge(bond[0],bond[1])
            else:
                  print("not adding bond ",bond," bond number
                  ",bond_number)
            bond_number=bond_number+1
      for broken in bonds_broken_l:
            print("broken                                                    ",broken)

      components=list(nx.connected_components(molecule_map))
```

Figure 2.7: Using the networkx library to identify connected fragments

Figure 2.8: An example where the networkx library has separated the graph describing the molecule into two fragments, with each atom specified by number.

2.3.4 Charges and Spins

In order to accommodate charges on fragments, the simplest approach was to include a directive for the printing of Mulliken charges within CP2K. Figure 2.9 shows a portion of the population analysis for a single frame.

```
MULLIKEN POPULATION ANALYSIS

# Atom Element Kind Atomic population (alpha,beta) Net charge Spin
moment
      1     1        3.041842  3.038536  -0.080378  0.003305
       22  1         3.273165  3.163761  -0.436926  0.109403
      25  1  3.266133  3.154618  -0.420751  0.111516 2
      2  1.987955  1.851171   0.160874  0.136783
...
```

Figure 2.8: Mulliken population analysis

The log files that were produced during MD were analyzed to get the charges of each fragment that was generated. We assigned each of the atomic charges to the fragments identified; this enabled us in subsequent steps to identify the fragment from each trajectory with the highest positive charge (which are then used to predict the mass spectra of different compounds). Figure 2.9 illustrates the code used to calculate the total charge on each fragment identified through the networkx utilities.

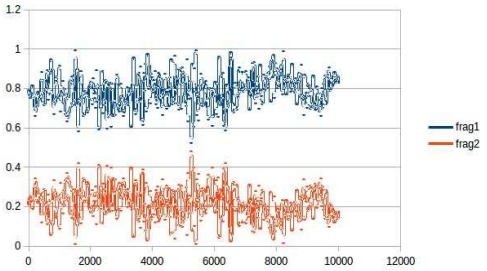
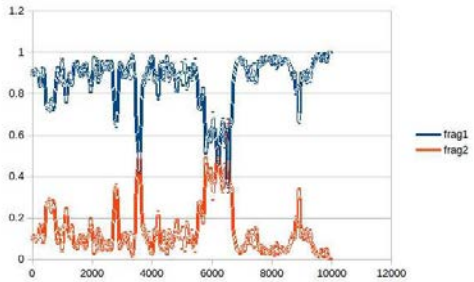
```
for fragment in components:
      #print("component number ",fragmentnumber,fragment)
      #print("spins ",spins)
      #print("charges
      ",charges)
      totalcharge=0.0
      totalspin=0.0
      for atom in
      fragment:
            totalcharge=totalcharge+float(charges[atom])
            totalspin=totalspin+float(spins[atom])
            #print("charge is ",totalcharge)
            #print("spin is ", totalspin)
            allcharges.append(totalcharge)
            allspins.append(totalspin)
      fragmentnumber=fragmentnumber+1
```

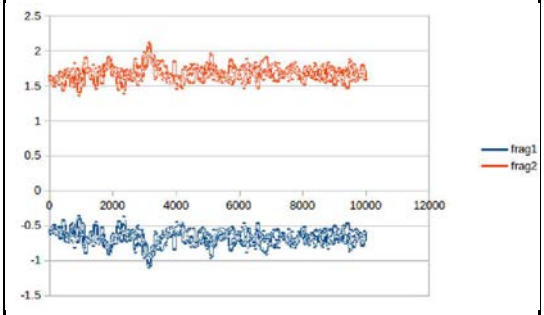
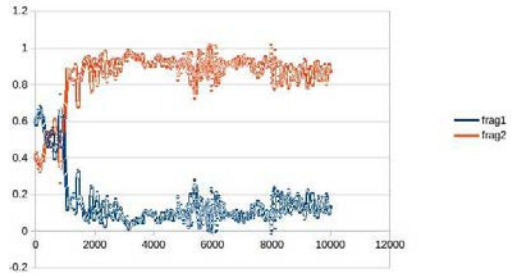
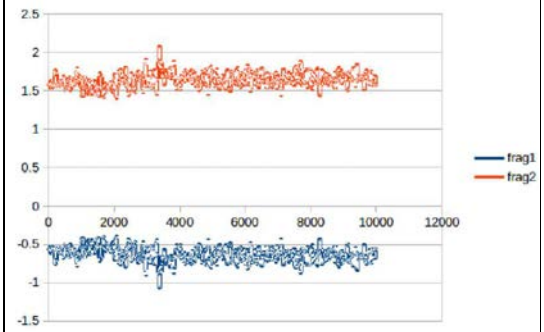
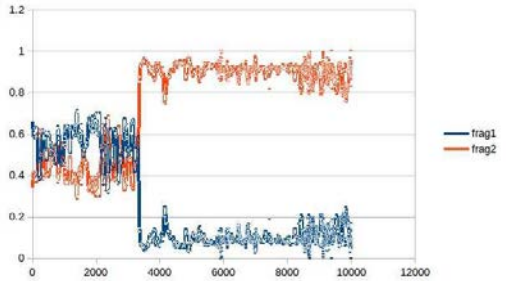
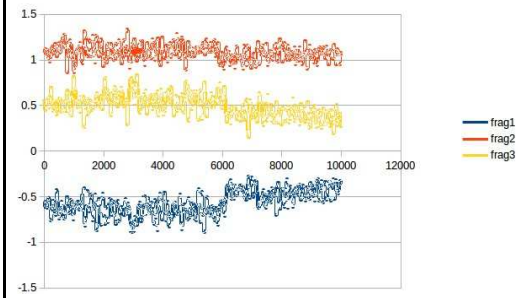
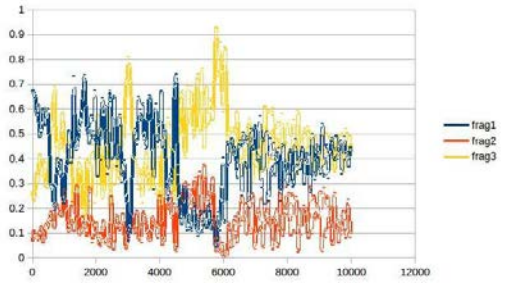
Figure 2.9: Calculation of total charge by fragment

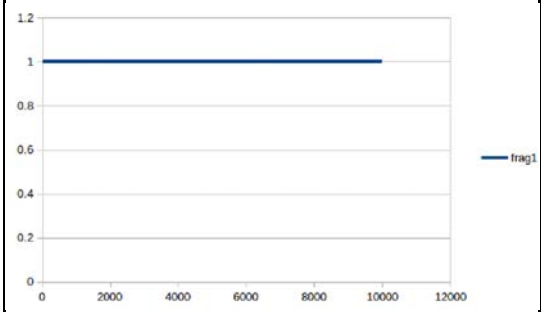
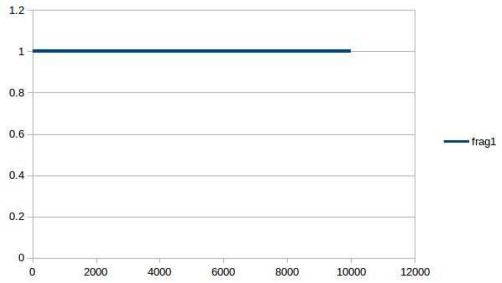
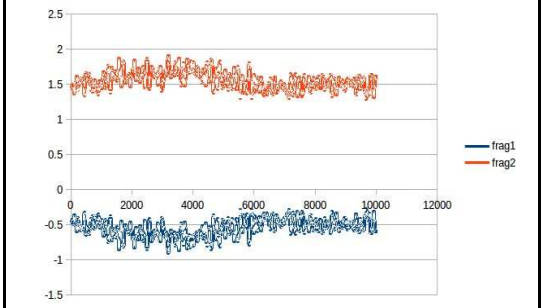
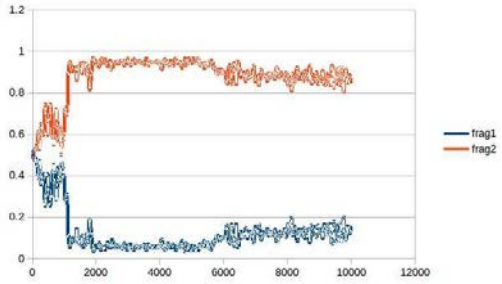
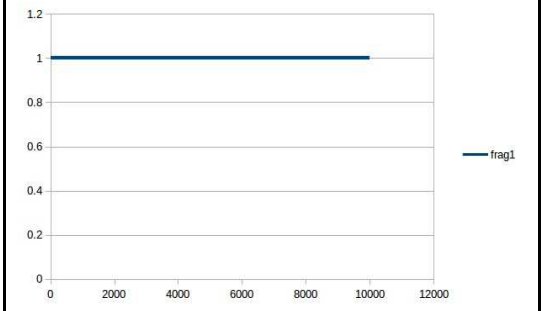
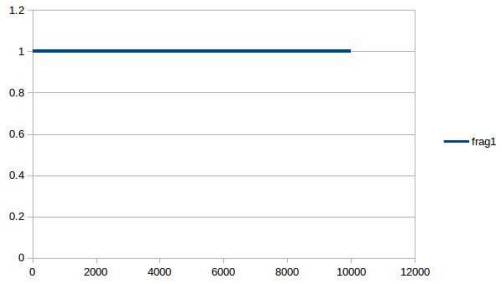
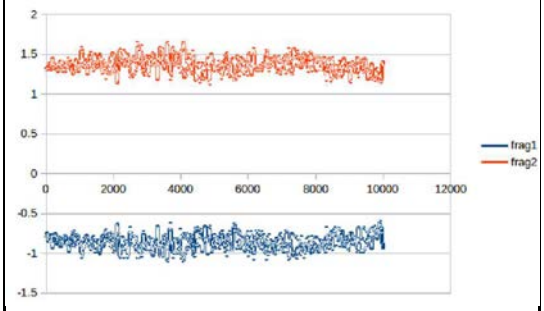
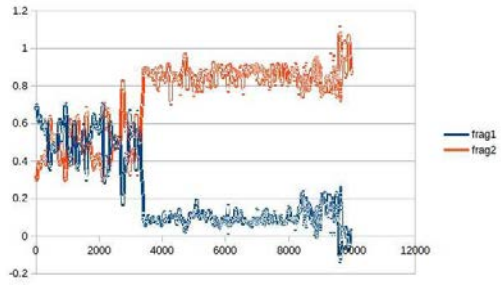
Although the total spin was calculated for each fragment, this was not used further in mass spectral generation. The distribution of charges per fragment during the limited number (10) trajectories was followed, to assess the feasibility of simply taking the fragment with the greatest positive charge through to mass spectra peak.

2.3.4.1 Evolution of Charge and Spin for α-hispanolol

The charges and spin on each final fragment were plotted during the course of the trajectories producing those fragments for α-hispanolol (Table 2.7). It was interesting to note that the chemical processes leading to fragmentation were dominated by homolytic cleavage - that is, the charges on separate parts of the molecule did not change as the parts fragmented, but the spin densities had a marked change at the events. So sudden changes in spin density on fragments was indicative of fragmentation. Where no fragments were observed to form, the total charge remained constant (1.0) as did the total spin.

	Charges	Spin
0		

1	 2.5 2 1.5 1 0.5 0 -0.5 -1 -1.5 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1.2 1 0.8 0.6 0.4 0.2 0 -0.2 0 2000 4000 6000 8000 10000 12000 frag1 frag2
2	 2.5 2 1.5 1 0.5 0 -0.5 -1 -1.5 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1 frag2
3	 1.5 1 0.5 0 -0.5 -1 -1.5 0 2000 4000 6000 8000 10000 12000 frag1 frag2 frag3	 1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 2000 4000 6000 8000 10000 12000 frag1 frag2 frag3

4	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1
5	 2.5 2 1.5 1 0.5 0 -0.5 -1 -1.5 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1 frag2
6	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1
7	 2 1.5 1 0.5 0 -0.5 -1 -1.5 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1.2 1 0.8 0.6 0.4 0.2 0 -0.2 0 2000 4000 6000 8000 10000 12000 frag1 frag2

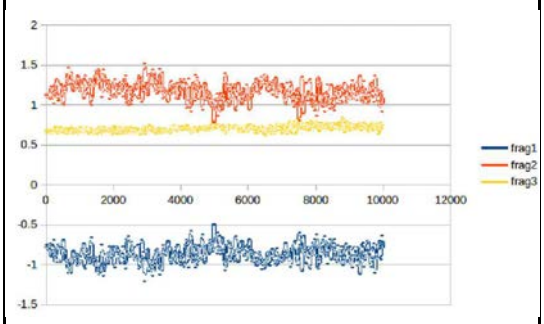
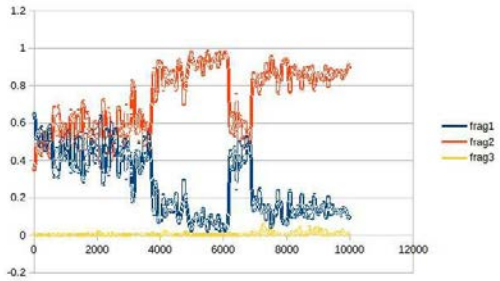
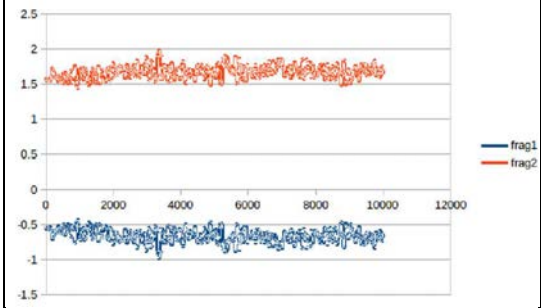
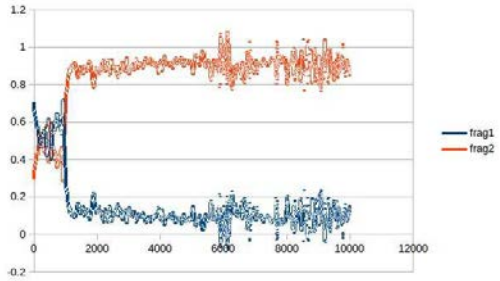
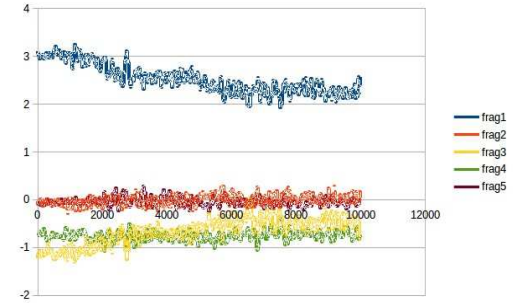
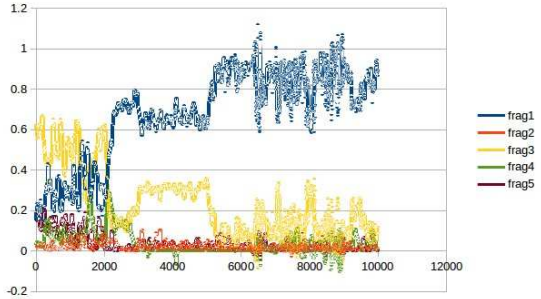
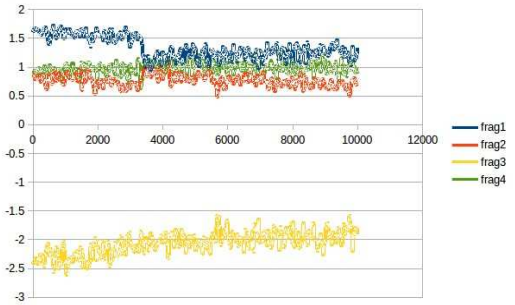
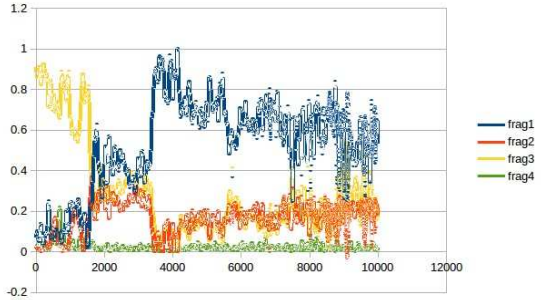
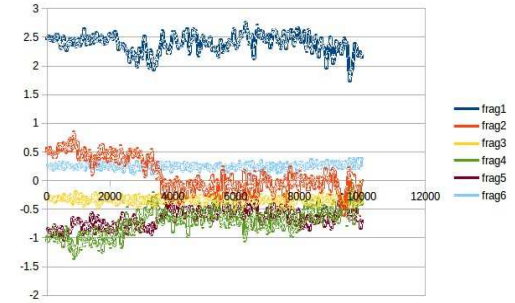
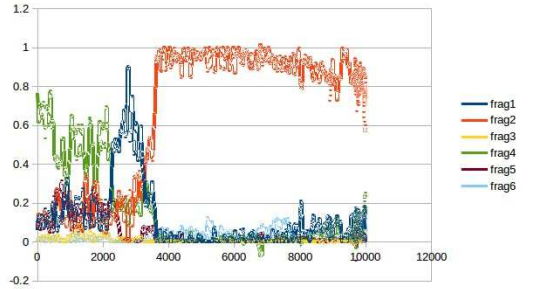
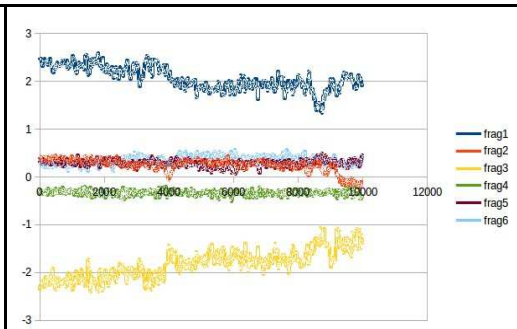
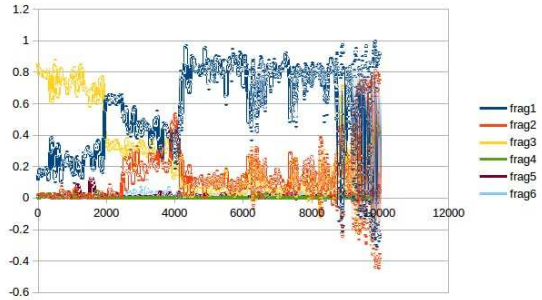
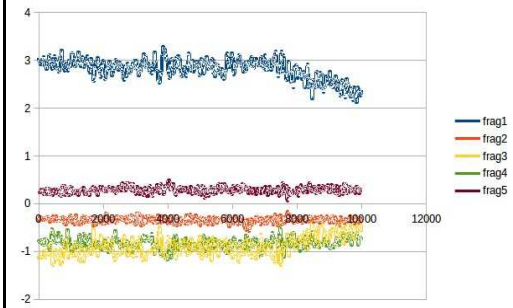
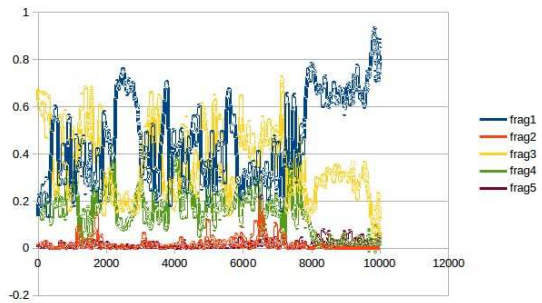
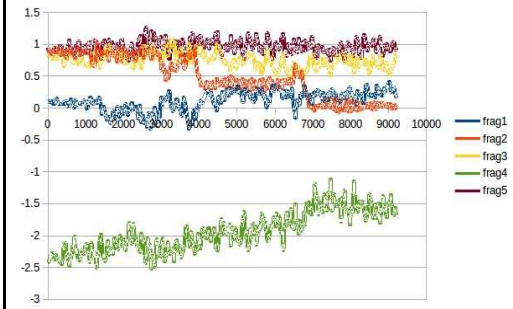
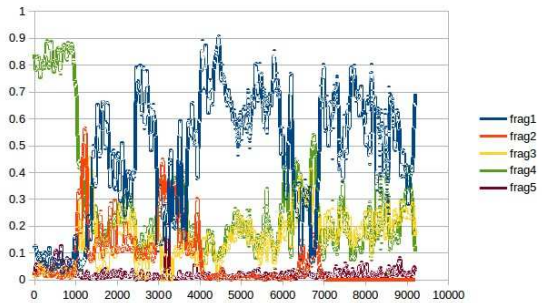
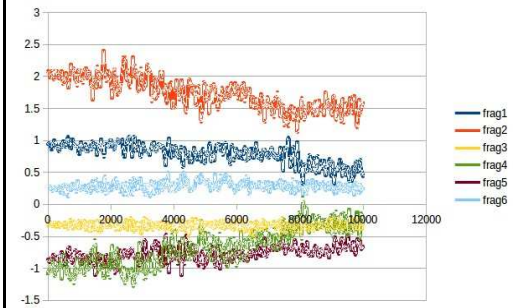
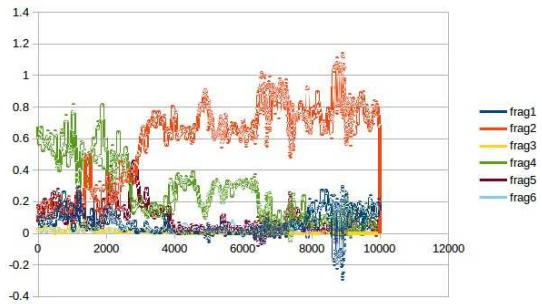
8		
9		

Table 2.7: the charges and spins of the alpha hispanolol fragments produced in each trajectory

2.3.4.1 Evolution of Charge and Spin for boronolide

For boronolide, much the same in terms of observation were made for the evolution of charge and spin throughout the course of trajectories (Table 2.8). However we observed cases where the fragment with the positive charge after fragmentation was more ambiguous. Trajectories 1 and 5, for example illustrate this. The choice was made in the generation of mass spectra, to only use the fragment with the greatest positive charge at the end of the trajectory.

	Charges	Spin
0		
1		
2		

3	 frag1 frag2 frag3 frag4 frag5 frag6	 frag1 frag2 frag3 frag4 frag5 frag6
4	 frag1 frag2 frag3 frag4 frag5	 frag1 frag2 frag3 frag4 frag5
5	 frag1 frag2 frag3 frag4 frag5	 frag1 frag2 frag3 frag4 frag5
6	 frag1 frag2 frag3 frag4 frag5 frag6	 frag1 frag2 frag3 frag4 frag5 frag6

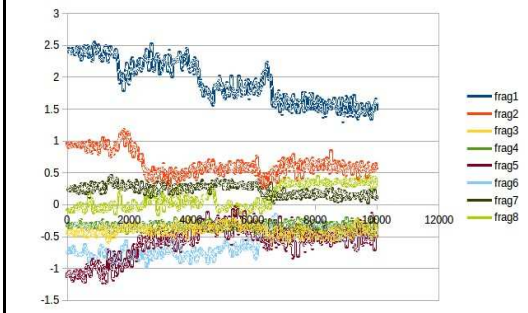
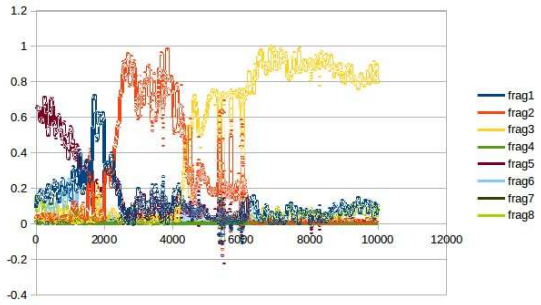
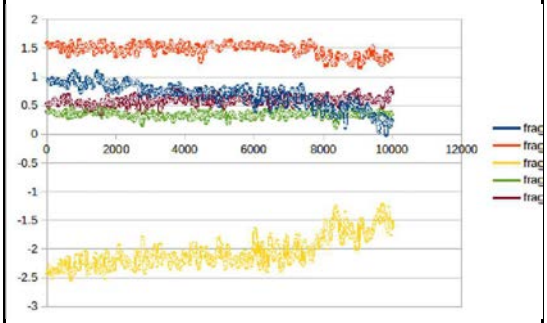
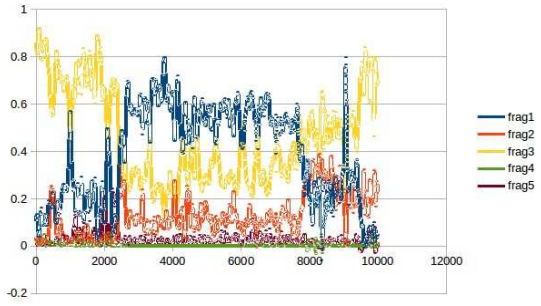
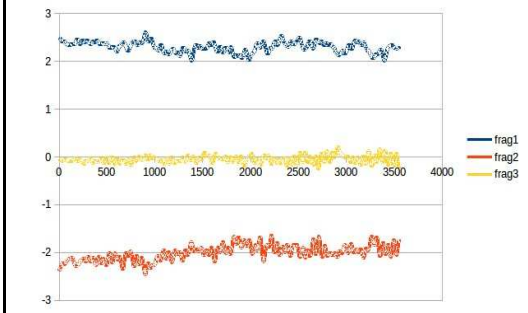
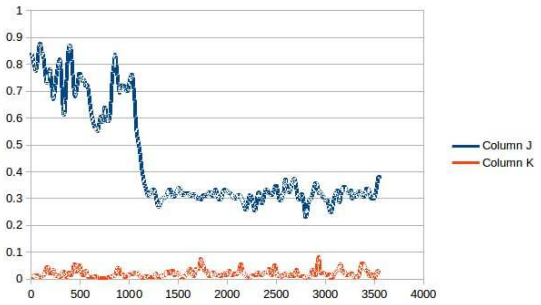
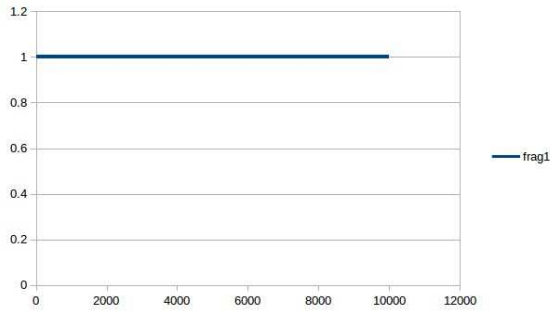
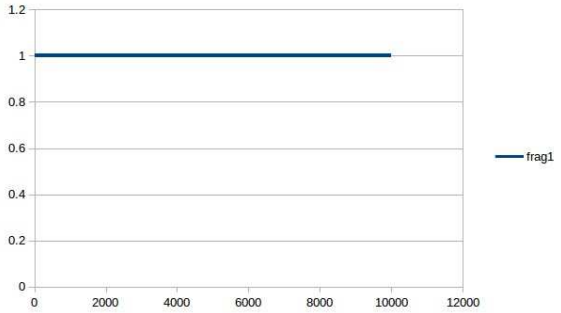
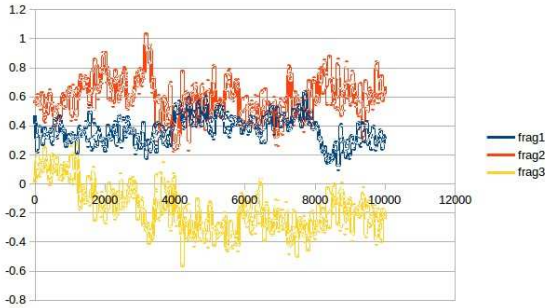
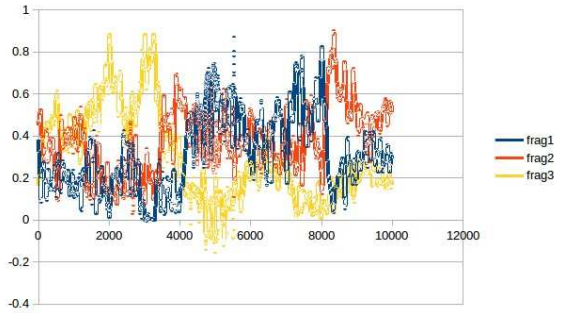
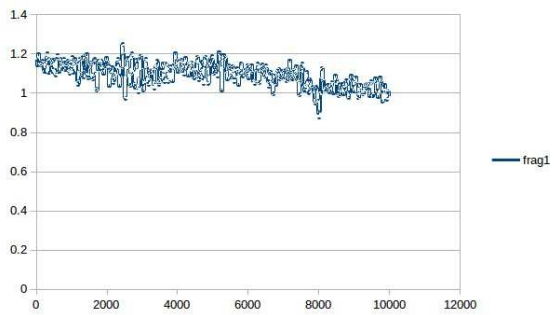
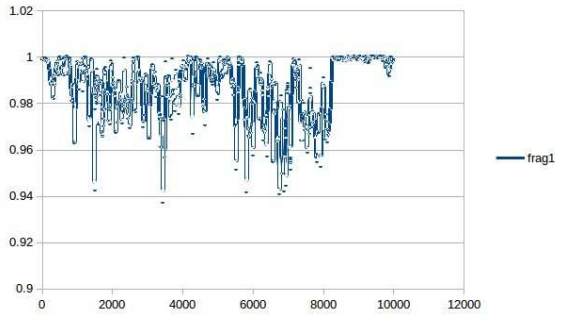
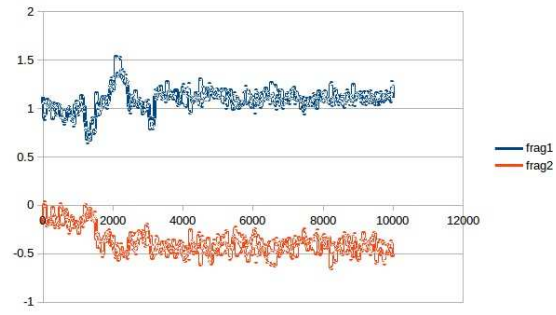
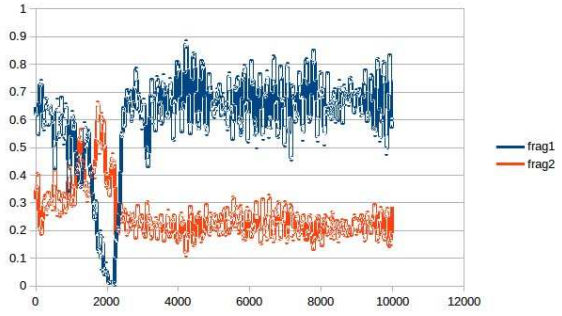
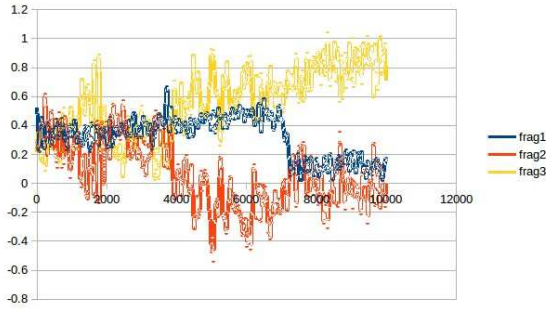
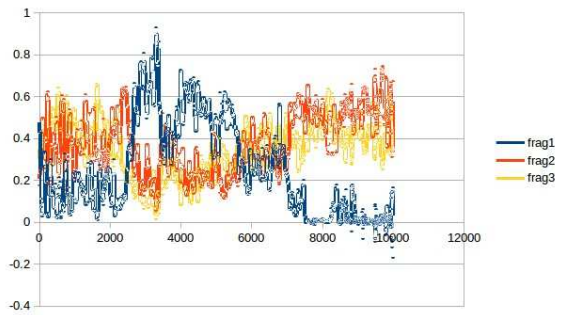
7		
8		
9		

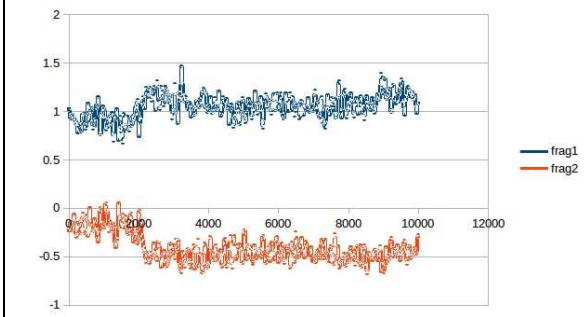
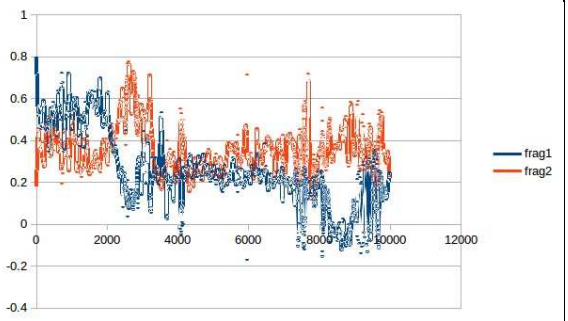
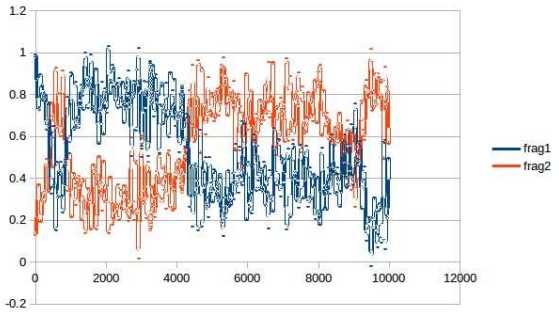
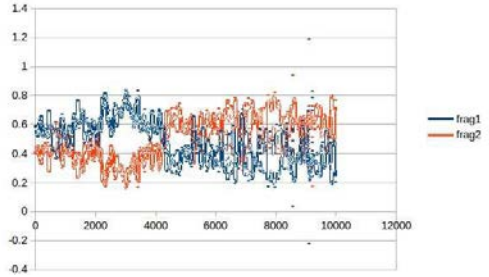
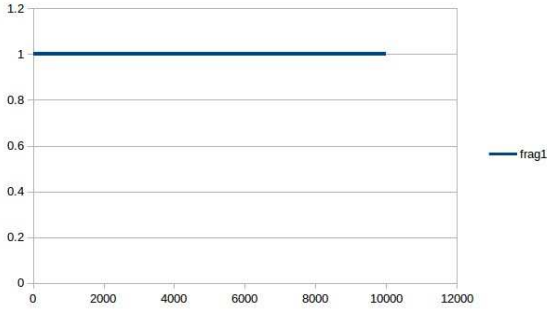
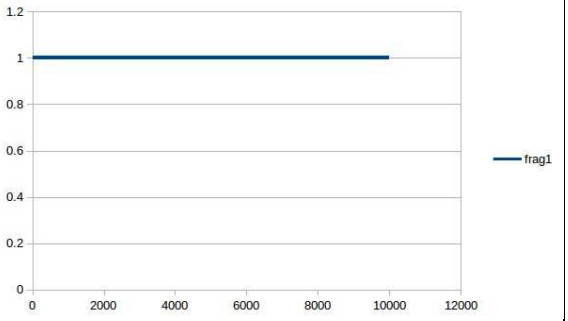
Table 2.8: The charges and spins of boronolide fragments that were generated during the AIMD of boronolide

2.3.4.1 Evolution of Charge and Spin for the PFB-oxime derivative

Table 2.9 illustrates the evolution of charge and spin for the PFB-oxime derivative. Again there were instances where the formation of the fragment with positive charge was ambiguous (Trajectories 4 and 9). An improvement may be (subject to testing) to use other forms of charge calculation, such as Atoms-In-Molecules calculations.

	Charges	Spins
0		
1		

2	 1.4 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1	 1.02 1 0.98 0.96 0.94 0.92 0.9 0 2000 4000 6000 8000 10000 12000 frag1
3	 2 1.5 1 0.5 0 -0.5 -1 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 2000 4000 6000 8000 10000 12000 frag1 frag2
4	 1.2 1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 0 2000 4000 6000 8000 10000 12000 frag1 frag2 frag3	 1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 0 2000 4000 6000 8000 10000 12000 frag1 frag2 frag3

5	 2 1.5 1 0.5 0 -0.5 -1 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 0 2000 4000 6000 8000 10000 12000 frag1 frag2
6	 1.2 1 0.8 0.6 0.4 0.2 0 -0.2 0 2000 4000 6000 8000 10000 12000 frag1 frag2	 1.4 1.2 1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 0 2000 4000 6000 8000 10000 12000 frag1 frag2
7	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1	 1.2 1 0.8 0.6 0.4 0.2 0 0 2000 4000 6000 8000 10000 12000 frag1

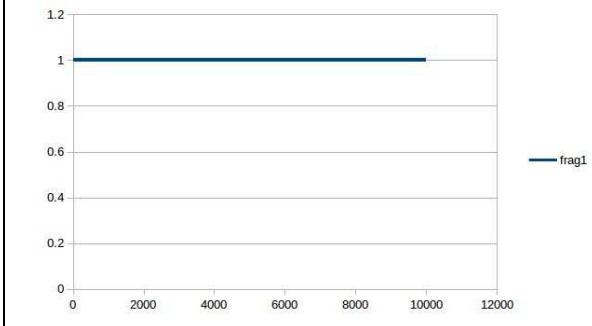
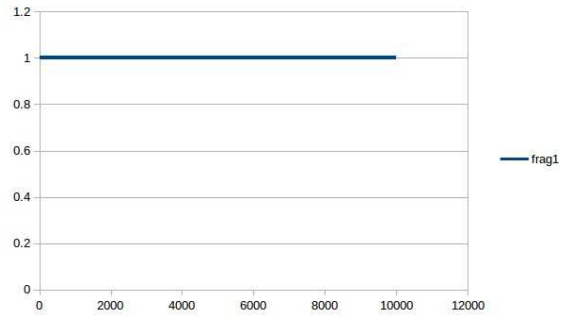
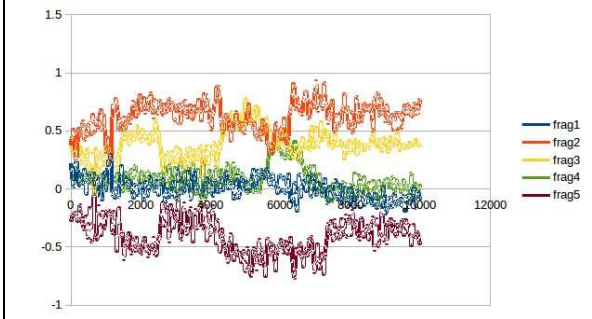
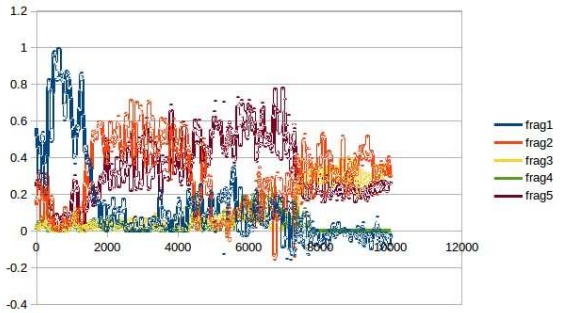
8		
9		
		

Table 2.9: The charges and spins of PFB oxime fragments from AIMD

2.5 Acquisition of multiple spectra using High Performance Computing

The acquisition of many trajectories made use not only of CP2K but gnu parallel to manage the run across 8 nodes at the CHPC. Figure 2.10 shows the submission script used to work with CP2K vs 6.1, compiled with Parallel Studio, using gnu parallel. This script simply finds all input files in a directory, and sends them for execution through gnu parallel using 1 node with 24 cpus for each trajectory.

```
#!/bin/bash
#PBS -e
/mnt/lustre/users/ynovokoza/nest_database/Nitrogen/2500_temp/stderr.o
ut #PBS -o
/mnt/lustre/users/ynovokoza/nest_database/Nitrogen/2500_temp/stdout.o
ut
#PBS -P CBBI1122
#PBS -l select=8:ncpus=24:mpiprocs=24:nodetype=haswell_reg
#PBS -l walltime=48:00:00
#PBS -q normal
#PBS
-m be
#PBS
-r n

module purge module add chpc/gnu/parallel-20180622
module add gcc/5.1.0 module add
chpc/parallel_studio_xe/17.0/2017.4.056 module add
chpc/cp2k/6.1/parallel_studio/2017  JOBSPERNODE=1  cd -P
/mnt/lustre/users/ynovokoza/nest_database/Nitrogen/2500_
temp

find . -name "*cp2k" | parallel -M --sshdelay 0.2 -j ${JOBSPERNODE} -u -
sshloginfile ${PBS_NODEFILE} "module add
chpc/parallel_studio_xe/17.0/2017.4.056;module add
chpc/cp2k/6.1/parallel_studio/2017;cd
/mnt/lustre/users/ynovokoza/nest_database/Nitrogen/2500_temp;cp2k.psmp -i
{} > {.}.log"
```

Figure 2.10: Job submission script for many trajectories on the CHPC

2.6 Theoretical Mass Spectra

The mass spectra of all the molecules were predicted by only plotting the fragments with the highest positive charge in each trajectory, since in experimental mass spectrometry only the positively charged ions are recorded of the mass spectra and the uncharged radicals are lost to the system. Our initial mass spectra prediction was without taking into account the isotopic distribution of elements present in the fragments. Figure 2.11 shows the predicted mass spectrum of α-hispanolol, without the inclusion of isotopic

abundances. This prediction was at 2000 K, using the DFTB+ level of theory for 100 trajectories at 10000 steps per trajectory.

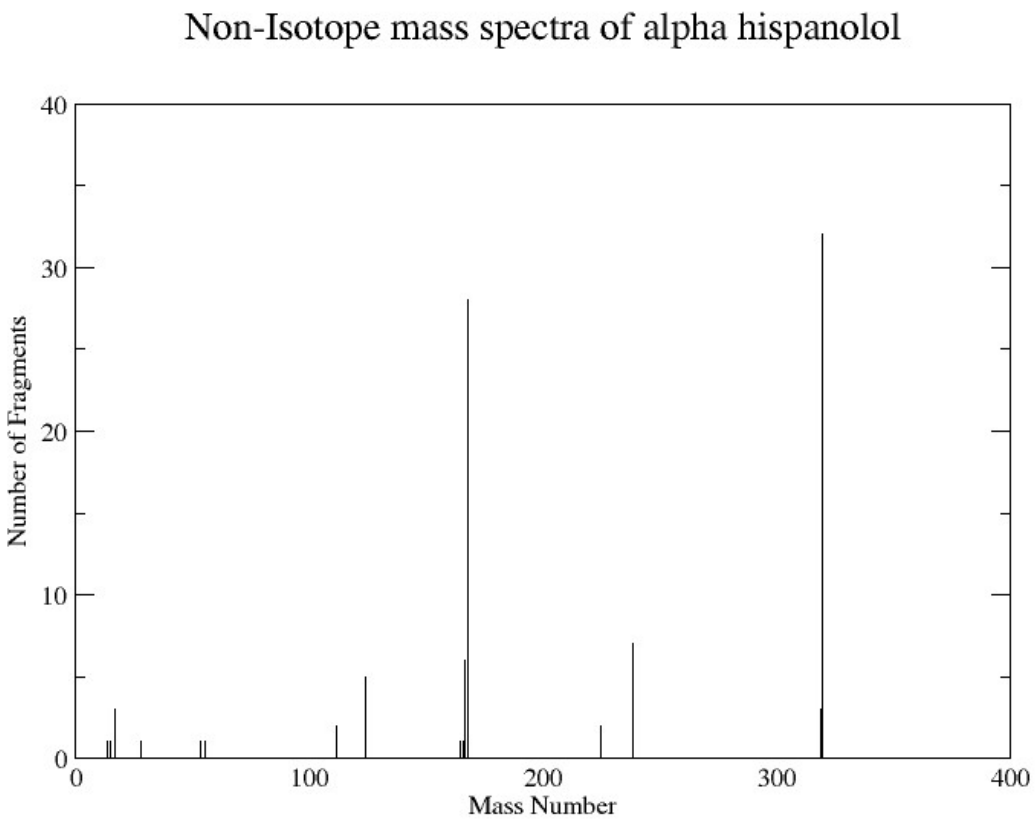


Figure 2.11: Theoretical mass spectra of alpha hispanolol with exclusion of isotopic abundance

2.5.1 Inclusion of Isotopic Abundances

Application of isotopic distributions is important in mass spectrometry in order to improve the details of predicted spectra to match experimental observations. As such no prediction of mass spectra without calculation of isotopic distribution will achieve accuracy with relation to experimental mass spectra. MS also involves study of increasingly larger biomolecules where isotopic distributions have an increased effect with a wider range of

isotopic variants which makes it important to take into consideration. Thus, analysis of fine isotopic distribution is absolutely essential (Dittwald *et al.,* 2015).

The importance of stable isotope distributions within mass spectrometry goes beyond systematic determination of natural abundances; stable isotopes play important roles in many structural elucidation studies. For example incorporation of artificial isotopic compositions synthetically into compounds (labelling a particular atom or set of atoms atoms) may provide information about reactivity at sites in a molecule or may allow tracer studies in the investigation of physiological processes in a safe and non-invasive manner (Bluck & Volmer, 2009). It is not possible for MS to detect one single molecule, but it depends on the availability of a number of identical copies of some molecules. These copies are identical only from a chemical point of view, the elements within these copies follow their natural isotope abundance. This is why in MS instead of a single peak per fragment, an isotope pattern of the fragment is observed.

In the present study isotopic distribution of each fragment produces in all the molecules was calculated in order to improve the accuracy of the predicted mass spectra.

The isotope array to set the mass spectra was offset starting with atomic number zero to match a zero based array; real isotopes were included from array value 1 matching hydrogen's isotope distribution (Figure 2.12)

```
base_values=[0,1,3,6,9,10,12,14,16,19,20,23,24,27,28,31,32,35,36,39,40,45,46,50,
50,55,54,59,58,63,64,69,70,75,74,79]

isotopes=[[0.0,0.0,0.0,0.0],[0.9998,0.0002,0.0,0.0],[0.000002,0.999998,0.0,0.0],
[0.0759,0.924,0.0,0.0],[1.0,0.0,0.0,0.0],[0.2,0.8,0.0,0.0],[0.989,0.011,0.0,0.0]
,[0.996,0.004,0.0,0.0],[0.9976,0.0004,0.002,0.0],[1.0,0.0,0.0,0.0],[0.9048,0.002
7,0.0925,0.0],[1.0,0.0,0.0,0.0],[0.79,0.1,0.11,0.0],[1.0,0.0,0.0,0.0],[0.922,0.0
47,0.031,0.0],[1.0,0.0,0.0,0.0],[0.9499,0.0075,0.0425,0.0],[0.76,0.0,0.240,0.0],
[0.00334,0.0,0.00063,0.0],[0.93258,0.00012,0.000673,0.0],[0.96941,0.0,0.0,0.0],[
1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],[0.0025,0.9975,0.0,0.0],[0.0435,0.0,0.83789,0
.09501],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],
[0.6917,0.0,0.3083,0.0],[0.0,0.0,0.0,0.0],[0.6011,0.0,0.3989,0.0],[0.0,0.0,0.0,0
.0],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],[0.51,0.49,0.0,0.0]]
```

Figure 2.12: Base value indicates the lowest isotope mass, e.g. for H it is 1, for He is 3. In the isotopes array, the first value is the abundance for the lowest isotope, the next for mass+1 etc.

The current implementation therefore can only handle up to Bromine, and a variation of four mass units for four isotopes; extension to heavier elements is not difficult, as is inclusion of a wider mass range.

Figure 2.13 illustrates how a mass spectrum is distributed according to the isotope distribution of one atom. Repeating this code for all atoms in the fragment will appropriately proportion the mass spectrum into the correct isotope distribution.

```
def expandlistofvalues(mass_distribution,atom):
   #NB start maybe with [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] (100%
at mass 0)
   #now we add an atom to it for example a C
   #print("--------expandlistofvalues--------------------------------
--------")    change=[0.0]    for i in range (0,10000):
      change.append(0.0)
      #we have created a change matrix to determine the next stage
of the mass        spectrum    for j in range (0,10000):
      if mass_distribution[j]>0.0:
      #only four possible isotopes for each element, so we create
four new        peaks from the existing peak
      #every time we add an atom to the molecular
formula       for k in range (0,4):
         base=j+base_values[atom]+k#12+0,12+1,12+2,12+3

change[base]=change[base]+mass_distribution[j]*isotopes[atom][k]
change[j]=change[j]-mass_distribution[j]
         #this last line is to reduce the original peak since it
now creates           the four new peaks. return change
```

Figure 2.13: Adding the isotope distribution for a single atom to the spectrum

Figure 2.14 shows the predicted mass spectrum for α-Hisplanalol with the inclusion of isotope distribution. This may be compared to Figure 2.10 where the isotope distribution is not present.

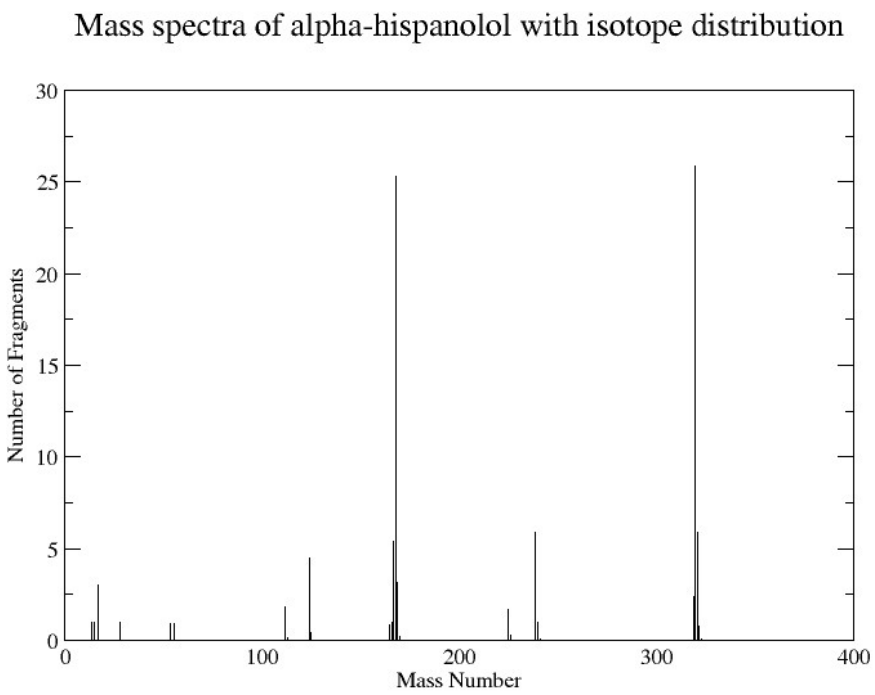


Figure 2.14: The theoretical mass spectra of α-hispanolol with isotopic distribution

The intention of using α-hispanolol in terms of mass spectral prediction was access to mass spectra for this compound under a wide variety of experimental conditions. Unfortunately, our EIMS instrument was unexpectedly down during the time the spectra would have been acquired. However, we were able to obtain an ESI spectrum of α-hispanolol. There are two aspects to consider - firstly ESI will produce even electron species (as compared to the odd electron species for which our predictions are based) and secondly the ESI spectrum produced the M+23 peak (due to the presence of sodium). This certainly provides paths for extension of this method for future work. Figure 2.15 shows the ESI spectrum of α-hispanolol.

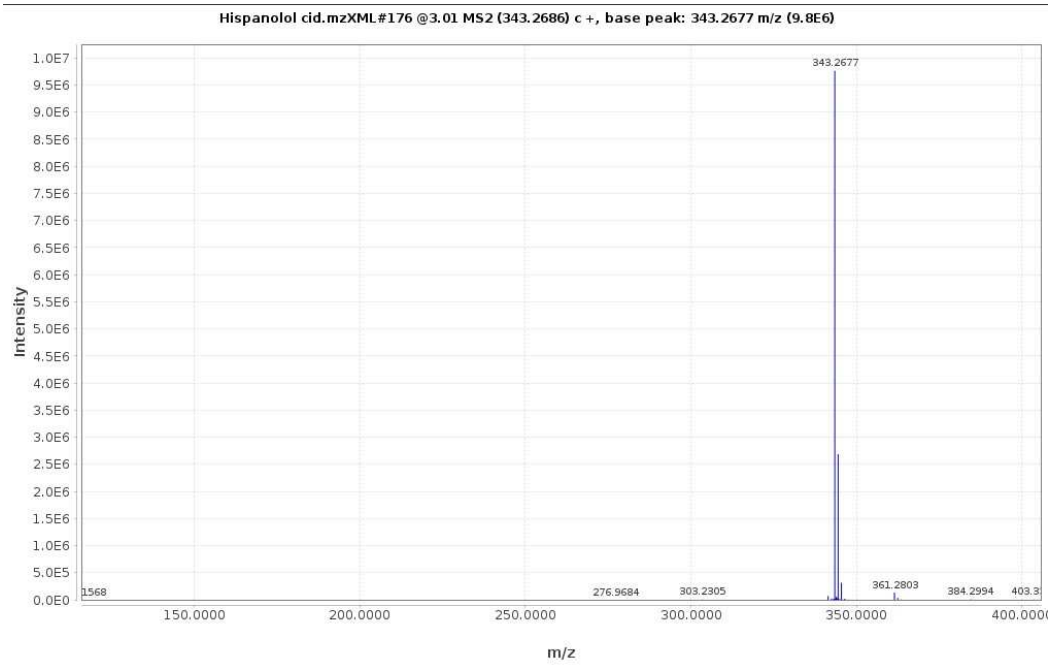


Figure 2.15: showing the ESI mass spectrum of alpha hispanolol

Given the presence of sodium, there is a shift between the molecular ion on the predicted mass spectra of mass 320 and the peak on the experimental mass spectra of 343.2677. However, given that ^{23}Na has 100% natural abundance, the isotope distributions for these two peaks should match between the experimental and predicted spectra, as is observed. In the ESI spectrum, there was no fragmentation which is quite typical for ESI spectrum, due to the lower energies imparted to the ions from a much softer ionization technique, and this explains the reduced detail between the theoretical mass spectra when the temperature was at 2000K and the experimental spectrum which only has an isotope cluster around the M+23 ion.

Figures 2.16 and 2.17 are the predicted mass spectra, from 100 trajectories (about 48 hours computational time) of 10000 steps each at the DFTB+ level of theory.

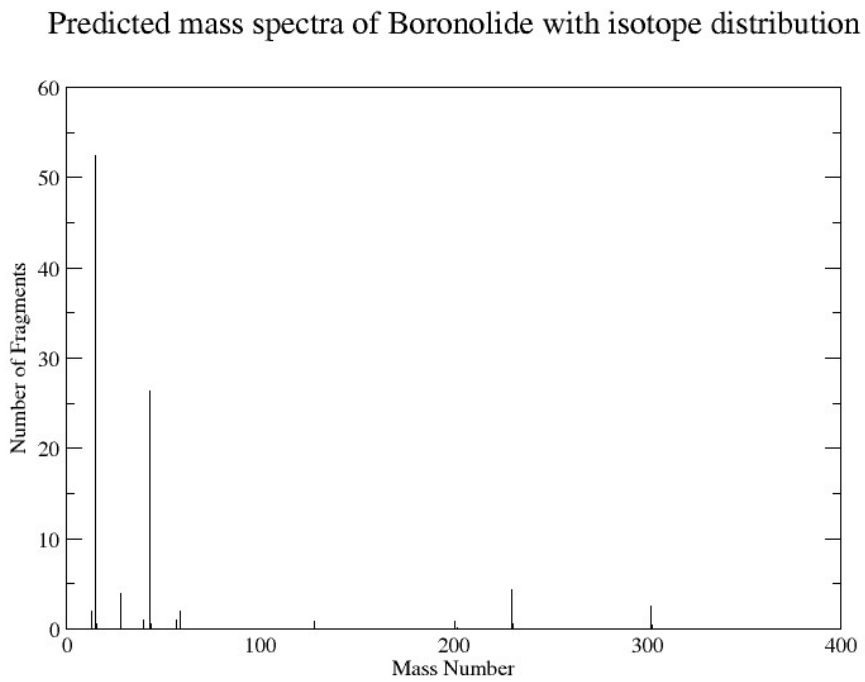


Figure 2.16: Theoretical mass spectra of boronolide at 2000K

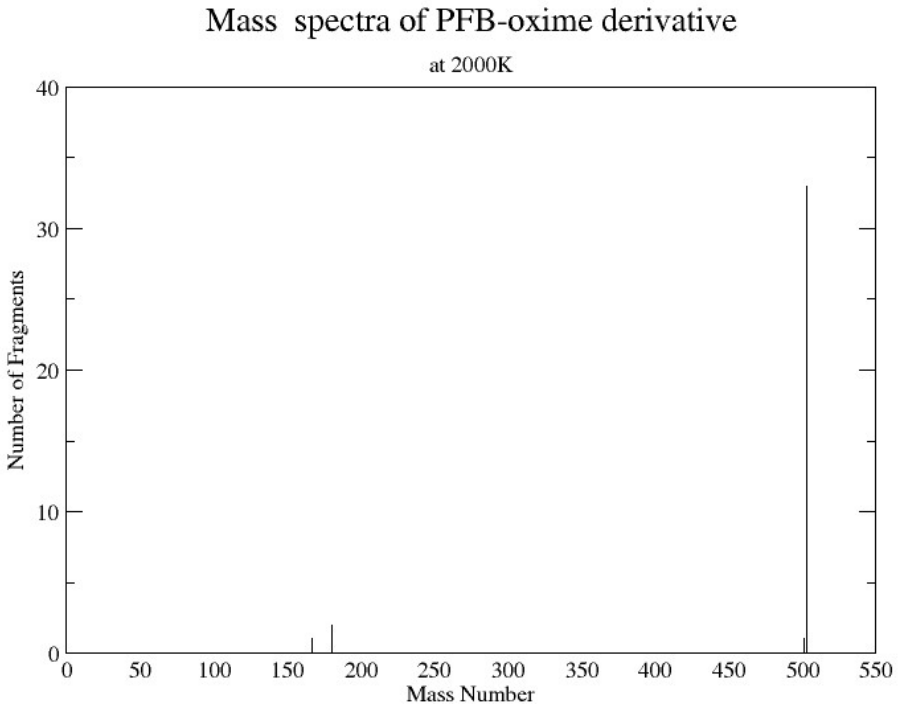


Figure 2.17: Predicted mass spectra of PFB-oxime derivative at 2000K

2.5.2 Prediction of Mass Spectra from the NIST databases

Given the lack of experimental data for the compounds we were focussing on, we turned to the NIST database in order to explore if our results for the predicted mass spectra could accurately match experimental mass spectra available from the database itself.
The two compounds chosen were 5-amino-1- (Phenylmethyl)- [1,2,3]triazole-4-carboxamide, and ethyl 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate. Figure 2.18 and 2.19 show the 2000 K predicted, and the experimental EI mass spectrum respectively.

In the predicted mass spectra of 5-amino-1- (Phenylmethyl)- [1,2,3]triazole-4-carboxamide, there were two most intense peaks were observed at *m/z* 91 (which was also intense in the experimental mass spectra as the base peak) and at *m/z* 217, which, although observed in the experimental mass spectra, was observed experimentally with a much lower intensity.

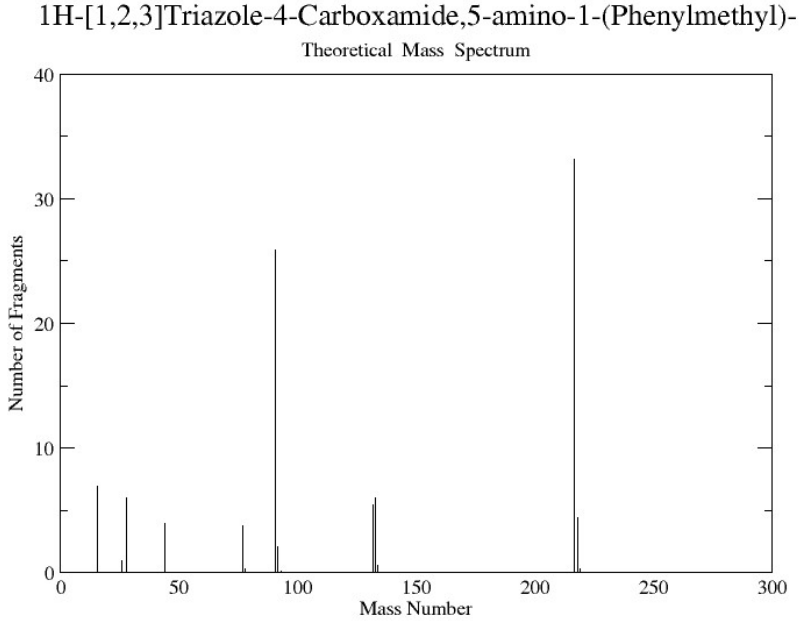


Figure 2.18: Predicted mass spectra of 5-amino-1-(Phenylmethyl)- [1,2,3] triazole-4-carboxamide, at 2000 K

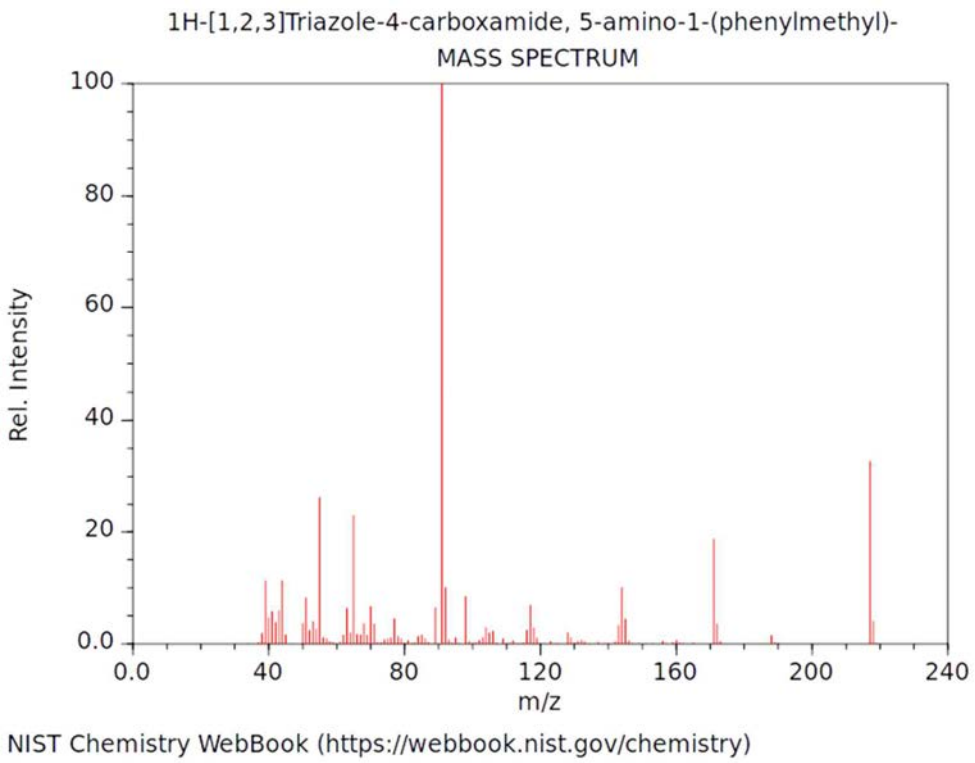


Figure 2.19: Experimental 5-amino-1-(Phenylmethyl)-[1,2,3]triazole-4-carboxamide mass spectrum (NIST)

The match between the experimental and theoretical spectra in terms of fine structure is far from perfect and it is clear that focus needs to turn now to the distribution of ion temperatures following instantaneous ionization, including ground state dynamics velocities at the starting point. However, it was interesting to explore the limits of prediction of the current method as it stands in the context of ethyl-3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate. Figures 2.20 and 2.21 show the theoretical and experimental spectra for this compound respectively.

As a point of reference, the doubling of the molecular ion in to M and M+2 is reproduced according to the isotopic distribution routine. However there are too few (although correct) peaks in the theoretical spectrum. Given the complexity of the experimental spectrum, it may have been worth exploring 1000 trajectories, but given the 20 days computation time for this, this was not feasible.

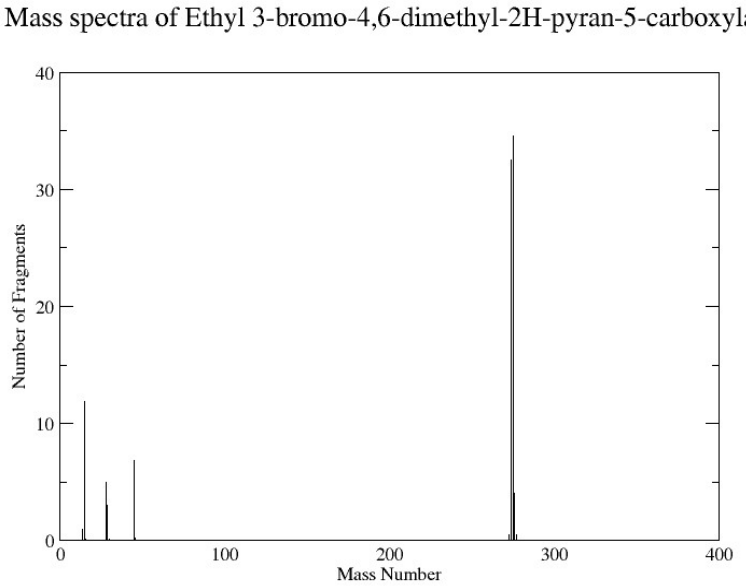


Figure 2.20: Theoretical mass spectra of ethyl 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate at 2000K

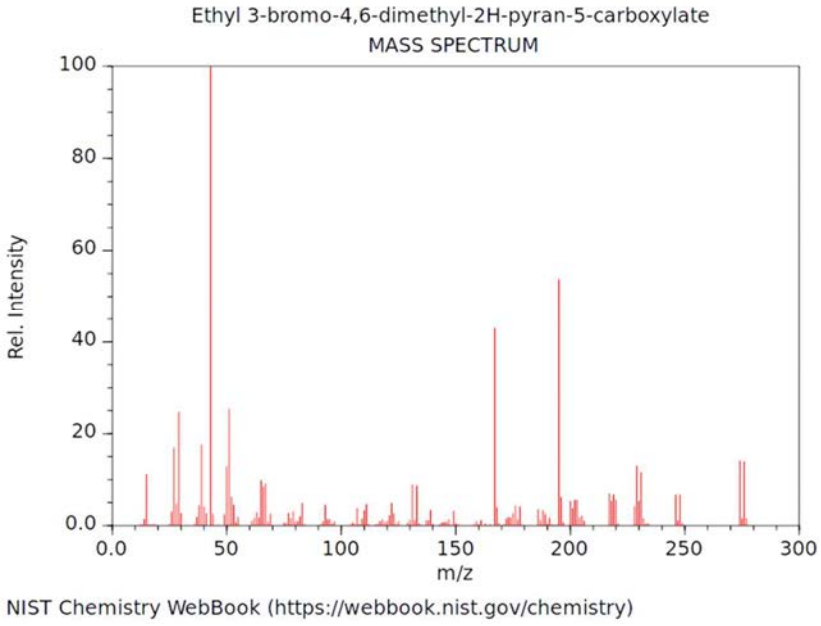


Figure 2.21: Experimental mass spectra of 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate from NIST database (NIST)

Other strategies were therefore followed. The theoretical mass spectra, as a first step, were predicted at different temperatures to investigate the extent of its effect on the predicted mass spectra. The first attempt to alter the predicted spectra in this way was for the mass spectra of ethyl 3-bromo-4,6dimethyl-2H-pyran-5-carboxylate at 3000 K (Figure 2.22).

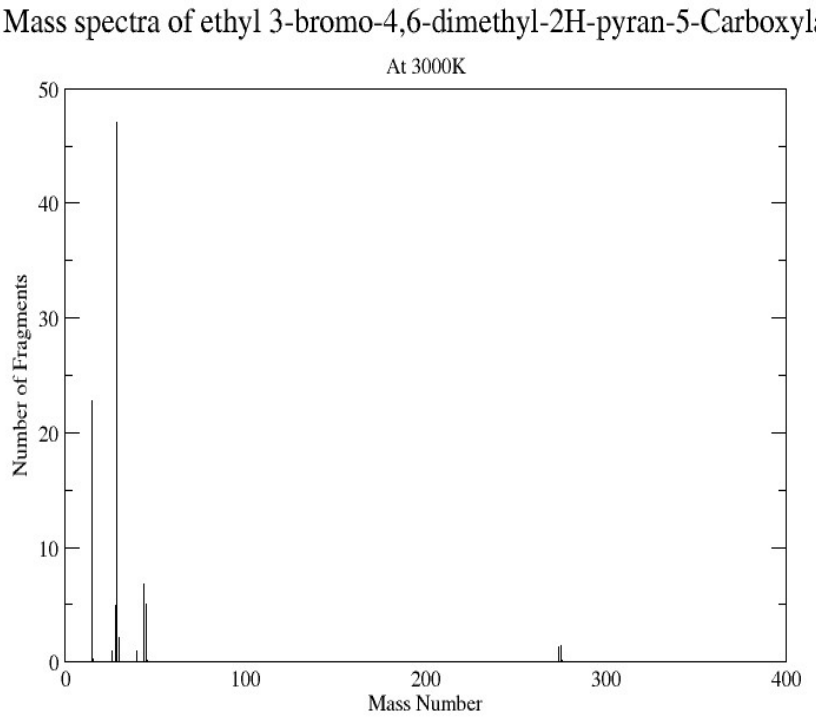


Figure 2.22: Theoretical mass spectrum of 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate at 3000 K.

This change in theoretical mass spectrum from 2000 K to 3000 K was remarkable for 3-bromo-4,6dimethyl-2H-pyran-5-carboxylate; rather than most trajectories producing the molecular ion, almost all trajectories produced the smaller fragments already observed at 2000K. As mentioned earlier in the thesis the use of higher temperatures was in one respect to accelerate the fragmentation and to reduce the simulation time. However, at 3000K too much fragmentation of the molecular ion was observed and the theoretical mass spectra of 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate was still not close to the experimental mass spectra. To at least achieve a balance between the molecular ion and

the observed fragments we attempted to run the MD at an intermediate temperature, between 2000 K and 3000 K.

As such we ran the mass spectral prediction of 3-bromo- 4,6-dimethyl- 2H-pyran-5-carboxylate at 2200 K, providing the closest match to the experimental spectra in terms of intensities of the peaks produced in the theoretical spectrum (Figure 2.23).

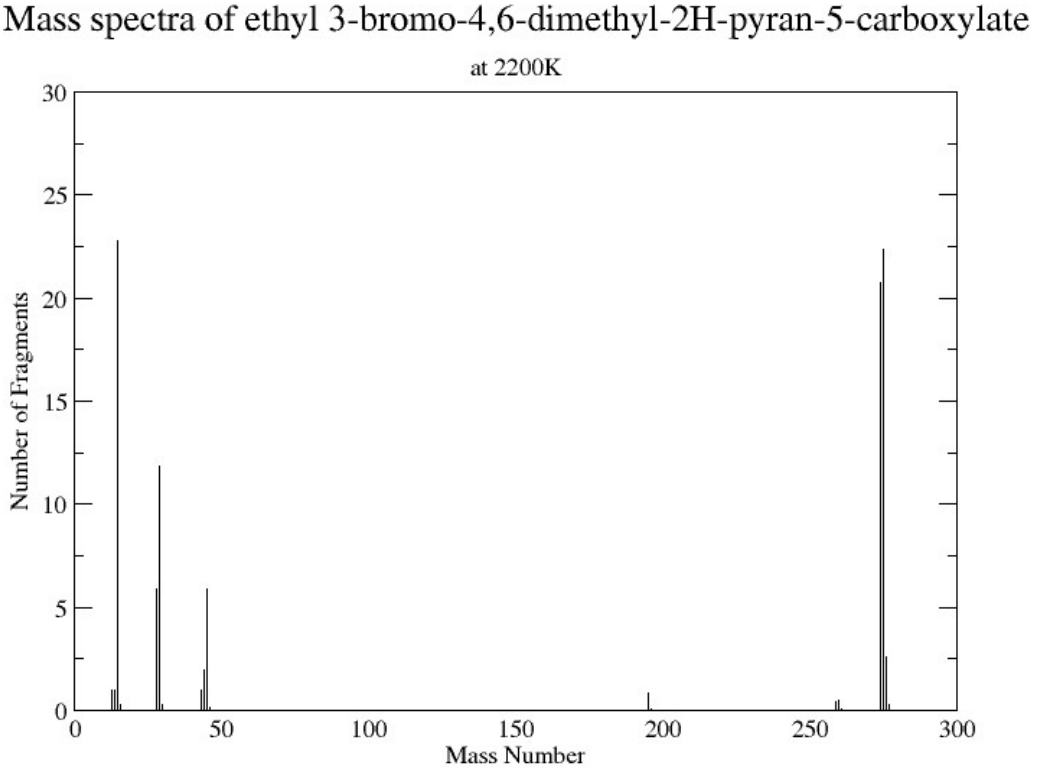


Figure 2.23: Theoretical mass spectra of 3-bromo-4,6-dimethyl-2H-pyran-5-carboxylate at 2200K

Several strategies are in place for continuation and improvement of the quality of theoretical spectra produced. The first focuses on production of the appropriate ion ensemble following ionization. Detection of fragmentation events in dynamics may allow for termination of trajectories earlier, accelerating the acquisition of multiple trajectories. Running more trajectories is likely to improve detail in the mass spectra produced. Choice of model from DFTB+ through to XTB may favour different fragmentations and therefore change the quality of spectra. As soon as the quality of the spectra is of a sufficient

standard, this implementation of QCEIMS will be compared to literature implementations of this method.

Other points of interest could be introducing flexibility in the ionization, and allowing the procedure to be used in ESI mass spectral prediction, or introduction of QM/MM for larger molecules to be evaluated. Further, a variety of chemistry will be explored, including the evaluation of metal containing compounds such as organometallic systems. Finally, given that the molecular dynamics informs fragmentation pathways, setting up pathways statically on the potential energy surface, may allow prediction of rates of formation of fragments, from the vibrational analysis, using RRKM theory (Knyazev & Stein, 2010).

CHAPTER 3

In this chapter, the feasibility of providing sets of fragments for use in drug discovery is explored. The two targets chosen for drug discovery are quite different, but a background to the technique of docking and how the fragments produced in mass spectral theoretical spectra could be used in this technique are presented. It is interesting to note that the word "fragment" has slightly different meanings in mass spectrometry (where it is used to denote the daughter ions and neutral species formed from breaking a larger molecule) as compared to drug discovery (where small molecule fragments with biological activity are combined through connections to construct a larger, highly active drug).

3.1 Fragment Docking

Molecular docking is used to predict the atomic coordinates of a protein-ligand complex (Brooijmans & Kuntz, 2003; Rognan, 2017). Fragment-based drug discovery (FBDD) has been increasingly studied for the early phase drug design, this is because fragment space can be more effectively sampled than drug-like space (Sandor *et al.*, 2010). Fragment docking may be challenging due to the sheer number of interaction sites on the surfaces of proteins which may accommodate low molecular weight compounds (English *et al.*, 2001). Further, the binding cavity of the target protein may be much larger than the molecular volume of the fragments resulting in incorrect binding modes (Nayal & Honig, 2006). Some docking programs are able to overcome these challenges in fragment docking. Programs like Gold, FlexX, Surflex and Glide have proven capability in pose prediction for fragments (Gilles & Didier, 2007). Glide in particular, has been well tested with fragment docking. For example, Glide XP in a study docked fragments to the binding site of twelve targets with RMSD within 1Å of the experimental binding mode (Loving *et al.*, 2009).

Fragments (in this context) are polar compounds with low molecular weight and low complexity. This makes them more efficient in sampling chemical space, both in terms of physicochemical/ADMET properties and in terms of binding to proteins. There are two

main strategies of fragment exploration, growing and linking. In the first case, a potential fragment is decorated with additional functionalities to build on promising activity, while in the latter case one two or more promising fragments that bind to separate portions of the target protein in close proximity are subsequently incorporated into a single molecule using a suitable linker moiety (Vass *et al.*, 2014).

Docking available small compounds (<250-300 Da) is an interesting and difficult combinatorial problem due to the immense number of compounds that exist - the size of the GDB17 database of small molecules (166 billion) illustrates this (Ruddigkeit *et al.*, 2012). However, the advantages of fragment docking include low cost, high speed and these experiments are not bound by some experimental limitations, such as the solubility of ligands. Molecular docking is used to screen fragment libraries, identifying inhibitors against targets. The fragment-based procedure is illustrated in Figure 3.1 (Chen & Pohlhaus, 2010).

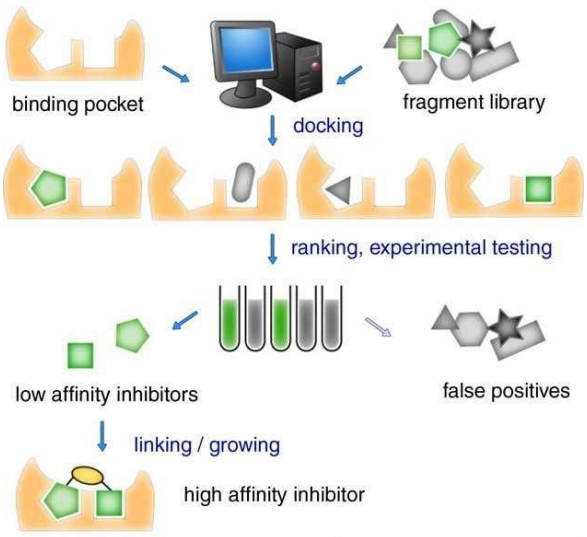


Figure 3.1: Screening of fragment libraries using molecular docking allows for the selection of compounds that complete sub-sites in the target protein binding site (Chen & Pohlhaus, 2010).

3.2 Targets identified for fragment docking

Two targets were chosen for proof-of-concept, *Plasmodium Falciparum* DXR and HIV-1 Protease.

These two targets are well studied in terms of rational drug design.

3.2.1 *Plasmodium Falciparum* DXR

Plasmodium falciparum (*P. falciparum, Pf*) is one of the protozoan parasites responsible for malaria. More than a million deaths occur each year as caused by malaria (Balconi *et al.,* 2009). Humans are infected by *P. falciparum* through bites by infected mosquitoes, which release sporozoites into the blood stream. These then affect the liver cells of humans, where the sporozoites develop into merozoites which leave to the red blood cells in trophozoite and schizont stages of its life cycle (WHO, 2017; Wang *et al.,* 2015).

DXP reductoisomerase (DXR) is an enzyme in the non-mevalonate pathway for the biosynthesis of isoprenoids; this pathway is present in some bacteria and protozoa (including malarial protozoa) but not in humans, making *Pf*DXR an attractive target for malarial therapy (Lell *et al.*, 2003). The antibiotic fosmidomycin [3-(*N*-formyl-*N*-hydroxyamino)propyl-phosphonate] has been reported to be a potent inhibitor of this enzyme (Kuzuyama *et al.,* 1998). It was also reported by Jamaa *et al*., that fosmidomycin and a related compound FR900098 were able to inhibit *Pf*DXR activity, to suppress the *P. falciparum.* The enzyme *Pf*DXR itself is a homo dimer when it is active where every subunit includes a NADPH cofactor together with a divalent metal ion. It has a molecular mass of approximately 47kDa. Its structure is similar to that of DXRs from other species. There are two large domains in the subunit of *Pf*DXR that are separated by a cleft with a deep pocket, a linker region and a small c-terminal domain that can be illustrated in Figure 3.2:

Figure 3.2: *Pf*DXR (PDB ID: 3AU9), showing the NADPH-binding domain (blue), the catalytic domain (green), the linker domain (yellow) and the C-terminal domain (red). NADPH is shown as "balls and sticks", the divalent magnesium as a sphere, and the fosmidomycin as tubes (Umeda *et al.,* 2013).

3.2.2 HIV-1 Protease

Human immunodeficiency virus (HIV) is the causative agent for acquired immunodeficiency syndrome (Gallo *et al*., 1983). HIV-1 protease is an essential enzyme in the HIV life-cycle involved in processing the gag and gag-pol gene. Its purpose is to cleave these genes producing the essential viral proteins that are required for the assembly of a new mature virus. As such inhibition of HIV-1 protease is an important part of controlling the progression of HIV-1 (Ghosh *et al.*, 2017, Brik & Wong, 2003).

HIV protease is a homodimer with two 99 amino acid chains and a C2-symmetric substrate binding pocket. The HIV active site has conserved binding triad residues, t D25/D25'-T26/T26'-G27/G27', of which the D25 and D25' (aspartate) residues are critical for the catalytic mechanism (Ahmed *et al.,* 2013). HIV-1 subtype C protease is mostly found in South Africa (c-SA) and is exhibiting resistance to common drug

regimens (Makatini *et al.,* 2001). A reason for the low binding affinity of inhibitors to the African prevalent HIV-1 subtypes A and C could be due to the development of these inhibitors in the context of subtype B, common in North America and Western Europe (Makatini *et al*., 2013).

Figure 3.3 shows the structure of HIV-1 protease (PDB ID: 1W5X).

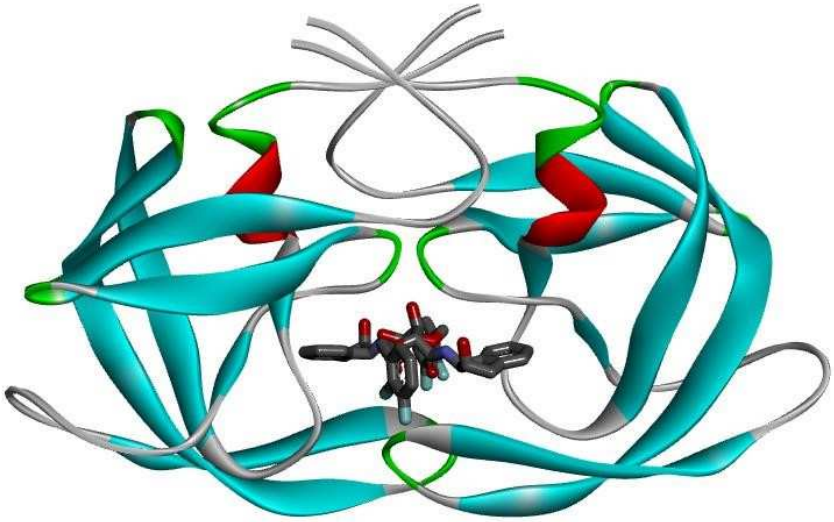


Figure 3.3: The homodimeric structure of HIV-1 protease (PDB ID: 1W5X) in complex with the protease inhibitor (2R,3R,4R,5R)-2,5-bis[(2,3-difluorobenzyl)oxy]- 3,4-dihydroxy- N,N'- bis[(1S,2R)- 2- hydroxy- 2,3- dihydro- 1H- inden- 1-yl] hexanediamide.

3.3 Methodology

Figure 3.4 illustrates the methodology followed, from collation of fragments from MS prediction through to preparation of both ligands and proteins for high throughput virtual screening (HTVS) using a molecular docking approach.

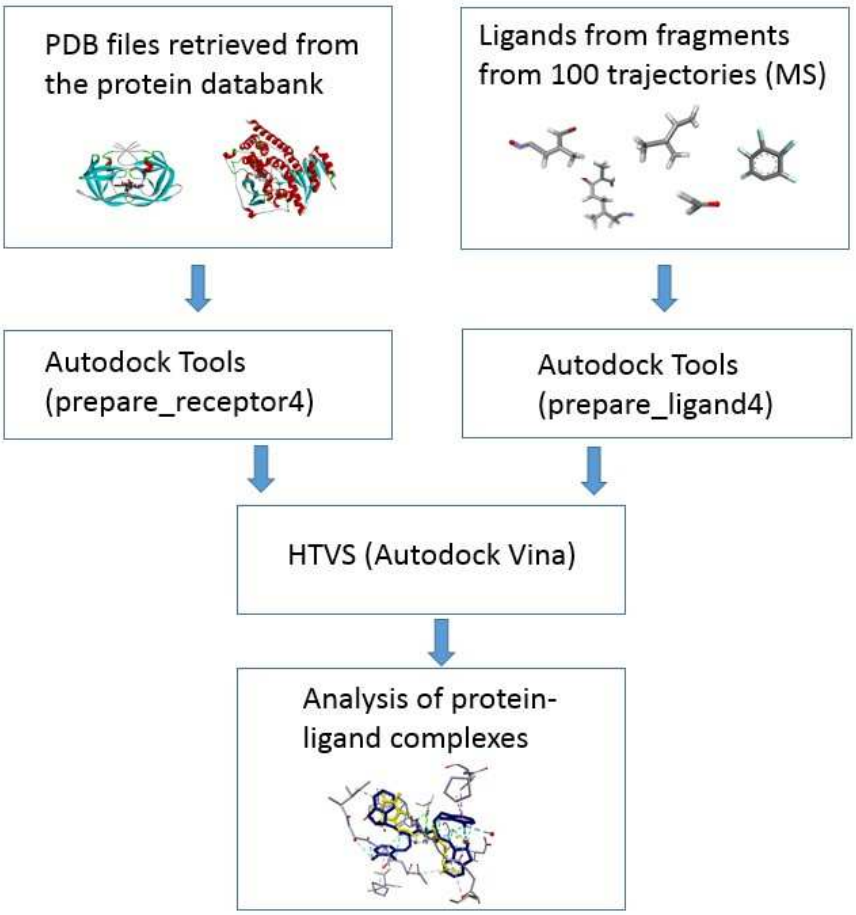


Figure 3.4: The steps and tools that were used for docking the fragments.

3.4.1 Vina Docking

Autodock Vina (http://vina.scripps.edu.) was used to perform fragment docking of the compounds, and the Vina scoring function was used to evaluate the docking between the unfragmented molecule and the fragments to the respective targets.

When Autodock Vina was developed, optimization of its efficiency came from many angles; from global optimization approaches through to genetic algorithms for example. Autodock Vina, for convenience, constructed with the ability to work with the file formats

that are used in older docking programs, for instance AutoDock 4 (Morris *et.al.,* 1998). Due to this, Vina works well with protein and ligand preparation tools available in auxilliary software toolkits, such as AutoDock Tools which handles preparation of the pdbqt file formats required by the docking software. Autodock tools was used in this study to prepare receptor and ligand pdbqt files.

3.4.1.1 Ligand Preparation

The ligands for docking were obtained from the fragments produced from all AIMD trajectories used to determine a mass spectrum. The fragment SMILES (from RDKit), was taken as text; however the fragmentation sites were marked in the SMILES as dummy atoms "*", and to create neutral non-radical species (molecules) these dummy atoms were replaced with "H" completing the molecule; the resultant SMILES fragments were pushed into a 3D format and optimized using the UFF force field. Finally the cleaned small molecules were written as pdb files for use in docking. Figure 3.5 shows this molecule preparation within the scheme of work.

```
     docking_count=0

      #######    Molecules
for                docking
broken_molecule=""
molecules_for_docking=""
if  bond_broken_list:
            #initial work for MS
            broken_molecule = AllChem.FragmentOnBonds
                                    (sanccompound,bond_broken_list,False)

            #collect fragments for docking
            molecules_for_docking = AllChem.FragmentOnBonds

(sanccompound,bond_broken_list,True)                  else:
            broken_molecule=sanccompound
      molecules_for_docking=sanccompound

      fragment_smiles=AllChem.MolToSmiles(broken_molecule)
#For MS      #For docking:
docking_smiles=AllChem.MolToSmiles(molecules_for_docking)
docking_smiles=re.sub("\*","H",docking_smiles)
#print("docking smiles ", docking_smiles)
list_for_docking=docking_smiles.split(".")

     for docking_molecule in list_for_docking:
                  print("docking molecule : ",docking_molecule)
mol=AllChem.MolFromSmiles(docking_molecule)
#molH=AlChem.AddHs(mol) # not necessary now
                 AllChem.EmbedMolecule(mol)
AllChem.UFFOptimizeMolecule(mol)

moleculename="docking/molecule_"+str(docking_count)+".pdb"
docking_count=docking_count+1
writer=AllChem.PDBWriter(moleculename)
writer.write(mol)
```

Figure 3.5: The portion of python script that was used for preparing the pdb files of the ligands for docking.

After obtaining the pdb files, the ligands were then prepared using the Autodock Tools (MGL Tools), **"prepare_ ligand.py -l filename"** producing the appropriate pdbqt files. Thus all ligands were prepared in terms of rotating bonds. In order to apply this across all the fragments we used a controlling python script to manage this for all fragments.

3.4.1.2 Protein Preparation

The protein structures were retrieved from the RCSB protein databank; for HIV-1 protease PDB ID: 1W5X was used, while for Plasmodium falciparum DXR PDB ID: 4Y6R was used. After protein structure retrieval the receptors were prepared for docking using Autodock Tools (prepare_receptor4.py -r receptor-file) to generate receptor pdbqt files. The proteins are prepared such that atom typing is appropriate for scoring the docking poses.

3.4.1.3 Docking Procedure

After the proteins and ligands were prepared, targeted docking was performed. Autodock Vina input files were prepared for all receptor ligand pairs using a python script. The search box centre xyz coordinates were obtained during initial examination of the receptors using Discovery Studio Visualizer. In terms of docking parameters, the exhaustiveness was set at 16, while the search area was set according to Table 3.1. The Vina input files were then submitted to a batch queue for docking on the local departmental cluster.

Protein	Box size Å	Centre xyz
1w5x	x = 20 y = 20 z = 20	x = 12.88 y = 22.66 z = 5.62
4y6r	x = 20 y = 20 z = 20	x = -11.85 y = 22.57 z = -10.38

Table 3.1 : Parameters that were used in the docking of 1w5x and 4y6r

3.5 Results

3.5.1 Docking Validation

As a validation of the docking procedure, we re-docked the original crystal structure ligands of the proteins into their targets to validate our docking procedure. The docking procedure should reproduce the docking pose found experimentally for the docking workflow to be considered acceptable. After redocking the ligand of 1w5x to its original protein the two ligands showed a nearly perfect overlap (Table 3.2) with a binding energy of -11.9 kcal/mol. Redocking of the ligand of 4y6r resulted in a changed position, although docking was acceptable in the active site as shown in Table 3.2. The reason for the change of pose is the size of ligand for 4y6r, given the many torsions and the flexibility of the ligand, a much more exhaustive search is required for accurate pose predictions. This extended search would go far beyond that required for our fragment docking, and so it was felt that the docking procedure was adequate under the circumstances.

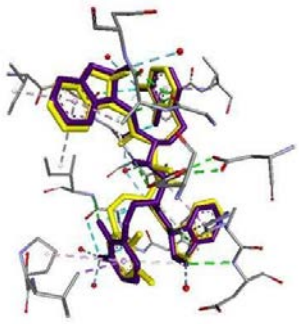
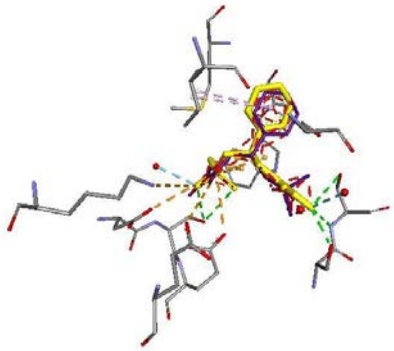
Protein	Protein-ligand interaction	Binding energy (kcal/mol)
1w5x		-11.9
4y6r		-6.1

Table 3.2: Docking validation was done using the original ligands of these two proteins which were obtained from PDB databank in order to check if our docking procedure was working correctly

3.5.1 Docking Results h

The binding scores that were obtained from docking the fragments (generated from 100 different trajectories each of PFB oxime, alpha hispanolol and boronolide) were collated (see Appendix 3, for example, for a full set of binding data for boronolide fragments). During molecular docking, the fragments were not filtered for uniqueness, so it was interesting to explore whether fragments that were docked more than once provided the

same binding energies, or whether the different conformations of the fragments biased the docking to result in different binding energies. However, the main focus was on the best binders of each set of fragments for the three SANCDB molecules.

3.5.2 Binding of PFB-oxime and its fragments to the two targets

The original PFB oxime molecule was docked into the two different targets, which are HIV-1protease (1w5x) and also plasmodium falciparum DXR (4y6r), in both proteins this molecule did bind to the active site. Since we were exploring the advantages of binding fragments to the targets, these two dockings were used as reference. In terms of binding energies we observed values of -8.5 kcal/mol (for binding of PFB oxime to 1w5x) and -8.2 kcal/mol (for binding of PFB oxime to 4y6r). Table 3.3 shows the binding modes within a subset of the protein. In the figures in this table the docked PFB oxime is yellow while the original crystal structure ligand in its experimental pose is blue. The crystal structure ligands are shown in order to illustrate how the original ligand is bound to the active site.

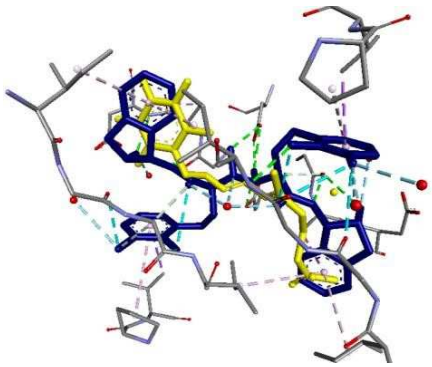
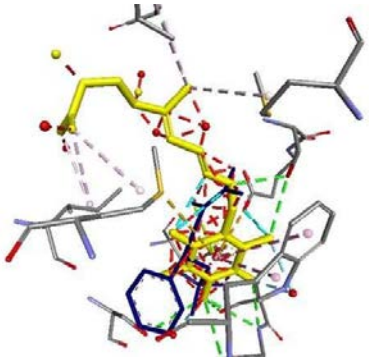
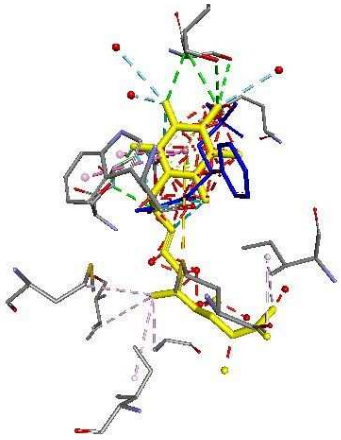
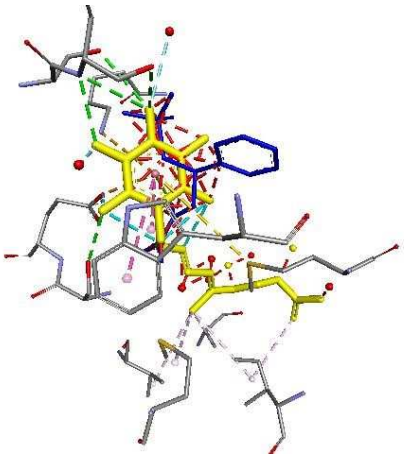
Macromolecule	Protein-ligand interaction	Binding energy (kcal/mol)
1w5x		-8.5
4y6r		-8.2

Table 3.3: Binding of PFB to 1w5x and 4y6r proteins

It is interesting to note the results of docking fragments of PFB oxime to these two targets. The best protein-fragment binding results are presented in Table 3.4. The fragment name "molecule10" resulted in a calculated binding of -8.7 kcal/mol to the target 1w5x. This is successful in the sense that using fragments from the MS prediction in docking has identified a slightly smaller system that has improved binding characteristics with respect to 1w5x. In Table 3.4 binding interactions are also shown to a subset of residues from the active site. Fragments are shown in yellow while crystal structure ligands are shown with a blue colour. The crystal structure ligand helps to show the active site of the protein and how it is bound to the active site and makes it easier to compare its binding to how the fragment is binding. In the following table 3.3, the H atoms are not shown as non-bonded

for example molecule 10 the Br is bonded but the software has not identified the bond and therefore not drawn it in.

Protein-fragment	Protein-fragment	Binding Energy (kcal/mol)
4y6r-molecule2		-8.4
4y6r-molecule29		-8.3

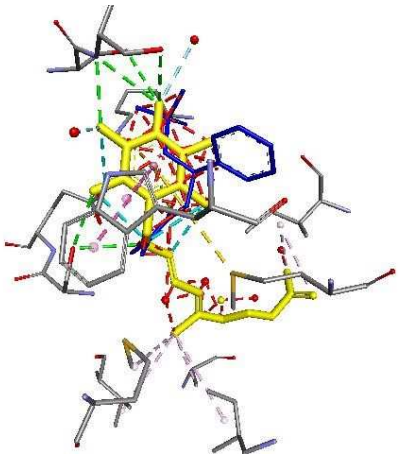
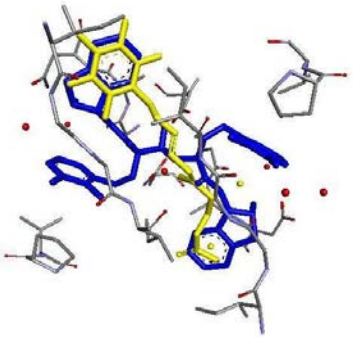
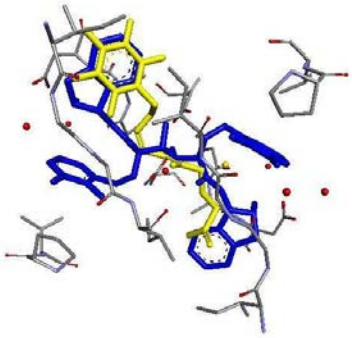
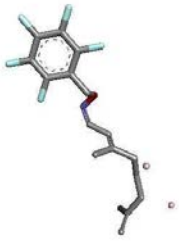
4y6r-molecule31		-8.1
1w5x-molecule8		-8.6
1w5x-molecule10		-8.7 

Table 3.3: Best results from docking PFB oxime fragments to the two target proteins.

3.5.3 Binding of α-hispanolol and its fragments to the two targets

Fore α-hispanolol the results are even more marked. Upon docking α-hispanolol itself to the two proteins, although the binding to 1w5x is reasonable (-7.1 kcal/mol), binding is quite poor to 4y6r (-5.0 kcal/mol). For 1w5x the mode of binding of α-hispanolol matches to some extent the original crystal structure. However, for binding to 4y6r, α-hispanolol did not bind to the active site of this protein. This is attributed to the large size of α-hispanalol relative to the active site.

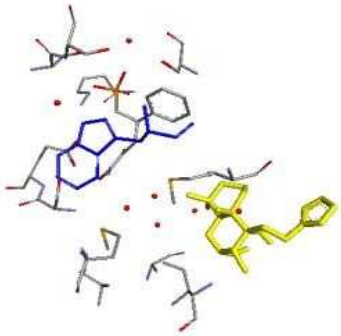
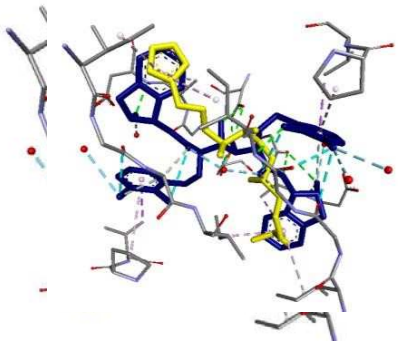
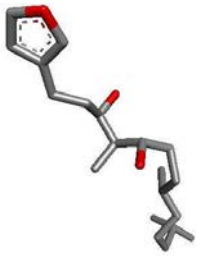
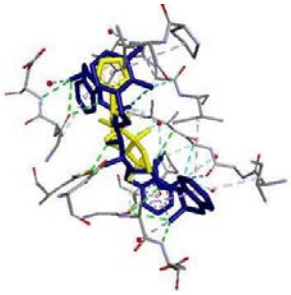
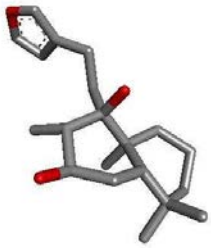
Macromolecule	Hisp-macro interaction	Binding energy (kcal)
1w5x		-7.1
4y6r		-5.0

Table 3.6: The calculated binding modes off α-hispanolol to the two proteins, together with the associated binding energy

The fragments that gave the best binding energies after docking to the two targets are shown in Table 3.7. The fragment docking provided species with much better binding to both proteins than the original α-hispanolol.

The interactions of these best-binding fragments with the macromolecules were viewed using Discovery Studio and this information is also included in Table 3.7. In the table the fragments docked are yellow while the original crystal structure ligand is blue. Fragmenting α-hispanolol reduced the size to the point that fragments were now able to bind with good binding characteristics to 4y6r (-7.2 kcal/mol is the best for fragment 204).

Fragment number	Fragment interaction	
1w5x-154		 -7.4
1w5x-174		 -7.8

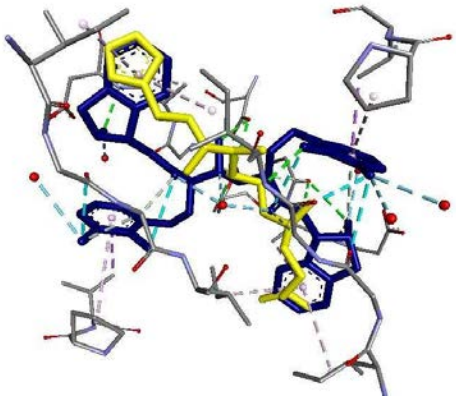
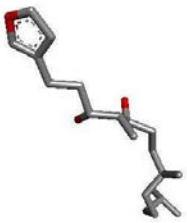
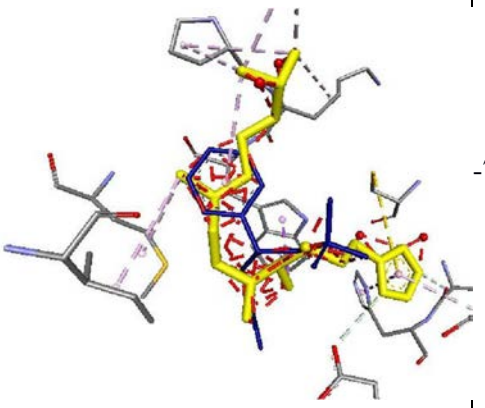
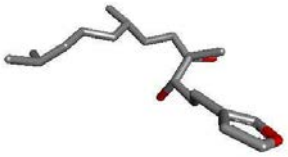
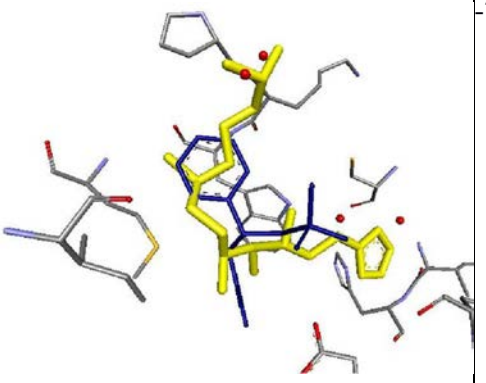
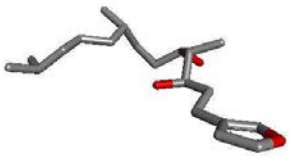
1w5x-279		-7.5	
4y6r-120		-7.0	
4y6r-204		-7.2	

Table 3.7: Best results from docking of fragments of α-Hispanalol to the two targets..

3.5.4 Binding of Boronolide and its fragments to the two targets

A similar procedure was followed for boronolide. Table 3.8 shows the interaction of boronolide to the two macromolecules (1w5x and 4y6r) after docking. For both systems, boronolide did not bind to the active site of the proteins, the yellow boronolide in the figures is clearly removed from the position of the crystal structure ligand that is in the active site. The binding of boronolide to both targets was also poor when viewed from the point of the calculated binding energies. Even though di-deacetylated boronolide is said to have some activity against malaria, regardless if this is the right target protein for boronolide, boronolide is certainly not active against *P*lasmodium *falciparum* DXR (4y6r).

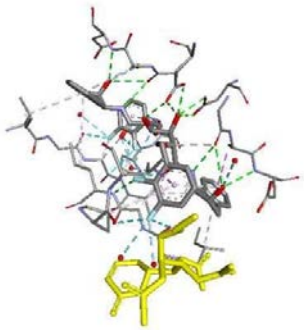
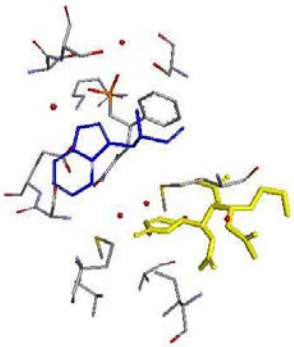
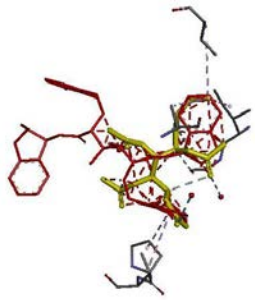
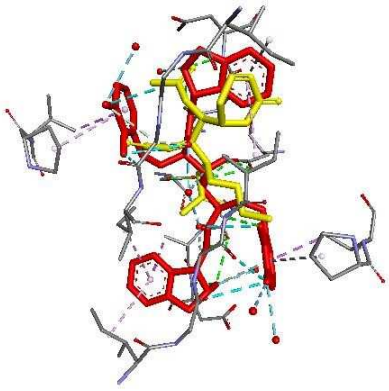
Protein	Ligand-interaction	Binding energy (Kcal)
1w5r		-4.3
4y6r		-4.8

Table 3.8: Interaction of boronolide to the two proteins with their binding energy scores

The fragments of boronolide were docked against the two target proteins as for the other two systems. Again, some of the fragments provided much better binding energies compared to the original boronolide compound. The best binding energy was obtained from fragment 83 that was docked against 4y6r, accounting for a binding energy of which was -7.1 kcal/mol (Table 3.9).

Table 3.9 also shows the binding modes for fragments in the respective active sites. The boronolide fragments in this table are coloured in yellow, compared to the crystal structure ligands as red (for
1w5x) and blue (for 4y6r). In both proteins the fragments have docked well to the active site (indicated by overlap with the position of the crystal structure ligand).

Protein	Ligand-protein interaction	Binding Energy (kcal/mol)
110_1w5x		-6.5 
45-1w5x		-6.6 

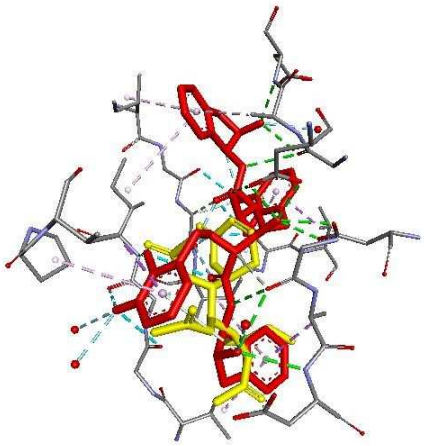
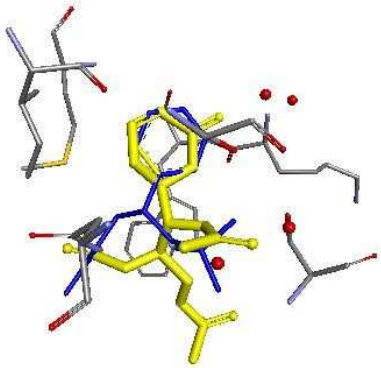
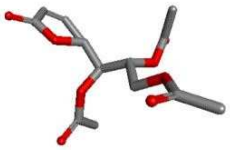
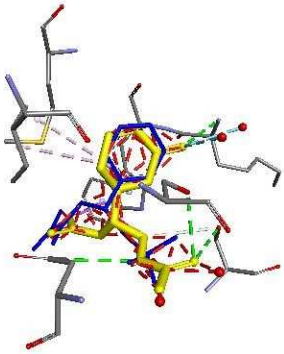
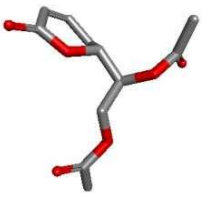
115-1w5x		 -6.7
83-4y6r		 -7.1
112-4y6r		 -7.0

Table 3.9: Interaction of the fragments of boronolide to the two proteins

3.6 Discussion

One of the challenges faced in drug discovery is the identification of new active ligands that bind to the target protein; this will often produce lead compounds that are the first step for exploration and optimization. Small molecules (fragments) are often a better choice in the search for hits, due to their favorable physical properties and the greater likelihood that they will produce an interaction in a small pocket of the target protein. Two approved drugs on the market have been discovered in this manner (Giordanetto ***et al.,*** 2019).

In the current study, fragments from boronolide, PFB-oxime and α-hispanolol gave promising results, more so than the original molecules after they were docked to two different target proteins (***Plasmodium falciparum*** DXR and HIV-1 protease). When the boronolide and α-hispanolol molecules were docked against ***Plasmodium falciparum*** DXR, both of these molecules were too large to bind to the active site. However, their fragments did bind in the active site, this could provide a starting point for drug discovery, even though we cannot conclusively say that these fragments will work well as drugs. There would still be a need to check for the drug-likeness and specificity of the fragments; and further to perform molecular dynamics of the fragments in the context of the protein to confirm binding.

Fragment screening gives higher hit rates than high-throughput screening of drug-like molecules(Chen & Shoichet, 2009). In this context, the addition of fragments from our MS prediction could extend databases such as SANCDB with new drugs predicted from fragments by allowing fragment based screening.

CONCLUSION

In this study predictive mass spectra have been obtained for 5 compounds including α-hispanalol, a PFB-oxime derivative and boronolide, and two compounds from the NIST database where EIMS are available. To obtain these mass spectra scripts have been constructed that are capable of

- taking a molecule from mol2 file format and create CP2K *ab initio* molecular dynamics input files for CP2K calculation
- taking multiple trajectories and run these using CP2K across many nodes in the context of high-performance computing (using gnu parallel)
- analysing the outputs of *ab initio* molecular dynamics, to determine where bond breakages have occurred during dynamics
- using bond breakage information to identify fragments using the RDKit FragmentOnBond, or the networkx tools to identify separate fragments and the atoms in those fragments
- calculating the charges and total spin on all fragments generated
- collating all fragments with the greatest charge across many trajectories
- computing the isotope distribution for a given fragment after determining the exact atomic composition of that fragment
- graphing the associated mass spectra from collation of fragments and isotope distributions.

As a side project, fragments have been used in a proof of concept for fragment-based drug discovery, illustrating a way to extend databases such as SANCDB.

Future work will concentrate on generating an accurate ensemble of fragments (with an appropriate energy distribution), will explore bond formation (for rearrangements) and will also include a wider range of systems from organometallic systems through to macromolecules. Exploration of fragmentation pathways through stationary states will be studied, where rates of fragmentation may be quantified using RRKM theory.

REFERENCES

Aebersold R, Mann M; **Spectrometry-based proteomics**. *Nature*, **2003,** 422(6928), 198-207.

Ahmed SM, Kruger HG, Govender T, Maguire GEM, Sayed Y, Ibrahim MAA, Naicker P, Soliman MES. **Comparison of the Molecular Dynamics and CalculatedBinding Free Energies for Nine FDA-Approved HIV-1 PRDrugs Against Subtype B and C-SA HIV PR**, *Chem Biol Drug Des***, 2013** 81:208–218

Alia A, Wawrzyniak PK, Janssen G, Buda F, Matysik J, de Groot HJM; **Differential charge polarization of axial histidines in bacterial reaction centres balances the asymmetry of the special pair**. *J. Am. Chem. Soc*., **2009,** 131, 9626–9627.

Allen F, Pon A, Greiner R, Wishart D**; Computational prediction of electron ionization mass spectra to assist in GC/MS compound identification**. *Anal. Chem*. **2016**, 88, 7689-7697.

Ásgeirsson V, Bauer CA, Grimme S**; Quantum Chemical Calculation of Electron Ionization Mass Spectra for General Organic and Inorganic Molecules**, *Chem. Sci.* **2017**

Aradi Bi, Hourahine B, Frauenheim TH. **DFTB+, a sparse matrix-based implementation of the DFTB method**, *J. Phys. Chem. A,* **2007**, 111, 5678.

Bluck L, Volmer DA; **The Role of Naturally Occurring Stable Isotopes in Mass Spectrometry, Part I**: **The Theory**, *Spectroscopy (Springer),* **2009**, 23(10), 36.

Bauer CA and Grimme S; **How to compute electron Ionization Mass Spectra from First Principles.** *J. Phys. Chem. A.* **2016**, 120, 21, 3755

Boeyens JCA; **Structure and Bonding**, **1985,** 63, 64.

Boeyens JCA, Comba P; **Molecular Mechanics: Theoretical Basis, Scope, Limits and Basic Rules, Coord**. *Chem. Rev*., **2001**, 212, 3.

Bowen JP, Allinger NL, Lipkowitz KB, Boyd DB; **Reviews in Computational Chemistry**, **1991** 2, 81 .

Brik A, Wong CH; **HIV-1 protease: mechanism and drug discovery,** *Org. Biomol.Chem.*, **2003**, 1, 5–14.

Brooijmans N, Kuntz ID, **Molecular Recognition and Docking Algorithms**, *Ann. Rev. Biophys. Biomol. Struct.*, **2003**, 32, 335–373.

Bauer CA, Grimme S, **Elucidation of Electron Ionization Induced Fragmentations of Adenine by Semiempirical and Density Functional Molecular Dynamics**, *J. Phys. Chem. A*, **2014**, 118, 11479-11484.

Chai JD, Head-Gordon M**: Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections***, Phys. Chem. Chem. Phys***., 2008**, 10, 6615-20.

Chen D, Ranganathan A, Jzerman AP, Siegal G, Carlsson J**: Complementarity between in silico and biophysical screening approaches in fragment-based lead discovery against the A2A adenosine receptor**. *J. Chem. Inf. Model*., **2013,** 53, 2701–2714.

Chen CY-C**: TCM Database@Taiwan: the world's largest traditional Chinese medicine database for drug screening in silico***, PLoS One***, 2011,** 6(1), e15939.

Chen Y, Shoichet BK; **Molecular docking and ligand specificity in fragment-based inhibitor discovery**, *Nat Chem Biol*., **2009**, 5, 358–364

Chen Y, Pohlhaus DT; **In silico docking and scoring of fragments**, *Drug Discovery Today: Technologies*, **2010**, 7(3), 1740-6749

Dagan S; Aviv Amirav: **Electron Impact Mass Spectrometry of Alkanes in Supersonic Molecular Beams. 1995**

Dreizler RM, Gross EKV; **Density functional theory**, *Springer, Berlin,* **1990**.

Dittwald P, Valkenborg D, Claesen J, Rockwood AL; **On the Fine Isotopic Distribution and Limits to Resolution in Mass Spectrometry, 2015**.

English AC, Groom CR, Hubbard RE**; Experimental and computational mapping of the binding surface of a crystalline protein.** *Protein Eng*, **2001**, 14, 47–59.

Fadee Bl, Pietroiusti A, Shvedova A; **Adverse effects of engineered nanomaterials**., **2012**, 1st Edition, Academic Press, Cambridge Massachusetts.

Foiles SM, Baskes MI, Daw MS: **Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd,Pt, and their alloys**, *Phys Rev B* **1986**, 33:7983–7991.

Gallo RC, Sarin PS, Gelmann EP, Robert-Guroff M, Richardson E, Kalyanaraman VS, Mann D, Sidhu GD, Stahl RE, Zolla-Pazner S, Leibowitch J, Popovic M; **Isolation of human T-cell leukemia virus in acquired immune deficiency syndrome (AIDS)**, *Science*, **1983**;220:865–867.

Gaalswyk, K and Rowley, CN: **An explicit-solvent conformation search method using open software.** *PeerJ* **2016** 4:2088

Ghosh AK, Osswald HL, Prato G**; Recent Progress in the Development of HIV-1 Protease Inhibitors for the Treatment of HIV/AIDS**. *J. Med. Chem.*, **2017**, 59(11),5172– 5208.

Gilles, M.; Didier, R; **Optimizing fragment and scaffold docking by use of molecular interaction fingerprints**. *J. Chem. Inf. Comput. Sci.* **2007**, 47, 195–207.

Giordanetto F, Jin C, Willmore L , Feher M , Shaw DE; **Fragment Hits: What do They Look Like and How do They Bind?***, J. Med. Chem*. **2019,** 62, 3381−3394

Giron N, Traves PG, Rodriguez B, Lopez-Fontal R, Bosca L, Hortelano S, De las Heras B; **Suppression of inflammatory responses by labdane-type diterpenoids,** *Toxicol. Appl. Pharmacol,* **2008,** 228, 179–189.

González MA: **Force fields and molecular dynamics simulations**. *Collection SFN*, **2011**, 12,169–200

GROMOS Volume 3: **Force Field and Topology Data Set**. The GROMOS Software for (Bio)MolecularSimulation, **2017**

Grunenberg J ed., Computational Spectroscopy, 2010, Wiley-VCH, Weinheim.

Gross J; Mass Spectrometry - A Textbook, **2011**, 2nd Ed, Springer-Verlag, Heidelberg.

Grimme S; **Towards First Principles Calculation of Electron Impact Mass Spectra of Molecules,** *Angew. Chem. Int. Ed*., **2013**, 52, 6306-6312

Gaussian 09, Revision E.01, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, *Gaussian, Inc., Wallingford CT,* **2016.**

Hermans J, Berendsen HJC, Vangunsteren WF, Postma JPM**; A consistent empirical potential for water-protein interactions**, *Biopolymers,* **1984**, 23(8), 1513–1518.

Hermansson M, Uphoff A, Kakela R, Somerharju P; **Automated quantative analysis of complex lipidomes by liquid chromatography/mass spectrometry**. *Anal Chem* **2005**, 77, 2166-2175

Hatherley R, Brown DK, Musyoka TM, Penkler DL, Faya N, Lobb KA, Bishop ÖT; **SANCDB: a South African natural compound database**. *J Cheminform*, **2015** 7:29.

Hobza P, Sponer J; **Structure, Energetics and Dynamics of the Nucleic Acid Base Pairs:Nonempirical Ab initio Calculations**. *Chemical Reviews*. **1999**, 99, 3247–76.

Houjou T, Yamatani K, Imagwa M, Shimizu T, Taguchi R; **A shotgun tandem mass spectrometric analysis of phospholipids with normal-phase and/or reverse-phase liquid chromatography/ electrospray ionization mass spectrometry**. *Rapid Comm Mass Spectrom*, **2005**, 19, 654-666

Hou T, Qiao X, Xu X. **Research and development of 3D molecular structure database of traditional chinese drugs**. *Coll Chem Mol Eng Univ Beijing* **2001**, 59, 1788–92.

Jalali-Heravi M, Fatemi M.H. **Simulation of mass spectra of noncyclic alkanes and alkenes using artificial neural network**, *Elsvier,* **2000**

Jomaa H, Wiesner J, Sanderbrand S, Altincicek B, Weidemeyer C, Hintz M,Turbachova I, Eberl M, Zeidler J, Lichtenthaler HK, Soldati D, Beck E; **Inhibitors of the nonmevalonate pathway of isoprenoid biosynthesis asantimalarial drugs**, *Science*, **1999**, 285, 1573– 1576.

Knyazev VD, Stein SE; **Classical Trajectories and RRKM Modelling of collisional Excitation and Dissociation of Benzylammonium and tert-Butyl Benzylammonium Ions in a Quadrupole-Hexapole-Quadrupole Tandem Mass Spectrometer***, J Am Soc Mass Spectrom,* **2010**, 21, 425–439.

Koehn FE, Carter GT; **The evolving role of natural products in drug discovery**, *Nat. Rev. Drug Discov*, **2005**, 4, 206–220.

Kohn W; *Rev. Mod. Phys***. 1999**, 71, 1253

Kohn W, Sham L*, Phys. Rev*. **1965**, 140, 1133

Kufareva I and R. Abagyan. **Methods of protein structure comparison.** *Mol Biol***. 2012**, 857, 231-257

Kuzuyama T, Shimizu T, Takahashi S, Fosmidomycin SH; **A specific inhibitor of 1deoxy-D-xylulose 5-phosphate reductoisomerase in thenonmevalonate pathway for terpenoid biosynthesis.***Tetrahedron Lett.* **1998** 39, 7913–7916

Lazaridis T, Karplus M; **Effective energy function for proteins in solution**, *Proteins* **1999**, 35(2), 133–152.

Leandro Martinez; **Automatic Identification of Mobile and Rigid Substructures in Molecular Dynamics Simulations and Fractional Structural Fluctuation Analysis**. *Plos One*. **2015**; 10(3): e0119264.

Lell B, Ruangweerayut R, Wiesner J, Missinou MA, Shindler A, Baranek T, Hintz M, Hutchinson D, Jomaa H, Kremsner PG; **Fosmidomycin, a novelchemotherapeutic agent for malaria.Antimicrob**, *Agents Chem,* **2003**, 47 735–738.

Li B , Ma C, Zhao X , Hu Z , Du T , Xu X, Wang Z, Lin J; YaTCM: **Yet another Traditional Chinese Medicine Database for Drug Discovery**. *Comput. Struct. Biotechnol. J.* **2018**,16, 600-610.

Lindsay RK, Buchanan BG, Feigenbaum EA, Lederberg J; **Applications of Artificial Intelligence for Organic Chemistry—The Den- dral Project**·, McGraw-Hill: New York, **1980.**

Lin J, Qiu XL, Qing FL; **Synthesis of gem-difluoromethylenated analogues of boronolide, 2010.**

Lobanov MY, Bogatyreva NS, Galzitskaya OV; **Radius of Gyration as an Indicator of Protein Structure Compactness.** *Molecular Biology*, **2008**,42(4) 623–628.

Loving K, Salam NK, Sherman W; **Energetic analysis of fragment docking and application to structure-based pharmacophore hypothesis generation**. *J. Comput.-Aided Mol. Des.* **2009**, 23, 541–554.

Makatini MM, Petzold K, Sriharsha SN, Soliman MES, Honarparvar B, Arvidsson PI, Sayed Y, Govender P , Maguire GEM, Kruger HG, Govender T; **Pentacycloundecanebased inhibitors of wild-type C-South African HIV-protease.** *Bioorganic & Medicinal Chemistry Letters* **2011**, 21, 2274–2277

Makatini MM, Petzold K, Alves CN, Arvidsson PI, Honarparvar B, Govender P, Thavendran Govender, Kruger HG, Sayed Y, Lameira J, Maguire GEM, Soliman MES: **Synthesis, 2D-NMR and molecular modellingstudies of pentacycloundecane lactampeptidesand peptoids as potential HIV-1 wild type C-SAprotease inhibitors**, *journal of Enzyme Inhibition and Medicinal Chemistry*, **2013**, 28(1), 78–88.

Mann MGA, Mkwananzi HB, Antunes EM, Whibley CE, Hendricks DT, Bolton JJ and Beukes DR; **Halogenated Monoterpene Aldehydes from the South African Marine Alga Plocamium corallorhiza**, *j.Nat.Prod* **2007**, 70,596-599

Marx D, Hutter J; **in Modern Methods and Algo-rithms of Quantum Chemistry, ed. Grotendorst, J. (Forschun-gszentrum, Jülich, Germany) John von Neumann-Institut für Computing**, **2000**, 1, 301–449.

McCammon JA, Gelin BR, Karplus M; **Dynamics of folded proteins**. *Nature*. **1977**, 267(5612), 585–590.

Mishra BB, Tiwari VK; **Natural products: an evolving role in future drug discovery**, *Eur. J. Med. Chem*, **2011**, 46, 4769–4807

Morris G, Goodsell D, Halliday R, Huey R, Hart W, Belew R, Olson A; **Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function,** *Journal of Computational Chemistry* **1998**, 15, 19:1639–1662.

Murdachaew G, Mundy CJ, Schenter GK, Laino T, Hutter J; **Semiempirical self-consistent polarization de-scription of bulk water, the liquid-vapor interface, andcubic ice**, *JPhysChemA*, **2011**, 115:6046–6053.

Naidu S.V, Gupta P and Kumar P: **Stereoselective synthesis of (+)-boronolide**, *Elsevier*, **2005** 46 2129–2131

Nayal M, Honig B; **On the nature of cavities on protein surfaces: application to the identification of drug-binding sites,** *Proteins,* **2006**, 63, 892–906.

Newman DJ, Cragg GM; **Natural products as sources of new drugs over the 30 years from 1981 to 2010**, *J. Nat. Prod*, **2012** 75, 311–335.

Nieto-Mendoza E, Guevara-Salazar JA, Ramirez-Apan MT, Frontana-Uribe BA, Cogordan JA, Cardenas J; **Electro-oxidation of hispanolone and anti- inflammatory properties of the obtained derivatives**, *J. Org. Chem*. **2005** 70, 4538–4541.

Ntie-Kang F, Onguéné PA, Scharfe M, Owono Owono LC, Megnassan E, Mbaze LM et al: **ConMedNP: a natural product library from Central African medicinal plants for drug discovery**. *RSC Adv* **2014**, 4:409

Oliveira IS, Bonagamba TJ, Sarthour RS, Freitas JCC, de Azevedo ER. **NMR quantum information processing, 1st Ed**, **2007**.

Orozco M, Orellana L, Hospital A, et al. **Coarse-grained representation of protein flexibility. Foundations, successes, and shortcomings. In: Christov C**, *Advances in Proteins Chemistry and Structural Biology*, **2011**, 85, 183–215.

Palafox MA, Gill M, Nunez NJ, Rastogi VK, Mittal L, Sharmam R; **Quantum Chemistry**, **2005**, 103(4), 394-421).

Palafox MA; **Scaling Factors for the Prediction of Vibrational Spectra. I. Benzene Molecule.** *Quant Chem* **2000** 77: 661–684,

Paquet E, Viktor HL; Computational Methods for Ab Initio Molecular Dynamics. Advances in chemistry 2018

Parr RG, Yang W; **Density-Functional Theory of Atoms and Molecules**, *Oxford Univ. Press, New York*, **1989.**

Ponder JW, Case DA; **Force Fields for Protein Simulations**, *Adv. Protein Chem***., 2003,** 66, 27-85.

Porezag D, Frauenheim T, K¨ohler T, Seifert G,Kaschner R; **Construction of tightbinding- like poten-tials on the basis of density-functional theory: application to carbon**, *Phys Rev* **1995**, 51:12947–12957.

Radu Iftimie, Peter Minary, and Mark E; **Tuckerman; Ab initio molecular dynamics: Concepts, recent developments, and future trends**. *PNAS* **2005** 102. 6654 – 6659

Rasool Hassan BA; **Mass Spectrometry (Importance and Uses)**. *Pharmaceut Anal Acta* **2012** 3: e138. doi:10.4172/2153-2435.1000e138

Roccatano D, Barthel A, Zacharias M**. Structural flexibility of the nucleosome core particle at atomic resolution studied by molecular dynamics simulation**. *Biopolymers***, 2007**;85(5–6):407–421.

Rognan D; **The Impact of in Silico Screening in the Discovery of Novel and Safer Drug Candidates,** *Pharmacol. Ther*, **2017**, 175, 47–66.

Ruddigkeit L, van Deursen R, Blum LC, Reymond J-L; **Enumeration of 166 Billion**

Organic Small Molecules in the Chemical Universe Database GDB-17, *J. Chem. Inf. Model,* **2012**, 52, 11, 2864-2875

Rychnovsky SD; **Predicting NMR Spectra by Computational Methods: StructureRevision of Hexacyclinol**, *Organic Letters***. 2006**, 8, 132895-2898

Schlegel B, Härt Al, Dahse HM, Gollmick FA, Gräfe U, Dörfelt H, Kappes B; **Hexacyclinol, a New Antiproliferative Metabolite of Panus rudis HKI 0254.** *J. Antibiot.,* **2002**, 55 (9), 814-817.

Shen M, Tian S, Li Y, Li Q, Xu X, Wang J, et al. **Drug-likeness analysis of traditional Chinese medicines: 1. property distributions of compounds and natural compoundsfrom traditional Chinese medicines, 2012**, 1–13.

Sturniolo S, Liborio L, Jackson S**; Comparison between Density Functional Theory and Density Functional Tight Binding approaches for finding the muon stopping site in organic molecular crystals.** *J. Chem. Phys*. **2019**, 150, 154301

Stein SE, Scott DR, J. Am. Soc. Mass Spectrum. **1994**, 5, 859-866

The CP2K Developers Group. Available at: http://www.cp2k.org/(2012). (Accessed May 31, 2013).

Traves PG, Lopez-Fontal R, Cuadrado I, Luque A, Bosca L, de las Heras B, Hortelano S; **Critical role of the death receptor pathway in the antitumoral effects induced by hispanolone derivatives***, Oncogene,* **2013** 32, 259–268.

Treier M, Pignedoli CA, Laino T, Rieger R, MuellenK, Passerone D, Fasel R. **Surfaceassisted cyclodehy-drogenation provides a synthetic route towards eas-ily processable and chemically tailored nanographenes,** *Nature Chem,* **2011**, 3:61–67.

Tuckerman, ME. *J. Phys. B Condens. Matter* **2002**, 14, 1297–1355.

Umeda T, Tanaka N, Kusakabe Y, Nakanishi M, Kitade Y, Nakamura KT; **Molecular basis of fosmidomycin's actionon the human malaria parasitePlasmodium falciparum**. *Gifu* **2011**, 501-119.

Vanommeslaeghe K, Guvench O, MacKerell AD**; Molecular Mechanics**; *Curr Pharm Des.* **2014**, 20, 3281–3292.

Vass ME´ va A gai-Csongor, Horti F, and Keseru GM; **Multiple Fragment Docking and Linking in Primary and Secondary Pockets of Dopamine Receptors.** *ACS Med. Chem. Lett*. **2014**, 5, 1010−1014

Vlachakis D, Bencurova E, Papangelopoulos N, Kossida S; **Chapter Seven - Current State-of-the-Art Molecular Dynamics Methods and Applications**. *Advances in Protein Chemistry and Structural Biology*, **2014,** 94, 269-313.

Wang Y, Yang F, Wu P, Bu D, Sun S**; OpenMS-Simulator:an open-source software for theoretical tandem mass spectrum prediction**. *BMC Bioinformatics*. **2015**. 16:110

Warshel A, Levitt M**; Theoretical studies of enzymic reactions–dielectric, electrostatic and steric stabilization of carbonium-ion in reaction of lysozyme.** *J Mol Biol*. **1976** ;103(2):227–249

Wawrzyniak PK, Alia A, Schaap RG, Heemskerk MM, de Groot HJM, Buda F; **Protein-induced geometric constraints and charge transfer in bacteriochlorophyllhistidine complexes in LH2**. *Phys.Chem*, **2008** 10:6971–6978

Xian Zeng, Peng Zhang, Weidong He, Chu Qin , Shangying Chen, Lin Tao,Yali Wang Ying Tan, Dan Gao, Bohua Wang, Zhe Chen, Weiping Chen, Yu Yang Jiang and Yu Zong Chen; **NPASS: natural product activity and species source database for**

natural product research, discovery and tool development. *Nucleic Acids Research*, **2018**, 46,

Yehuda B. Band & Yshai Avishai, **Quantum mechanics with application to nanotechnology and information science, 2013**

Yetukuri L, Katajamma M, Mdina-Gomez G, Seppanen-Laakso T, Vidal-Puig A, Oresic M; **Bioinformatics strategies for lipidomics analysis: characterization of obesity related hepatic steatosis.** *BMC Systems Biology***, 2007**

APPENDIX 1

Script used to analyze the trajectories and plot the mass spectra graphs

```
#!/opt/chemistry/anaconda3/envs/rdk-env/bin/python  import
cp2k
from rdkit.Chem import AllChem
from rdkit.Chem import
Descriptors  import os  import
math  import sys  import os
import numpy  import pandas
import networkx as nx  import
matplotlib.pyplot as plt  import
re    mass_list=[]  for i in
range(0,700):

        mass_list.append(0)

mass_dictionary={'H':1,'He':3,'C':12,'O':16,'Br':79,'N':14,'F':19,'S':32}

base_values=[0,1,3,6,9,10,12,14,16,19,20,23,24,27,28,31,32,35,36,39,40,45,46,50,50,55,54,59,58,63,64
,69,70,75,74,79]

isotopes=[[0.0,0.0,0.0,0.0],[0.9998,0.0002,0.0,0.0],[0.000002,0.999998,0.0,0.0],[0.0759,0.924,0.0,0.
0],[1.0,0.0,0.0,0.0],[0.2,0.8,0.0,0.0],[0.989,0.011,0.0,0.0],[0.996,0.004,0.0,0.0],[0.9976,0.0004,0.
002,0.0],[1.0,0.0,0.0,0.0],[0.9048,0.0027,0.0925,0.0],[1.0,0.0,0.0,0.0],[0.79,0.1,0.11,0.0],[1.0,0.0
,0.0,0.0],[0.922,0.047,0.031,0.0],[1.0,0.0,0.0,0.0],[0.9499,0.0075,0.0425,0.0],[0.76,0.0,0.240,0.0],
[0.00334,0.0,0.00063,0.0],[0.93258,0.00012,0.000673,0.0],[0.96941,0.0,0.0,0.0],[1.0,0.0,0.0,0.0],[0.
0,0.0,0.0,0.0],[0.0025,0.9975,0.0,0.0],[0.0435,0.0,0.83789,0.09501],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0
.0],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],[0.6917,0.0,0.3083,0.0],[0.0,0.0,0.0,0.0],[0.6011,0.0,0.3989
,0.0],[0.0,0.0,0.0,0.0],[1.0,0.0,0.0,0.0],[0.0,0.0,0.0,0.0],[0.51,0.49,0.0,0.0]]



def expandlistofvalues(mass_distribution,atom):
     #NB start maybe with [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] (100% at mass 0)
     #now we add an atom to it for example a 12C
     #print("--------expandlistofvalues----------------------------------------------")

     change=[0.0]       for i in
range (0,10000):
change.append(0.0)
     #we have created a change matrix to determine the next stage of the mass spectrum

     for j in range (0,10000):
             if mass_distribution[j]>0.0:
                     #only four possible isotopes for each element, so we create four new peaks from
the existing peak
                     #every time we add an atom to the molecular formula

                     for k in range (0,4):
base=j+base_values[atom]+k#12+0,12+1,12+2,12+3

                             change[base]=change[base]+mass_distribution[j]*isotopes[atom][k]
change[j]=change[j]-mass_distribution[j]
                     #this last line is to reduce the original peak since it now creates the four
new peaks.

     return change
```

```
  def
createdistributionforformula(atomnumbers):

 #need an array of numbers of atom by element e.g. C6H10O would be   # 0  1
2  3  4  5  6  7  8  9  10 ...
        #    H  He Li Be B  C  N  O  F  N ...
        #[0,10, 0, 0, 0, 0, 6, 0, 1, 0, 0 ...]#####
        ##initialize mass_distribution array
        ##initalize also change array

        change=[]
        mass_distribution=[1.0]
for i in range (0,10000):
                mass_distribution.append(0.0)
        for atomic_number in range
(0,36):
        number_of_atoms=atomnumbers[atomic_number]       for i
in range(0,number_of_atoms):
################################################################
change=expandlistofvalues(mass_distribution,atomic_number)#6 is carbon             for k
in range (0,10000):
                    mass_distribution[k]=mass_distribution[k]+change[k]
return mass_distribution

#at this point we have sys.argv as a list of all arguments
#first member of the list is the name of this file analysis.py
#second member sys.argv[1] is what you type after analysis.py
count = 1

#sanccompound = AllChem.MolFromPDBFile('alpha_hisp.pdb',False,False)

debug = 1  def atom_pairs(sanccompound):

        #print("C - sorting atom pairs")
        #get list of rdkit bonds


        rdk_bonds=sanccompound.GetBonds()

        #this is our list of atom pairs as bond_list

        bond_list=[]
        bond_broken=[]


        #this is our counter (matching bond_list) for the number of times the bond is broken in
the trajectory        for rdk_bond in rdk_bonds:

              start_atom_idx=rdk_bond.GetBeginAtomIdx()#extract first atom index from this bond
            start_atom=sanccompound.GetAtomWithIdx(start_atom_idx)
start_atom_symbol=start_atom.GetSymbol()

             end_atom_idx = rdk_bond.GetEndAtomIdx()#extract last atom index from this bond
            end_atom =sanccompound.GetAtomWithIdx(end_atom_idx)
end_atom_symbol = end_atom.GetSymbol()
              #now we have a pair of atoms start_atom_idx, end_atom_idx)
atom_pair=[start_atom_idx,end_atom_idx]          bond_list.append(atom_pair)#add a pair of
start/end atoms to list [7,12]              bond_broken.append(0)         #for each bond we
have a table counting breakages. initialize to 0 here
        return bond_list,bond_broken


    #read line by line, so must start with one line so while works.
```

```
def bond_distance(filename,bond_list,bond_broken):
traj=open(filename)    line=traj.readline()
       #print("E - bond_distance have read first line",line)
while line:

line.strip("\n")


     numberofatoms=int(line)


     comment=traj.readline()
     comment.strip("\n")

atom_coordlist=[]
            for i in range(0,numberofatoms):
#print("reading atom number ",i)
atomline=traj.readline()
atomline.strip(" \n")
               atom_coordlist.append(atomline)


               #now we have a list of lines we check in this frame for
breakages              counter=0# to keep track of the bond number       for
bond in bond_list:


               #at this point must extract x1 y1 z1 from string atom_coordlist[bond[0]]
atom_coords = atom_coordlist[bond[0]]


               x1 = atom_coords[12:24]
y1 = atom_coords[31:44]              z1 =
atom_coords[50:64]


               #at this point must extract x2 y2 z2 from string atom_coordlist[bond[1]]
atom_coords2 =atom_coordlist[bond[1]]


               x2 = atom_coords2[12:24]
y2 = atom_coords2[31:44]             z2 =
atom_coords2[50:64]


               #need to calculate distance
using                  dx = float(x1)-float(x2)
dy = float(y1)-float(y2)             dz =
float(z1)-float(z2)           #print(x2,y2,z2)
               distance = math.sqrt((dx)**2 + (dy)**2 + (dz)**2 )


               BOND_BREAK=3
                              if
(distance>BOND_BREAK):

            bond_broken[counter]=bond_broken[counter]+1



        counter=counter+1#still to keep track of the bond number
line=traj.readline()
       return bond_broken
  def
fragment_generated(bond_broken,bond_list,sanccompound,docking_count):
```

```
        new_fragment_list=[]
counter=0       bond_broken_list=[]
for no_bonds in bond_broken:
     #print("in loop no_bonds is ",no_bonds)
if(no_bonds >1):


                bond_broken_list.append(counter)
counter=counter+1


        print(bond_broken_list)


     #so that we can go back to our rdkit molecule sanccompound
     #print(AllChem.MolToSmiles(sanccompound))


##############################################################################################
####### What about molecules for docking run for 100 trajectories make sure a docking folder exists
broken_molecule=""     molecules_for_docking=""      if  bond_broken_list:
     broken_molecule=AllChem.FragmentOnBonds(sanccompound,bond_broken_list,False)
        #molecules_for_docking=AllChem.FragmentOnBonds(sanccompound,bond_broken_list,True,dummyLabel
s=[(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1)])
molecules_for_docking=AllChem.FragmentOnBonds(sanccompound,bond_broken_list,True)         else:
     broken_molecule=sanccompound
molecules_for_docking=sanccompound
        fragment_smiles=AllChem.MolToSmiles(broken_molecule)
docking_smiles=AllChem.MolToSmiles(molecules_for_docking)
docking_smiles=re.sub("\*","H",docking_smiles)
        #print("docking smiles ", docking_smiles)
        list_for_docking=docking_smiles.split(".")
for docking_molecule in list_for_docking:
print("docking molecule : ",docking_molecule)
mol=AllChem.MolFromSmiles(docking_molecule)
     #molH=AlChem.AddHs(mol)
     AllChem.EmbedMolecule(mol)
AllChem.UFFOptimizeMolecule(mol)
     moleculename="docking/molecule_"+str(docking_count)+".pdb"
docking_count=docking_count+1
writer=AllChem.PDBWriter(moleculename)       writer.write(mol)

list_of_fragments=fragment_smiles.split(".")        for
fragment in list_of_fragments:
new_fragment_list.append(fragment)
rdk_fragment=AllChem.MolFromSmiles(fragment,False)
m_mass=Descriptors.ExactMolWt(rdk_fragment,True)
h_noh=fragment.count('H')       h_nol=fragment.count('h')
h_no=h_nol+h_noh;       h_mass=h_no*1.00784
tot_mass=m_mass+h_mass       approx_mass=int(tot_mass)
     #mass_list[approx_mass]=mass_list[approx_mass]+1
        #print(new_fragment_list)
        return new_fragment_list,bond_broken_list,docking_count
  def
analyze_charges(components,filename):
        filename=filename.replace("-pos-1.xyz",".log")
filename=filename.replace("Traj_16_","alpha_hisp_00")
print("about to anlayze component charges")
print("for filename : ",filename)     for component in
components:
     print("   component :",component)
        #analyze_charges(components)
        charges=[]     spins=[]
atomnumber=0
framenumber=0
reading_mulliken=0     if
os.path.isfile(filename):
```

```
logfile=open(filename,"r")
line=logfile.readline()
while(line):
              #stuff                #more things
if "MULLIKEN" in line:               reading_mulliken=1
charges=[]               spins=[]
atomnumber=0                    if "Atom" in line and
reading_mulliken==1:
            reading_mulliken=2
line=logfile.readline()              if "Total" in line
and reading_mulliken==2:
            reading_mulliken=0
                #print("--------------------------------------------------------------------------")
framenumber=framenumber+1
            #print(atomnumber)
            #print(charges)
            #print(spins)
fragmentnumber=0
allcharges=[]               allspins=[]
for fragment in components:
                     #print("component number ",fragmentnumber,fragment)
                     #print("spins ",spins)
       #print("charges ",charges)
totalcharge=0.0
totalspin=0.0                        for atom in
fragment:
                  totalcharge=totalcharge+float(charges[atom])
totalspin=totalspin+float(spins[atom])
                     #print("charge is
",totalcharge)                       #print("spin is
", totalspin)
allcharges.append(totalcharge)
allspins.append(totalspin)
fragmentnumber=fragmentnumber+1
chargestring=""               for charge in
allcharges:
                     mystring="{0:8.4f}".format(charge)
chargestring=chargestring+" "+mystring

#print(chargestring)
spinstring=""               for spin
in allspins:
                     mystring="{0:8.4f}".format(spin)
spinstring=spinstring+" "+mystring
                     #print(spinstring)
            print("Frame :",framenumber,"c",chargestring,"s",spinstring)
            #print("--------------------------------------------------------------------------")
if reading_mulliken==2:               words=line.split()
charges.append(words[4])              spins.append(words[5])
atomnumber=atomnumber+1


# # Atom  Element  Kind  Atomic population (alpha,beta)   Net charge  Spin moment
#       1             1              2.000090     1.974676     0.025233  0.025414
#       2             1             2.129853        2.081305    -0.211158  0.048548
#       3             1             2.107931        2.100232    -0.208163  0.007699
#       4             1             1.968239        1.962520       0.069241         0.005719
#       5             1             2.072211        2.056698    -0.128908  0.015513


line=logfile.readline()
logfile.close()       return
allcharges
```

```
    def
create_map(bond_list_l,bonds_broken_l,filename):


        molecule_map=nx.Graph()


        bond_number=0         for bond in
bond_list_l:       if not (bond_number in
bonds_broken_l):
                #print("adding bond ",bond," bond number ",bond_number," to our
map")           molecule_map.add_edge(bond[0],bond[1])       else:
                print("not adding bond ",bond," bond number ",bond_number," to our
map")       bond_number=bond_number+1        for broken in bonds_broken_l:
     print("broken ",broken)
        #plt.subplot(121)
        #nx.draw(molecule_map, with_labels=True,font_weight='bold')
        #plt.show()
        components=list(nx.connected_components(molecule_map))


#########################################################################################################
#########
#########################################################################################################
#########
### We have map of the molecule and the pieces
### What we must do is find the charge on each component at each frame of the trajectory
### what about calculating the mass of each fragment here?         for component in
components:
     print(component)
     #########total_charge=0
     ############read through one frame in log file file to extract charges
     ############print the charge
     ############repeat for the next frame
     ############if atom in log file is in component we add the charge
#for atom_number in component:
     #   print(atom_number)


    #plt.subplot(122)
        #nx.draw_shell(molecule_map,nlist=[range(5, 10), range(5)], with_labels=True,
font_weight='bold')
        charges=analyze_charges(components,filename)
        return components,charges  def
greatest_charge(charges):
current_number=0       maximum_number=-1
maximum_charge=-10000         for charge in
charges:       if(charge > maximum_charge):
charge=maximum_charge
maximum_number=current_number
current_number=current_number+1       return
maximum_charge,maximum_number  def main():
        docking_count=0
isotope_list = []
        #print("B1 - now in the main procedure")     sanccompound =
AllChem.MolFromMol2File('alpha_hisp.mol2',False,False)
        #-------------------------------------------------------------------------
perfect        current_dir = os.getcwd()     filelist = os.listdir(current_dir)
        #loop through all xyz files found in current folder
        #all trajectories
        #########good point to initialize lists that will be
used    fragment_list=[]       all_components=[]
        ############################################################
for files in filelist:
     checking_file = os.path.splitext(files)
     #if((checking_file[1] == ".xyz")and("000" in files)):
if((checking_file[1] == ".xyz")):            filename=
```

```
"".join(checking_file)
print("filename",filename)
                #need to create a list of bonds (bond_list) which are pairs of atom numbers)
#also need to initialize the array counting when a bond is broken (>3A)
bond_list,bond_broken = atom_pairs(sanccompound)
                #now we can actually check the trajectory for bond breakages
bond_broken = bond_distance(filename,bond_list,bond_broken)
                #print("bond_broken",bond_broken)
                #fragment_generated uses FragmentByBond, the smiles is easily split into fragments
#which are returned as new_fragment_list and appended to the growing list of all fragments
                #from all trajectories
                #fragment_generated also increases the appropriate point in the global mass_list
# e.g. if fragment mass 238 is found then mass_list[238] increases by 1
new_fragment_list,bond_broken_list,docking_count =
fragment_generated(bond_broken,bond_list,sanccompound,docking_count)
                #this fragment list is from the smiles - we do not really want to use this
going forward                fragment_list.append(new_fragment_list)             #new
procedure to try to map out fragments
components,charges=create_map(bond_list,bond_broken_list,filename)
component_charge,component_number=greatest_charge(charges);
                ####dont go "for component in components: all_components.append(component)
####rather all_components.append(components[component_number]) --- only add ONE component for each
trajectory
                all_components.append(components[component_number])
                #b=bond_distance(traj,line)
                #c=atom_pairs(sanccompound)
                ######################SUMMARY OF ALL THE INFORMATION AT HAND
print("--------------------------------------------------------------------------------")
print("--------------------------------------------------------------------------------")
print("--------------------------------------------------------------------------------")
                #bond list is for one trajectory
                print("Bond list:------------------------------------------------------------------")
                #print(bond_list)
                #bond broken is for one trajectory
                print("Bond broken:----------------------------------------------------------------")
                #print(bond_broken)
                #new fragment is for one trajectory
                print("New fragment list:----------------------------------------------------------")
                #print(new_fragment_list)
#bond broken list is for one trajectory
                print("Bond broken list:-----------------------------------------------------------")
            #print(bond_broken_list)
                #docking count is for ALL trajectories (save them as we go to docking folder)
print("Docking count:--------------------------------------------------------------")
                #print(docking_count)
                #fragment list is for ALL trajectories
                #print("Fragment list:--------------------------------------------------------------")
#print(fragment_list)
          #components is for one trajectory ***************we need all components from all
trajectories
                #print("Components:-----------------------------------------------------------------
")             for component in components:              print(component)
                #charges is for one trajectory *************we need charges for all trajectories
                #print("Charges:--------------------------------------------------------------------")
                #print(charges)
                #print("--------------------------------------------------------------------------------")
                #print("--------------------------------------------------------------------------------")
                #print("--------------------------------------------------------------------------------")
          #print("all_components: ", all_components)
```

```
        #mass_list is the mass spectrum now
        print(mass_list)
        print("================================================================================
=======")
    #for fragment in fragment_list:
        #   print(fragment)
isotope_list=[]###initialize 10000 elements to zero
for p in range(0,10000):       isotope_list.append(0.0)
        for component in
all_components:
     print(component)
mass_of_component=0
     iso =0
         #need an array of numbers of atom by element e.g. C6H10O would be
# 0  1  2  3  4  5  6  7  8  9  10 ...
               #    H  He Li Be B  C  N  O  F  N ...
               #[0,10, 0, 0, 0, 0, 6, 0, 1, 0, 0 ...]#####
               ##initialize mass_distribution
array                  ##initalize also change
array       atomarray=[]       for i in
range(0,92):
               atomarray.append(0)
for atom_number in component:
               atom=sanccompound.GetAtomWithIdx(atom_number)
atom_symbol=atom.GetSymbol()
#print(atom_number,atom_symbol)
atomic_number=atom.GetAtomicNum()
               #print("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
               #print("Have an atom of atomic number ",atomic_number)
atomarray[atomic_number]=atomarray[atomic_number]+1
               mass_of_component=mass_of_component+mass_dictionary[atom_symbol]
     ###MASS_LIST IS THE MONOISOTOPIC MASS SPECTRUM
mass_list[mass_of_component]=mass_list[mass_of_component]+1
     ##have a look at the atomarray
     ##NB THIS IS IMPORTANT
          #print("atom array from which we can get isotope distribution")
     #print(atomarray)
component_isotopes=createdistributionforformula(atomarray)
     ###add all of the component_isotopes to total_isotopes
     #print("the mass spectrum if only this component is ever produced, together with isotopic
abundance")
     #for i in range (0,400):
     #   print(i,component_isotopes[i])
     #print("component isotope", component_isotopes)


     for x in range(0,10000):
               isotope_list[x]=component_isotopes[x]+isotope_list[x]
#l = [x + y for x, y in zip(isotope_list, component_isotopes)]


     #print("isotope_list",isotope_list)
     #l = [x + y for x, y in zip(isotope_list, component_isotopes)]
     #total_isotopes.append(l)
               for i in range
(0,400):
print(i,isotope_list[i])



     #print("*****************************************")
     #print("LLLLLL",l)


     #print("isotope list",isotope_list)
```

```
    #print(" printing total isotopes", total_isotopes)
    #print("**************************************************************************")
    ###THIS COMPONENT ISOTOPES SPECTRUM MUST BE ADDED TO THE FINAL SPECTRUM ISOTOPE_LIST
    #print("isotope_listttttttttttttttttt", isotope_list)
    #print("monoisotopic mass ",mass_of_component)
    #print("**************************************************************************")
    #print("mass_list", mass_list)
    #############can we use these 10 monoisotopic masses to create another mass spectrum for
comparison
       print("=============================================================================================
=======")
       user = input("Please  enter the file name of yor mass spectra coordinates: ")
file_name = open(user, 'w')
 file_name.write("@    title  \"Mass Spectrum without isotope distribution\n@    xaxis  label \"Mass
Number\n@ xaxis  label \"Mass Number\"\n@     yaxis  label \"Number of Fragments\"\n@s0 line type
0\n@TYPE bar\n")


       for i in range
(0,700):
    #file_name.write("{} {}\n".format(i," ",mass_list[i]))
file_name.write(str(i)+" "+str(mass_list[i])+"\n")
    #file_name.write(i)


 #print("FRAG,",fragment,",",rdk_fragment.GetNumAtoms(),",",m_mass,",",h_no,",",approx_mass)


  if(__name__ ==
"__main__"):
       print("A - entry to system")
main()
```

APPENDIX 2

Example of CP2K input file for α-Hispanalol, including the requirement to print out Mulliken population analysis

```
&GLOBAL
  PROJECT Traj_16_000
  RUN_TYPE MD
  PRINT_LEVEL LOW
&END GLOBAL
&FORCE_EVAL
   METHOD Quickstep
   &DFT
        CHARGE 1
       LSD
        &PRINT
          &MULLIKEN
        &END
        &END
        &QS
          METHOD DFTB
```

```
      &SE
        &COULOMB
         CUTOFF [angstrom] 10.0
       &END
        &EXCHANGE
         CUTOFF [angstrom] 10.0
       &END
       &END
       &DFTB
        &PARAMETER
              PARAM_FILE_NAME list.txt
           PARAM_FILE_PATH /home/yolanda/dftb
        &END PARAMETER
       &END DFTB
       &END QS
  &SCF
        SCF_GUESS ATOMIC
      EPS_SCF 1.0E-6
       MAX_SCF 50
      &OT
       MINIMIZER DIIS
            PRECONDITIONER FULL_SINGLE_INVERSE
      &END
     &PRINT
        &RESTART OFF
      &END
         &RESTART_HISTORY OFF
      &END
      &END
       &OUTER_SCF
      EPS_SCF 1.0E-6
       MAX_SCF 10
        &END OUTER_SCF
       &END SCF
  &END DFT
 &SUBSYS
       &CELL
      ABC 40 40 40
         ALPHA_BETA_GAMMA 90 90 90
       &END CELL
 &VELOCITY
 7.761982492584113e-05 8.660639686177165e-05 1.6860179933866703e-05
 4.8288680141529064e-05 6.927193434400183e-05 3.0279680227063945e-05
 9.27500117133566e-05 5.788052357957972e-05 9.586718407210437e-05
 7.194421203371593e-05 2.183801166214565e-05 8.471926495721098e-05
 3.666743082024399e-05 5.079009939641208e-05 3.591902384819931e-05
 2.48225984597239e-05 9.902632845463011e-05 3.9859124597861866e-05
 5.849958633504147e-05 8.826191545344437e-05 8.648113495166638e-05
 1.8250048730680623e-05 1.1775949546984566e-05 5.1070108164916955e-05
 2.073636423127O1e-05 7.934739276642752e-05 3.185343528067014e-05
 2.350329984017181e-05 6.265488262172568e-05 1.748821012990477e-05
 5.0121707738844435e-05 2.9929022579622102e-05 9.090236486752231e-05
 1.748648766011902e-05 6.13958349911761e-05 6.84231294840656e-05
 9.49732022726942e-05 2.7530534175443078e-05 5.2555200490087796e-05  4.4286303462828e-05
2.4046439922030173e-05 2.3306310811401854e-05
 5.3687863815403595e-05 8.501486110800376e-05 6.273447431405177e-05
 6.827659621139426e-05 3.273096209541936e-05 6.829538718735711e-05
 7.561259901094109e-05 5.3294435425995S7e-05 5.440075607181885e-05
```

```
 9.332757754106008e-05 6.261519341460594e-05 4.5183362593634226e-05
 9.995366691333372e-05 9.477149297010936e-05 7.290861136115812e-05
 2.3665000625966162e-05 1.758676998619819e-05 3.2272428336018976e-05
 8.167213207730127e-05 6.85104974519296e-05 1.3851411911879597e-05
 8.711264055637135e-05 1.6287987098358437e-06 2.9464658025125346e-05  5.518345685039289e-
05 2.970164711865491e-05 6.011064521035553e-05
 2.3849465604060327e-05 2.3541876199914124e-05 9.351482203612626e-05
 1.9501573115801753e-05 3.853019439831125e-05 8.799189735979096e-05
 1.16405603938756e-05 8.985976635008077e-05 4.0716754653826227e-05
 8.50089600634386e-05 9.112088423920806e-06 4.6211801155377365e-05
 7.291160263140458e-05 4.9597789964392714e-05 4.3164109063510083e-05
 2.3790045881687295e-05 8.940437474776125e-05 7.005293101836894e-05
 9.02153673987747e-05 5.231141125825932e-05 6.957026718089388e-05
 8.187512185845764e-05 9.076929993826307e-05 4.894311668807098e-05
 8.369737986463405e-05 8.838670086731323e-05 5.0970249869458344e-05
 8.302225902171631e-05 4.5162044516763645e-06 5.041142532630327e-07
 4.720232555372154e-05 6.784067520710893e-05 3.385984345843171e-05
 9.806007466340472e-05 3.392273031748766e-05 2.758851835056804e-05
 7.111826663038684e-05 1.5934187173314164e-05 3.4723906344392156e-05
 2.3675003282216844e-05 5.759339644174786e-05 7.367638629987321e-05
 7.299722901617476e-05 4.766994594149525e-05 4.552611236870642e-05
 5.2807827422058864e-05 4.024916476510305e-05 5.258589595086815e-05
 6.09608257904684356-05 3.7889994198802226e-05 8.404381090701296e-05
 5.44965865386910856-05 9.85330196124541e-05 9.380971379039783e-06
 4.382575341382454e-05 6.60665241643934e-05 7.687026701237e-05
 9.190149559785896e-05 8.126719579039523e-05 3.883806322111282e-05
 5.9442407245163825e-05 1.0733697304589286e-05 3.8665196919574705e-06
 7.981409142863317e-06 3.2266122542139887e-05 5.039652817328667e-05
 2.693301466185374e-05 1.2625215667311474e-05 1.081856064430573e-05
 9.924874592348037e-05 9.410731337829089e-05 8.510406145428702e-05
 5.49855342843678e-05 3.797594393483841e-05 3.606257804294735e-05
 5.8233454669458885e-05 8.929274193288328e-05 5.5320486651835616e-05
 6.657202249273987e-05 2.35750035940702e-05 2.457226324502624e-05
 2.5396449726233436-05 2.0271786496158883e-05 1.7855274046297798e-05
 1.6878961384186375e-05 3.0207604642636656e-05 5.542439228078046e-05  4.256302411123807e-
05 5.39072570978222636-05 8.385734580153192e-06
 2.7491148993453752e-05 1.2508803571382955e-06 9.153034411123385e-05
 9.924735204138241e-05 6.815841234357165e-05 2.7772042698038237e-06
  &END VELOCITY
  &COORD
 O 5.32759154906 -1.66877261205 -0.804879906022
 C 5.89217168661 -1.18823751218 0.276847813048
 C 5.0803346527 -0.200134553065 0.820036352856
 C 4.09333194051 0.0104830381452 -0.101220545462
 C 4.27008014106 -0.97030808347 -1.07071111658
 C 2.94797839848 0.979026528567 -0.0763716624175
 C 1.80238528168 0.485445118197 0.739628649594
 C 0.436801340307 0.816493747258 0.281665025574
 C 0.343653310744 1.71161833951 -0.91582695343
 C -1.08314625783 1.97077760122 -1.30455191627
 C -1.91744732635 0.741354677654 -1.46642751932
 C -1.7943601623 -0.0389877001809 -0.182710596795
 C -0.385086143124 -0.462106122586 0.121474215445
 C -0.416768997661 -1.16771693838 1.45608753046
 C -1.69420191219 -0.966585684166 2.21564430521
 C -2.9328938472 -1.37082944094 1.47674588039
 C -2.81671964715 -1.10818495554 -0.0134369056984
 H -2.0311937744 0.71756286033 0.614183841162
 C -4.13128233227 -0.601884774197 -0.595650370394
 C -2.53238761059 -2.41282637867 -0.678523682724
 C 0.254156161336 -1.27708330403 -0.963699820713
```

```
 O -0.210980543935 1.51182300404 1.31981684562
 H 0.375775375993 1.68369239717 2.08466889803
 C 0.928051167104 3.06172868882 -0.521062105798
 O -1.6607161783 2.75542144773 -0.293677735061
 H -2.64506835732 2.59756538162 -0.353536258411
 H 6.69787757424 -1.72459967401 0.773198418186
 H 5.39703523514 0.526103944312 1.54135361068
 H 3.79791152938 -0.915800368923 -2.05263467235
 H 3.38876208232 1.90548482562 0.373517106321
 H 2.72737771484 1.19236208429 -1.12767999733
 H 1.9904204778 0.83509872062 1.78804749538
 H 1.92701100942 -0.626560041663 0.803750972237
 H 0.83131730665 1.29545015124 -1.82514637216
 H -1.09184736479 2.61731669288 -2.21705936045
 H -1.73064834114 0.162089001742 -2.37350528772
 H -2.98322415288 1.07973727315 -1.52451253007
 H 0.389505337499 -0.764113496844 2.12184494102
 H -0.148605782437 -2.24633749039 1.3230646536
 H -1.62973965056 -1.63557514912 3.11152187729
 H -1.72182384889 0.0609331890628 2.6146221755
 H -3.76442573874 -0.729255801347 1.86588809888
 H -3.24521435738 -2.41763859507 1.68080184788
 H -4.88413168229 -1.38852399235 -0.332640983307
 H -4.02597851635 -0.663030435388 -1.70147992164
 H -4.40196285292 0.396854653691 -0.228887656983
 H -2.19038731008 -2.33999814107 -1.72247713977
 H -3.51734335848 -2.94730516561 -0.741297643181
 H -1.90382597463 -3.10310759159 -0.0592201305856
 H 1.33542306793 -1.01953181222 -1.10190434803
 H -0.183585857497 -1.0858442046 -1.97612748338
 H 0.26691516324 -2.37391896962 -0.737891830434
 H 1.31059953982 3.07616512954 0.508668124833
 H 0.167478464708 3.87873039222 -0.571585000642
 H 1.6950523711 3.34548010066 -1.28373856458
  &END COORD   &END
SUBSYS
&END FORCE_EVAL
&MOTION
  &MD
          ENSEMBLE LANGEVIN
         TEMPERATURE 2000
        TIMESTEP 0.2
       STEPS 10000
         &LANGEVIN
        GAMMA 0.01
         &END LANGEVIN
  &END MD
&END MOTION
```

APPENDIX 3

Dockings

Example of a Vina input script for docking

```
receptor = proteins_folder/1w5x_apo.pdbqt ligand
= ligands_folder/molecule_97.pdbqt
  out =
1w5x_apomolecule_97.all.pdbqt
  log =
1w5x_apomolecule_97.log
  center_x = 12.88
center_y = 22.66
center_z = 5.62
size_x = 20 size_y
= 20 size_z = 20
energy_range = 4
exhaustiveness =
16

cpu = 1
```

All Docking Results Boronolide

Protein	Ligand	Binding Energy
1w5x	molecule_0	-6.2
1w5x	molecule_1	-6.2
1w5x	molecule_10	-6.4
1w5x	molecule_100	-6.1
1w5x	molecule_101	-6.3
1w5x	molecule_102	-6.4
1w5x	molecule_103	-2
1w5x	molecule_104	-1.2
1w5x	molecule_106	-5.6
1w5x	molecule_107	-3.5
1w5x	molecule_108	-6.4
1w5x	molecule_109	-6.2
1w5x	molecule_11	-6
1w5x	molecule_110	-6.5
1w5x	molecule_111	-6.2
1w5x	molecule_112	-6
1w5x	molecule_113	-4
1w5x	molecule_114	-6.5
1w5x	molecule_115	-6.7
1w5x	molecule_116	-6.3
1w5x	molecule_118	-1.2
1w5x	molecule_119	-1.2
1w5x	molecule_12	-6.5
1w5x	molecule_120	-5.1
1w5x	molecule_121	-3.2
1w5x	molecule_122	0
1w5x	molecule_123	0
1w5x	molecule_124	0
1w5x	molecule_125	-6.2
1w5x	molecule_126	-2
1w5x	molecule_127	-2
1w5x	molecule_128	-4.4
1w5x	molecule_129	-4.4
1w5x	molecule_13	-6.2
1w5x	molecule_130	-6.3
1w5x	molecule_131	-6.4
1w5x	molecule_132	-5.6
1w5x	molecule_133	-4.2
1w5x	molecule_134	-6.4
1w5x	molecule_135	-6.2
1w5x	molecule_136	-6.5
1w5x	molecule_137	-6.4
1w5x	molecule_138	-5.8
1w5x	molecule_139	-6.1
1w5x	molecule_14	-6.6
1w5x	molecule_140	-6.1
1w5x	molecule_141	-2
1w5x	molecule_142	-4.2
1w5x	molecule_143	-5.1
1w5x	molecule_144	-6.2
1w5x	molecule_145	-2
1w5x	molecule_146	-4.2
1w5x	molecule_147	-5.1
1w5x	molecule_148	-6.4
1w5x	molecule_149	-6.2
1w5x	molecule_15	-6.3
1w5x	molecule_150	-6.3
1w5x	molecule_151	-6.4
1w5x	molecule_152	-2
1w5x	molecule_153	-5.1
1w5x	molecule_154	-3.5

1w5x	**molecule_155**	**-6**
1w5x	**molecule_16**	**-2**
1w5x	**molecule_17**	**-2**
1w5x	**molecule_18**	**-4.4**
1w5x	**molecule_19**	**-4.7**
1w5x	**molecule_2**	**-6.2**
1w5x	**molecule_20**	**-6.3**
1w5x	**molecule_21**	**-6.7**
1w5x	**molecule_22**	**-2**
1w5x	**molecule_23**	**-4.2**
1w5x	**molecule_24**	**-5.2**
1w5x	**molecule_25**	**-5.6**
1w5x	**molecule_26**	**-3.6**
1w5x	**molecule_27**	**-2**
1w5x	**molecule_28**	**-5.1**
1w5x	**molecule_29**	**-4.6**
1w5x	**molecule_3**	**-6.1**
1w5x	**molecule_30**	**-2**
1w5x	**molecule_31**	**-1.2**
1w5x	**molecule_33**	**-5.5**
1w5x	**molecule_34**	**-3.5**
1w5x	**molecule_35**	**-2**
1w5x	**molecule_37**	**-1.2**
1w5x	**molecule_38**	**-4.3**
1w5x	**molecule_39**	**-4.5**
1w5x	**molecule_4**	**-6**
1w5x	**molecule_40**	**0**
1w5x	**molecule_41**	**0**
1w5x	**molecule_42**	**0**
1w5x	**molecule_43**	**-6.3**
1w5x	**molecule_44**	**-6.5**
1w5x	**molecule_45**	**-6.6**

1w5x	**molecule_46**	**-6.4**
1w5x	**molecule_47**	**-6.1**
1w5x	**molecule_48**	**-5.9**
1w5x	**molecule_49**	**-6.2**
1w5x	**molecule_5**	**-6.4**
1w5x	**molecule_50**	**-6.2**
1w5x	**molecule_51**	**-6.6**
1w5x	**molecule_52**	**-6.2**
1w5x	**molecule_53**	**-6.3**
1w5x	**molecule_54**	**-2**
1w5x	**molecule_55**	**-2**
1w5x	**molecule_56**	**-6.2**
1w5x	**molecule_57**	**-5.5**
1w5x	**molecule_58**	**-4.2**
1w5x	**molecule_59**	**-6.3**
1w5x	**molecule_6**	**-6.1**
1w5x	**molecule_60**	**-6.1**
1w5x	**molecule_61**	**-6.1**
1w5x	**molecule_62**	**-6.1**
1w5x	**molecule_63**	**-6.5**
1w5x	**molecule_64**	**-6.6**
1w5x	**molecule_65**	**-2**
1w5x	**molecule_66**	**-4.2**
1w5x	**molecule_67**	**-5.1**
1w5x	**molecule_68**	**-5.8**
1w5x	**molecule_69**	**-4**
1w5x	**molecule_7**	**-6.6**
1w5x	**molecule_70**	**-6.6**
1w5x	**molecule_71**	**-6.2**
1w5x	**molecule_72**	**-6.1**
1w5x	**molecule_73**	**-2**
1w5x	**molecule_74**	**-5.2**

1w5x	molecule_75	-4.4
1w5x	molecule_76	-6.6
1w5x	molecule_77	-6.2
1w5x	molecule_78	-2
1w5x	molecule_79	-5.1
1w5x	molecule_8	-6.2
1w5x	molecule_80	-3.6
1w5x	molecule_81	-5.7
1w5x	molecule_82	-4.2
1w5x	molecule_83	-6.4
1w5x	molecule_84	-3.1
1w5x	molecule_85	-2
1w5x	molecule_86	-5
1w5x	molecule_87	-3.6
1w5x	molecule_88	-6.1
1w5x	molecule_89	-6.1
1w5x	molecule_9	-6.3
1w5x	molecule_90	-6.6
1w5x	molecule_91	-6.4
1w5x	molecule_92	-5.5
1w5x	molecule_93	-4.2
1w5x	molecule_94	-6.5
1w5x	molecule_95	-6.2
1w5x	molecule_96	-5.8
1w5x	molecule_97	-6.5
1w5x	molecule_98	-6.6
1w5x	molecule_99	-6.5
4y6r	molecule_0	-5.4
4y6r	molecule_1	-4.6
4y6r	molecule_10	-6
4y6r	molecule_100	-5.8
4y6r	molecule_101	-5.8
4y6r	molecule_102	-5.1
4y6r	molecule_103	-2.3
4y6r	molecule_104	-1.5
4y6r	molecule_106	-6.4
4y6r	molecule_107	-3.4
4y6r	molecule_108	-4.8
4y6r	molecule_109	-6.2
4y6r	molecule_11	-5.2
4y6r	molecule_110	-6.2
4y6r	molecule_111	-6
4y6r	molecule_112	-7
4y6r	molecule_113	-4.5
4y6r	molecule_114	-5.6
4y6r	molecule_115	-5.7
4y6r	molecule_116	-5.6
4y6r	molecule_118	-1.5
4y6r	molecule_119	-1.5
4y6r	molecule_12	-5.5
4y6r	molecule_120	-5.9
4y6r	molecule_121	-3.2
4y6r	molecule_122	0
4y6r	molecule_123	0
4y6r	molecule_124	0
4y6r	molecule_125	-6
4y6r	molecule_126	-2.3
4y6r	molecule_127	-2.3
4y6r	molecule_128	-5
4y6r	molecule_129	-4.6
4y6r	molecule_13	-6
4y6r	molecule_130	-5.4
4y6r	molecule_131	-5.3
4y6r	molecule_132	-6.4

4y6r	**molecule_133**	**-4.3**
4y6r	**molecule_134**	**-6.1**
4y6r	**molecule_135**	**-5.9**
4y6r	**molecule_136**	**-4.6**
4y6r	**molecule_137**	**-5**
4y6r	**molecule_138**	**-5.6**
4y6r	**molecule_139**	**-6**
4y6r	**molecule_14**	**-4.1**
4y6r	**molecule_140**	**-5.8**
4y6r	**molecule_141**	**-2.3**
4y6r	**molecule_142**	**-4.3**
4y6r	**molecule_143**	**-5.7**
4y6r	**molecule_144**	**-5.4**
4y6r	**molecule_145**	**-2.3**
4y6r	**molecule_146**	**-4.3**
4y6r	**molecule_147**	**-5.6**
4y6r	**molecule_148**	**-5.7**
4y6r	**molecule_149**	**-5.5**
4y6r	**molecule_15**	**-5.8**
4y6r	**molecule_150**	**-6.4**
4y6r	**molecule_151**	**-5.3**
4y6r	**molecule_152**	**-2.3**
4y6r	**molecule_153**	**-5.7**
4y6r	**molecule_154**	**-3.7**
4y6r	**molecule_155**	**-5.2**
4y6r	**molecule_16**	**-2.3**
4y6r	**molecule_17**	**-2.3**
4y6r	**molecule_18**	**-5**
4y6r	**molecule_19**	**-5**
4y6r	**molecule_2**	**-6**
4y6r	**molecule_20**	**-5.8**
4y6r	**molecule_21**	**-5.4**

4y6r	**molecule_22**	**-2.3**
4y6r	**molecule_23**	**-4.3**
4y6r	**molecule_24**	**-5.7**
4y6r	**molecule_25**	**-6.3**
4y6r	**molecule_26**	**-3.7**
4y6r	**molecule_27**	**-2.3**
4y6r	**molecule_28**	**-5.9**
4y6r	**molecule_29**	**-4.9**
4y6r	**molecule_3**	**-5.2**
4y6r	**molecule_30**	**-2.3**
4y6r	**molecule_31**	**-1.5**
4y6r	**molecule_33**	**-6.5**
4y6r	**molecule_34**	**-3.4**
4y6r	**molecule_35**	**-2.3**
4y6r	**molecule_37**	**-1.5**
4y6r	**molecule_38**	**-5.1**
4y6r	**molecule_39**	**-4.9**
4y6r	**molecule_4**	**-5.2**
4y6r	**molecule_40**	**0**
4y6r	**molecule_41**	**0**
4y6r	**molecule_42**	**0**
4y6r	**molecule_43**	**-5.4**
4y6r	**molecule_44**	**-5.3**
4y6r	**molecule_45**	**-4.3**
4y6r	**molecule_46**	**-4.8**
4y6r	**molecule_47**	**-4.3**
4y6r	**molecule_48**	**-5.3**
4y6r	**molecule_49**	**-5.5**
4y6r	**molecule_5**	**-6.1**
4y6r	**molecule_50**	**-5.4**
4y6r	**molecule_51**	**-6**
4y6r	**molecule_52**	**-6**

4y6r	**molecule_53**	**-5.1**
4y6r	**molecule_54**	**-2.3**
4y6r	**molecule_55**	**-2.3**
4y6r	**molecule_56**	**-6.7**
4y6r	**molecule_57**	**-6.4**
4y6r	**molecule_58**	**-4.3**
4y6r	**molecule_59**	**-5.9**
4y6r	**molecule_6**	**-5.7**
4y6r	**molecule_60**	**-5.2**
4y6r	**molecule_61**	**-5.4**
4y6r	**molecule_62**	**-5.8**
4y6r	**molecule_63**	**-4.7**
4y6r	**molecule_64**	**-5.3**
4y6r	**molecule_65**	**-2.3**
4y6r	**molecule_66**	**-4.3**
4y6r	**molecule_67**	**-5.9**
4y6r	**molecule_68**	**-6.9**
4y6r	**molecule_69**	**-4.5**
4y6r	**molecule_7**	**-4.8**
4y6r	**molecule_70**	**-5.9**
4y6r	**molecule_71**	**-6.1**
4y6r	**molecule_72**	**-5.4**
4y6r	**molecule_73**	**-2.3**
4y6r	**molecule_74**	**-5.7**
4y6r	**molecule_75**	**-5**
4y6r	**molecule_76**	**-6.1**
4y6r	**molecule_77**	**-5.6**
4y6r	**molecule_78**	**-2.3**
4y6r	**molecule_79**	**-5.9**
4y6r	**molecule_8**	**-6.5**
4y6r	**molecule_80**	**-3.7**
4y6r	**molecule_81**	**-6.4**

4y6r	**molecule_82**	**-4.3**
4y6r	**molecule_83**	**-7.1**
4y6r	**molecule_84**	**-2.9**
4y6r	**molecule_85**	**-2.3**
4y6r	**molecule_86**	**-5.7**
4y6r	**molecule_87**	**-3.7**
4y6r	**molecule_88**	**-5.9**
y6r	**molecule_89**	**-5.4**
4y6r	**molecule_9**	**-5.6**
4y6r	**molecule_90**	**-5.9**
4y6r	**molecule_91**	**-5.4**
4y6r	**molecule_92**	**-6.3**
4y6r	**molecule_93**	**-4.3**
4y6r	**molecule_94**	**-6.2**
4y6r	**molecule_95**	**-5.5**
4y6r	**molecule_96**	**-5.8**
4y6r	**molecule_97**	**-4.1**
4y6r	**molecule_98**	**-5.5**
4y6r	**molecule_99**	**-5.5**