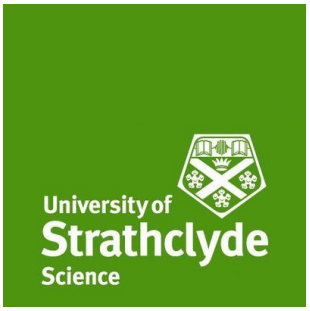

Department of Pure and Applied Chemistry

Calculation of Octanol-Water Partition Coefficents using Computational Chemistry and Cheminformatics

Completed within the SCoTCH Group

Author:
Daniel Gaimster

Supervisor:
Dr David Palmer

22nd July 2020

Declaration of Ownership

A thesis submitted to the Department of Pure and Applied Chemistry, University of Strathclyde, in part fulfilment of the regulations for the degree of Masters of Science in Applied Chemistry and Chemical Engineering.

I certify that the thesis has been written by me. Any help I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Acknowledgements

I would firstly like to thank my project supervisor David Palmer for the many long discussions we had over the life of the project and the fine attention to detail in the correction of my final draft. Secondly, all the members of the SCoTCH research group for their insight and support, especially Johnny with whom I had many extensive discussions on ML theory and Alex who gave me excellent technical advice. Finally, I would like to thank my boyfriend Felix for dealing with my project-driven anxiety, and my friends and family for the emotional support throughout the entire endeavour.

Acronyms

COSMO-RS Conductor-like Screening Model for Realistic Solvation.

DFT density functional theory.

E-state electrotopological state.

H-bond hydrogen bond.

HF Hartree-Fock.

HPLC high performance liquid chromatography.

LDA local density approximation.

logP octanol-water partition coefficient.

LSER Linear Solvation Energy Relationship.

ML machine learning.

MW molecular weight.

OECD Organisation for Economic Cooperation and Development.

QM quantum mechanical.

RFE recursive feature elimination.

RMSE root mean square error.

ro5 rule of five.

SAMPL6 Statistical Assessment of the Modelling of Proteins and Ligands.

SCF self-consistent field.

SDE standard deviation of the error.

SMD universal Solvation Model based on Density.

USD United States Dollars.

Contents

Abstract

In this thesis hydration free energy, solvation free energy, and primarily logP were calculated for a dataset of small organic molecules via quantum mechanical (QM) and machine learning (ML) methods, with the aim of finding the most desirable method. It was found for the dataset of approximately 100 molecules that ML predicted logP values with a RMSE = 0.357, whilst the best QM model had a RMSE = 0.418. Similarly, the free energy values were also best calculated through ML methods. These results contradicted literature which found that QM methods gave more accurate predictions, this is believed to be due to the small size of the molecules in this data set. ML methods have a further advantage of computing logP values with less computational expense then their QM counterparts. This work could be extended by the application of the same methods to a database of larger and more drug-like molecules, to see if the results hold up for more practical molecules with industrial applications.

Chapter 1

Introduction

1.1 Thesis Motivation

Lipophilicity and hydrophilicity are important molecular attributes that help drive the development and understanding of vital chemicals in a variety of sectors essential to society: pharmaceutical, agricultural, environmental etc. These attributes are measured through the octanol-water partition coefficient (logP), which is the ratio of the concentration of the neutral form of a solute molecule between two immiscible solvent phases, commonly octanol and water. LogP can indicate whether or not a drug molecule will be accepted in the human body, if a herbicide will be absorbed by a plant it is sprayed on, if a waste chemical dumped into a stream will be ingested by fauna.

Experimental calculation of logP is a slow process that is both tedious and hard to automate, leading to a low throughput of molecules to be tested. There is a vast search space of molecules that have the potential to be used as drugs and agrochemicals, but a limited capacity to find pathways to their creation. This leads to the need to find options to narrow this search space, if a drug molecule has to be created before its logP value can be measured, there is the possibility that it will be too high or too low to ever function as a drug. This is wasted time and resources that could have been spent on a more valuable and potentially life-saving drug. The need to computationally calculate accurate logP values is therefore of paramount importance. It allows for the logP value of molecules to be obtained far more rapidly than could ever be possible through experimental means, whilst never having to first synthesise. Even if logP can only be calculated within a certain range of the real experimental value, this allows for far fewer molecules to have to be considered. Computational calculation of logP also allows for a more green chemistry approach to measurement: the molecule does not have to be formed and octanol and water

solvents do not have to be wasted.

In the computational world, ML has taken the spotlight in recent years with large-scale companies such as Google and BP extensively utilising and developing the method. With so much buzz and its ability to be used in almost every sector, there is no surprise that ML has attracted a similar degree of attention in the computational chemistry world. With enough data supplied, theoretically any chemical property has the potential to be calculated through ML means. However, is the attention ML methods are receiving justified? Or is another more longstanding approach to logP calculation such as through QM methods more deserving of scientific attention? In this work, the relative merits of QM and ML methods of logP calculation will be investigated through the predictions of a dataset of small organic molecules. The QM and ML methods will then be compared on their accuracy, computational expense, amount of *a priori* data required, and practicality with the aim to find which method performs the best overall.

1.2 Thesis Layout

The thesis starts with a review of the current state of literature in Chapter 2. In this section the importance of logP as a metric in the pharmaceutical and environmental/agrochemical sectors is discussed, and the size and importance of these industries evaluated. The most common method of experimental measurement and its deficiencies will be discussed. An overview of other computational methods of logP calculation will be highlighted and finally the current state of QM and ML logP calculation will be evaluated.

Chapter 3 covers the underlying theory behind the methods employed in this thesis. As such, it contains a section describing quantum mechanical chemistry and how these methods are applied in density functional theory (DFT). The ML methods and concepts are then explained, finishing with a short explanation of molecular descriptors.

Chapter 4 discusses how logP calculations were computed. It contains a section on how the database of molecules to be tested was formed and from what datasets the experimental data came from. A description of the program used for the QM calculations and the levels of theory used is supplied, along with a description of how the ML algorithm was created.

The main results found from the project are presented and discussed in Chapter 5.

An evaluation of the database created in terms of molecule types, molecular weights (MWs) and logP range is performed. The effect the functional and the basis set has on QM prediction of logP is evaluated, along with an investigation on hydration and solvation energy prediction. The ML predictions accuracy with increasing complexity of the model is observed, and the accuracy of the ML results is compared to the QM results for logP, hydration energy, and solvation energy.

Chapter 6 reviews how this work is of relevance to chemical engineering. It highlights the use of solvation energy in the design of chemical processes. How solvation energy can be used in the design of absorption towers and a overview of a prominent application of absorption, CO_2 capture, is given. Finally an assessment of the potential for QM & ML applications in the prediction of properties for chemical process simulation is given.

In the final chapter, the relative merits of the QM and ML methods are evaluated and potential directions for improvement of both methods are discussed.

Chapter 2

Literature Review

2.1 The Importance of LogP

LogP is the most common method to describe the lipophilicity and hydrophobicity of an organic compound.[1,2] It represents the ratio of the concentration of the neutral form of a solute molecule between two immiscible solvent phases, as can be seen in Figure 2.1. These two solvents are almost always n-octanol and water, so much so that logP is synonymous with this particular combination.[3] LogP is an indication of a neutral compound's propensity to selectively partition between nonpolar and polar environments.[4] This property is of particular interest to pharmaceutical, environmental, and chemical industries, amongst others.[1]

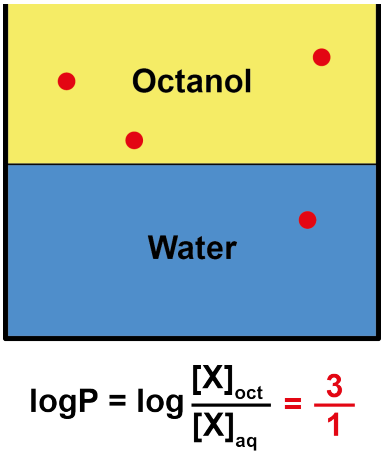

Figure 2.1: Image depicting a solute molecule that has partitioned itself between the two immiscible phases of octanol and water and its corresponding logP value.

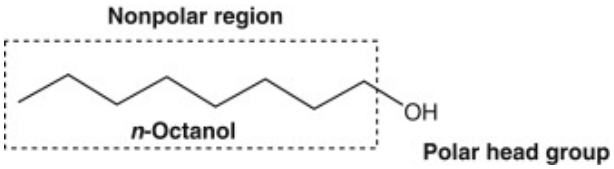

Figure 2.2: Structure of octanol, displaying areas of nonpolarity and polarity. Image taken from Rice (2014)[5]

2.1.1 Pharmaceutical Industry

In order for a drug molecule to be received by the human body, the drug must possess a balance of polar and nonpolar characteristics. The drug must have some degree of hydrophilic properties in order to traverse through the bloodstream and once the drug has reached the desired site it must have sufficient nonpolar properties in order to pass through the cell membranes. Cell membranes are comprised of bilayers of lipid molecules as described in Figure 2.3, and if we compare to octanol as shown in Figure 2.2, we can see similarities of the nonpolar carbon chain and the polar head. These similarities are a reason why the octanol-water coefficient is regarded as a good proxy for drug molecule behaviour in the human body.[1,5,6]

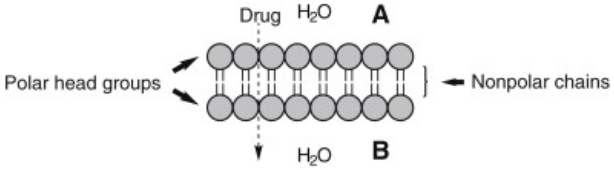

Figure 2.3: Diagram depicting the structure of a lipid bilayer in the body. At A & B, the drug molecule will come into contact with the polar heads of the bilayer, while between the heads lies an area comprised of nonpolar hydrocarbon chains. Image taken from Rice (2014).[5]

LogP is one of the primary determining factors for drug-like molecules, as famously identified by Lipinski's "rule of five (ro5)".[7] Lipinski proposed 4 rules to indicate if a drug molecule would be soluble and permeable in the human body. If a drug cannot obtain at least 3 out of 4 of these criteria, it is unlikely to be accepted orally. The "five" refers to each rule being a multiple of 5:

1. Hydrogen bond donors < 5
2. Hydrogen bond acceptors < 10
3. MW < 500
4. LogP < 5

These rules are by no means exhaustive,[8–10] however, they are applicable to the vast majority of drugs released to market.[11] The existence of the ro5 allows for screening of the vast chemical search space that comprises potential drug candidates which greatly increases the efficiency at which new drug molecules can be found. Whilst logP is only 1 factor amongst 4 others, it is the only variable that cannot be found from the visual inspection of a molecule's structure. Therefore, determination of logP is of great importance to the pharmaceutical industry.

Phase 1 trials primarily exist to ascertain the human pharmacokinetics of a drug molecule,[12] of which lipophilicity plays a major role. Pharmacokinetics/bioavailability caused 40% of failures in 1991, in 2000 this was reduced to only 10%.[13] This can be at least partially attributed to be proof of the positive effect the ro5 (published in 1997) had on the industry.[12] Estimates for the cost of discovering a new drug can range wildly from approximately $100 million United States Dollars (USD) to $2.5 billion,[14] and as can been seen in Figure 2.4, the global investment into the research and development of drugs is only expected to increase. Regardless of the exact value there can be no doubt that these are large sums of funds that can be prohibitive to the development of lifesaving medication. The pharmaceutical industry has a global annual revenue of over $1.2 trillion dollars each year.[15] *In silico* predictions of logP can allow for the search space of potential drug molecules to be reduced allowing for more efficient evaluation.

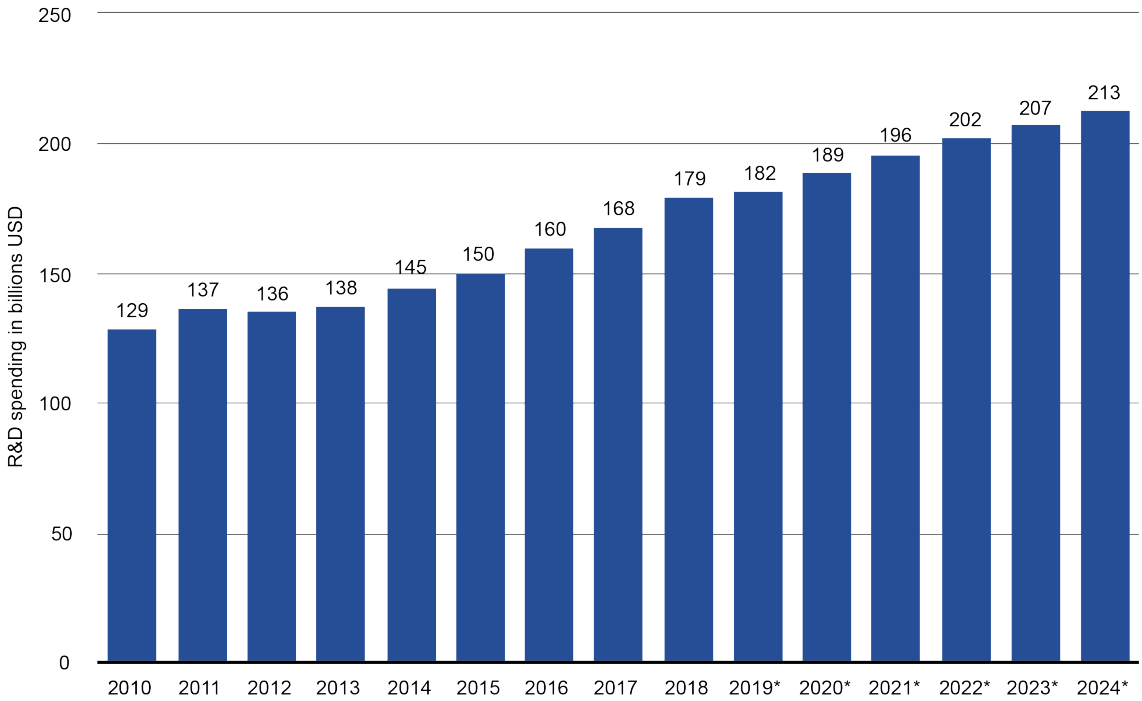

Figure 2.4: Bar graph depicting the global expenditure into the research and development of new drug molecules. *Denotes that these are prediction values. Data from.[15]

2.1.2 Environmental and Agrochemical

Solubility is a fundamental property to environmental chemistry, whenever a pollutant is released to the environment it will find its way into all areas of the ecosystem, from the oceans and soil to animals and ultimately humans.[16] Understanding the nature of a where a pollutant will spread in significant concentrations is therefore a matter of import. As discussed in Section 2.1, logP is related to solubility, therefore understanding the logP of pollutants is critical to environmental chemistry.

LogP is used as a fundamental measurement of solubility in the environmental industry: it measures the partitioning of a chemical between water and lipid material. As examined in Section 2.1.1, logP is a reasonable measurement for a drug molecule's solubility in the human body, and this is no different for a pollutant. LogP is just as applicable for the fats in animals and fish as it is for the lipids in the human body.[17] Plants also share fatty structure of animals: the cuticle of a plant is a thin top layer that is comprised of mostly lipid materials,[18] logP has been proven to be an effective measurement of a chemical's ability to penetrate through this barrier.[18,19]

The use of crop protection chemicals and fertilisers has been a tremendous benefit to modern society, they have allowed for increased food and fibre production while decreasing waste by allowing more crops to be successfully harvested, helping combat the ever growing planet-wide need for food resources. However, agrochemicals can also have an adverse effect on the environment, flora, and fauna,[20] therefore understanding how they enter the environment has been extensively investigated. LogP has long been recognised as the most important metric in determining translocation in plants and sorption to soils.[21] Concerning tracking the movement of pesticides in the environment, logP is linked to a molecule's water solubility,[22] soil/sediment adsorption coefficients[23] and bioconcentration factors for aquatic organisms.[24] Therefore, easy and reliable measurement of logP is of great benefit to the agrochemical industry.

The prevalence of the ro5 in the pharmaceutical industry has been discussed in Section 2.1.1, and Lipinski's paper[7] had similarly defining influence on the agrochemical industry. Following on from Lipinski's work, Briggs gave a talk outlining his "ground rules of three" concerning physical property properties of fungicides.[25] However, the most influential work stemming from ro5 in this sector was published by Tice.[26] Tice's work was an evaluation of how the ro5 could be applied to the agrochemical industry, determining that for herbicides and fungicides Lipinski's rules generally

applied, with the exception of the rule of hydrogen-bond donors. This work was expanded to include insecticides and agrochemicals in general by Clarke and Delaney.[27] Upper limits on pesticide-likeness properties, akin to the ro5, were developed in further work.[28,29] These properties included the 4 original ro5 parameters, with the addition of rotational and aromatic bonds. As with Lipinski's ro5, the only variable that cannot be visually computed from the molecules structure is logP, therefore the need for its calculation is apparent.

The global agrochemical industry as of 2016 had a market value of USD 215 billion, which is predicted to rise to USD 310 billion by 2025.[30] Pesticide compounds are used in quantities equal to 1.1 million kg every year.[31] The ever increasing demand for crop protection chemicals is expected to be the largest driving factor of this growth,[30] as can been seen in Figure 2.5 . It can be seen that the need for logP data towards the development of pesticides will only grow in the coming years, there is therefore a need for the science of logP data collection to expand.

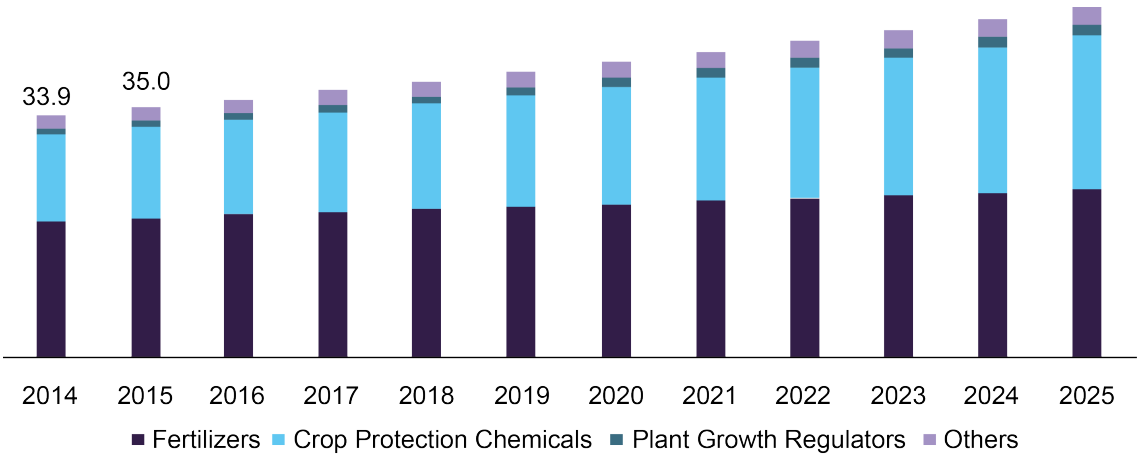

Figure 2.5: US market value for agrochemicals separated by key categories, known values for 2014, 2015 and projections for 2016-2025. Data from Grand View Research.[30]

2.2 Laboratory Measurement of LogP

Experimental values of logP in literature have been acquired through a variety of experimental procedures. These procedures are divided into two camps: direct and indirect. Direct methods determine logP by measuring the concentration of a solute partitioned between octanol and water phases. Indirect methods determine logP via calculation/correlation of a parameter other than the concentration of the two layers.

The shake-flask method[32] (Figure 2.6) is the most classical direct measurement of logP and was for many years the recommended method for determination of logP by the Organisation for Economic Cooperation and Development (OECD).[33] A small quantity of solute is dissolved in a flask of octanol and water. The flask is then shaken to accelerate the partition equilibrium of the solute, the octanol and water layers are then left to separate once more. After full separation has been completed, either one or both of the layers are analysed and the concentration of solute in the layer is determined, from which logP is calculated. The concentration determination is commonly done either via UV-vis spectroscopy or high performance liquid chromatography (HPLC)[1,3,34]

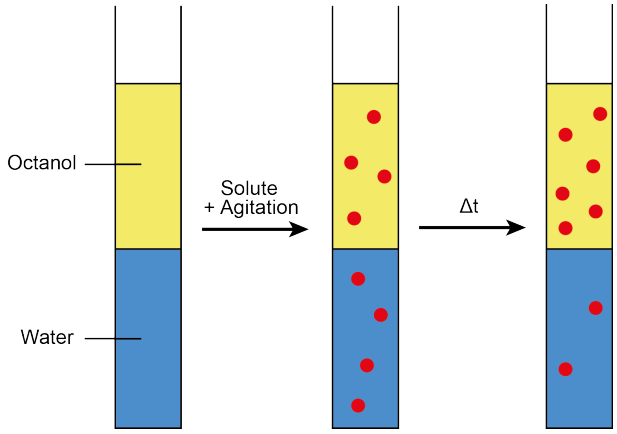

Figure 2.6: Image depicting the process of the shake-flask logP measurement method.

The shake-flask technique has its advantages: it is cheap, does not require any specialist equipment, and is a simple experiment, making it the most accessible method to measure the partition coefficient. However, it has a number of drawbacks: it is prone to emulsion problems, requires large amounts of pure compounds, has a low level of available automation, and is a time consuming process taking several hours per sample[3] leading to low throughput. The latter two are the most significant drawbacks.[1,3,34]

A number of remedies have been proposed throughout the years to address the aforementioned drawbacks. Notably, Hill et al.[35] suggested these 5 improvements:

1. Chromatography sample vials as containers for both equilibration and analysis.
2. Measure only the solute concentration in the water layer, from which the octanol concentration can be calculated.
3. Vial roller used for 30 minutes for equilibration step, giving uniform mixing across samples and preventing emulsion.

4. Direct injection from vial into the HPLC system without allowing time for separation.

5. Use of a fast gradient method for sample analysis.

These improvements allow for a sample to be processed in approximately an hour, a significant improvement on the standard experimental protocol, yet is still a slow process. A fully automated solution was proposed by Hitzel et al..[36] Instead of vials a 96-well plate is used. A robotic pipetting system generates the partitions of octanol and water, and the introduction of the sample. Samples are equilibrated for 30 minutes before fast gradient analysis is performed, which gains results in 10 minutes. Even with these improvements, processing a large sample of molecules takes significant time when considering the large chemical space that has unknown logP values.

Other direct methods of measuring logP include: slow-stirring,[37] potentiometric titration,[38] flow-based,[39] and water-plg aspiration/injection.[40]

2.3 Cheminformatical Methods

The available experimental logP data is insignificant when compared to the quantity of compounds for which this metric is desired and unknown.[41] This had lead to a push to find *in silico* methods for computation of logP values. *In silico* methods come with a variety of benefits over traditional experimental measurements. As discussed in Section 2.2, laboratory methods have a limited throughput capacity. The implementation of computer-based logP calculations allow for a greatly increased throughput, combined with the green chemistry benefits of requiring no physical compound material or solvents to be used. Following on from the lack of need for physical compound material to be used to gain a measurement, *in silico* methods allow for the calculation of a logP without first having to synthesise the molecule in question. This is of great benefit to both research and industry: by knowing the logP of a compound before expending resources into the creation process, molecules unfit for their intended purpose (e.g. molecules with a logP>5 in the use of oral pharmaceutical drugs as dictated by Lipinski's ro5) can be avoided in favour of better candidates. These *in silico* methods are being improved year on year, however, they are still no substitute for experimental values when it comes to accurately predicting logP values.[41] There are three basic categories of cheminformatical methods to calculate logP: atomistic, fragment, and property based.

2.3.1 Atomistic

First introduced by Ghone and Crippen[42] and dubbed 'AlogP'. This technique operates by splitting molecules into single atoms, each atom type has a contribution to logP and by summing all of these contributions a prediction of the molecule's overall logP is formed. Each atom of the periodic table that has been parameterised for the AlogP model has numerous atom types associated with it that accounts for the structural environment the atom is placed in, for a total of 120 atom classifications.[41]

XLOGP2 and XLOGP3[43] are modifications based on the AlogP approach, they were developed in an attempt to address some of the shortcomings of atomistic method by introducing correction factors. This implementation includes contributions from each atoms neighbours and correction factors allow for the larger electronic effects such as internal hydrogen bonds (H-bonds) and adjacency to π systems. These corrections allow for the model to better represent larger structures. XLOGP3 has the additional use of a reference compound of a known experimental logP value as a beginning basis for the calculation, this compound is chosen with respect to 2D-structural similarity.

2.3.2 Fragment

Fragment approaches operate by splitting the molecule into "fragments" which are compared against a pre-existing database of experimental logP values, either of fragments or full compounds. Correction factors are used to allow for intramolecular effects to be taken into account. By using fragments of atoms rather than individual atoms, electronic effects between atoms in the fragment are able to be represented in the prediction.[41] CLOGP[44] was the first commercially available logP prediction software[45] and is the most well known.

2.3.3 Property Based

Property based methods use full descriptions of the molecule to determine logP, they are more broad than those previously mentioned as this can be accomplished using empirical relationships, 3D-structure, topological descriptors, or any combination of these.[41]

Relationship

Relationship methods depend on the association between logP and other physical properties. These approaches aim to find physical parameters that accurately describe this energy. One such method which describes the process of solvation is the

Linear Solvation Energy Relationship (LSER).[46] This method utilises 5 descriptors that are built on the incorporation of a solute into a solvent. The dominant descriptors in this relationship are solute size and solute H-bond basicity. The major drawback of such approaches is the need for experimental descriptors for accurate computation of the model.

Topological

The main advantage of topological-based methods of logP prediction is the speed at which molecules can be processed, in fact "these algorithms can be hundreds of thousands times faster than the resource-demanding algorithms based on *ab initio* or (molecular dynamical) calculations".[41] Speed of processing is particularly important for industries such as pharmaceuticals, where screening large numbers of potential drug candidate molecules is required.

MLOGP[47] is a notable example of topological-based approaches. It first operated by taking the sum of lipophilic atoms (number of carbon and halogenic atoms), and the sum of hydrophilic atoms (number of oxygen and nitrogen atoms) as the only descriptors. Their dataset had a total of 1230 molecules of general organic compounds "together with various drugs and agrochemicals", although there is no evidence presented on the ratio of small molecules compared to larger drug-like molecules in the dataset. With just these two variables, their model had a standard deviation of the error (SDE) = 0.912 and a $R^2 = 0.730$. After 11 correction factors such as presence of ring structures, proximity effect of nearby hydrophilic atoms, quantity of unsaturated bonds etc., the model was improved to a SDE = 0.41 and a $R^2 = 0.91$. The elementary nature of MLOGP allows for fast results, as a product of this MLOGP was implemented into popular molecular descriptor software DRAGON.[48]

ML is becoming an ever increasing focus in the area of chemical information prediction,[49] and logP is no exception. One noteworthy example of the use of ML in the field of logP prediction is ALOGPS,[50] which applies an associative neural network[51] ML method. This model makes use of electrotopological state (E-state) descriptors, which includes the electronic state of the atoms in tandem with the topological characteristics. Use of this ML technique was able to garner a root mean square error (RMSE) = 0.35, for their dataset of $\approx$ 13,000 molecules.

2.4 Quantum Mechanical Measurement

QM prediction initially began as a property based cheminformatic method, where QM was used to generate molecular descriptors. Semi-empirical QM approaches were first introduced in 1969,[52] at the time semi-empirical calculations were required due to the limited computational power. These methods performed their calculations only in the gas phase of the molecule in question, ignoring the effect of either octanol or water on the molecule's geometry and energy due to computational limitations. This approach and subsequent work[53,54] proved that QM-calculated descriptors, such as atomic charges and dipole moments, could be used to predict logP for small, basic molecules.

As computation power has continued to grow and develop through the years, full *ab initio* implementations of logP calculation have been enabled. This has allowed direct calculation of the solvation energy to the end of logP computation. One such study by Chuman et al.[55] was performed by optimising the molecular structure of the compound in question in both octanol and water using Hartree-Fock (HF) and the 3-21G* basis set, solvation energy was then calculated through the B3LYP/6-31+G(d) level of theory. This was done using COSMO,[56] which models the solvent as a continuum. This method combines use of ΔG_{OW} and the water accessible surface area. This model struggled most with prediction of logP for molecules in their amphitropoic set with a SDE = 0.50 and a R^2 = 0.895, which have a range of logP $\approx$ -2 to 3.5. This is notable as it includes a majority of the range specified in the ro5, and therefore its practical applications are questionable. Furthermore, their dataset included only small molecules with fewer than 30 atoms. More recent approaches to QM have yielded increasingly superior results. A 2017 paper[31] utilises DFT and the universal Solvation Model based on Density (SMD) for modelling of octanol and water, for a dataset of 22 pesticides. Their most successful level of theory [M06L/6-31+G(d,p)] gave a RMSE = 0.53.

2.5 Quantum Mechanical and Machine Learning Method Comparison

The Statistical Assessment of the Modelling of Proteins and Ligands (SAMPL6) part II logP prediction challenge[57] was a community-wide blind prediction challenge issued in 2018. The SAMPL challenges aim to progress the field of computational prediction methods towards to the embetterment of *in silico* drug development. In this challenge, participants were asked to predict the logP value of 11 small aromatic

molecules, which can be seen in Figure 2.7. This was a blind challenge where experimental data for the molecules was not available until after the closure of submissions. In total, there were 91 submissions made from 27 individual research groups. This challenge presents a unique opportunity to compare, modern computational approaches towards the prediction of logP for the same data sample. Submissions had overall RMSEs ranging from $0.38 - 5.46$ logP units. The two methods of interest for this thesis are of course QM and ML.

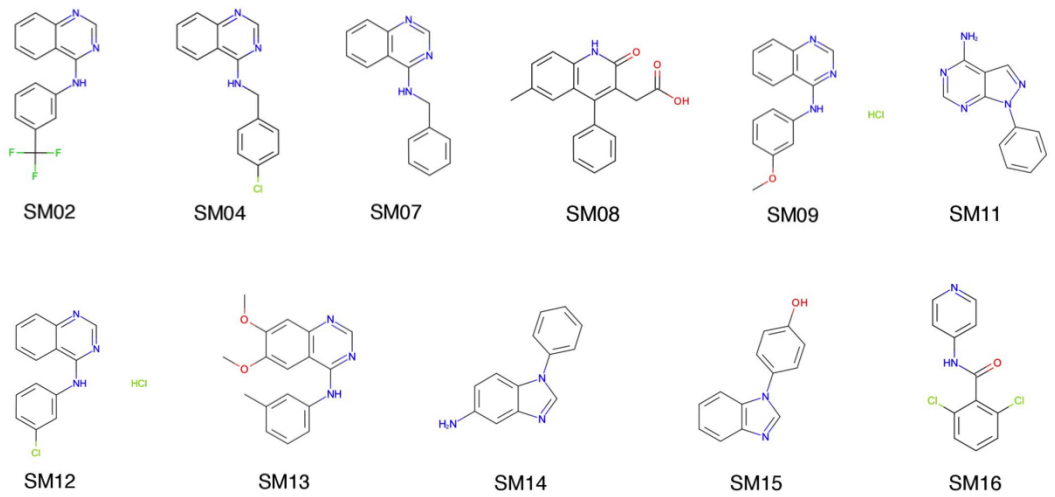

Figure 2.7: Molecular structure of the 11 compounds for logP prediction in the SAMPL6 challenge.[57]

Starting with the QM, these models performed very well overall, with 5 models having a RMSE < 0.5.

$$logP = \frac{(\Delta G_H - \Delta G_S)}{RT(ln10)} \tag{2.1}$$

A submission by Arslan et al.[58] made use of DFT and implicit solvent modelling through SMD, as well as explicit solvent modelling through the use of a single water molecule. They evaluated the use of the B3LYP, M06-2X and ωB97XD functions and the 6-311+G(d,p) and 6-31G(d) basis sets. They optimised the geometry, then calculated the solvation energies of the molecules in both octanol and water to find their logP prediction values through use of Equation 2.1, where ΔG_H denotes hydration free energy and ΔG_S denotes solvation free energy. They found that the B3LYP/6-311+G(d,p) was the most accurate level of theory, and interestingly, that B3LYP/6-311+G(d,p) theory performed significantly better than the B3LYP/6-31G(d) theory, whereas in this work minimal difference between the two basis set complexities was found. The molecules predicted for the SAMPL6 were considerably larger than the molecules in this work, which would indicate that as

molecule size increases, the need for more complex orbital description is required. As will be discussed in Section 3.1.12, use of only one explicit solvent molecule is not seen as an accurate method of modelling the electronic change experienced when in a solvent. However, in most cases Arslan et al. found that this explicit method gained better results than the implicit SMD method. They believe that this is due to the H-bond interactions being more accurately described. Their submission to the challenge had a RMSE of 1.50, which was high for the QM methods, around the bottom third of submissions.

Another group[58] also made use of the M06-2X functional and the SMD solvation model, but to a greater effect. As with the previously mentioned submission, they optimised geometries and then calculated solvation energies to the end of logP calculation through the use of Equation 2.1. With the use of the M06-2X/def2-SVP level of theory, they were able to achieve a RMSE = 0.49, a notable improvement on the Arslan et al. submission. As Arslan et al. tested the M06-2X functional and found that it was outperformed by B3LYP, this could suggest that the def2-SVP basis set gives a significantly better representation of the SAMPL6 molecules and has led to the 3-fold improvement in error. This submission had the joint 4th lowest RMSE value out of all attempts of the challenge.

The best performing submission was also QM, based on the Conductor-like Screening Model for Realistic Solvation (COSMO-RS) implicit solvation model through use of the COSMOtherm software. They conducted geometry optimisation of their molecules through the BP86/TZVP level of theory (BP-TZVP-COSMO), for some molecules a further step of optimisation utilising the def2-TZVPD basis set (FINE-COSMO) was done. FINE-COSMO calculations of the molecule's energies were computed and Equation 2.1 was used to calculate logP values. These calculations obtained a RMSE of 0.38. The most significant difference between this submission and the other two QM-based submissions previously discussed is the use of the COSMOtherm method, where a property known as sigma profiles is calculated. These profiles are then used to find logP. This suggests that either the sigma profile method is more accurate than direct calculation of the molecule's energies, or the COSMO solvation model is more accurate than other solvation models. The latter would suggest that the solvation model used could have a key effect on the accuracy of the results gained.

In the SAMPL6 challenge, ML models were to be presented under the empirical methods category, however many showed up as mixed as they employ either QM or molecular modelling methods to obtain their training data. In comparison to the

QM submissions, there were fewer ML submissions and this should be noted for the sake of fair comparison.

One ML attempt was by Patel et al.[59] Interestingly, their approach included QM methods, but instead of calculating hydration and solvation energies, it was used to create molecular descriptors for a ML model. These QM-generated descriptors were used in conjunction with 2D descriptors which were fed into a multilinear regression model and a partial least-squares model. They trained their ML models using 97 molecules selected from the Sangster database[3] (a database also used in this work) that were hand selected according to structural similarities to those molecules in the SAMPL6 challenge, which can be viewed in Figure 2.7. When fed into the multilinear regression model a RMSE = 0.76 was achieved, this was improved on in their final partial least-squares model which achieved a RMSE = 0.41. When their most successful partial least-squares model was applied to blind challenge molecules, they achieved a RMSE prediction of 0.69, this was the 4th most accurate ML-based model. The molecules in the Sangster database are, in general, smaller than the challenge molecules which can explain the large difference in the model's in-house accuracy when compared to the accuracy achieved when the model was tested against the challenge molecules. This illustrates a principal drawback of ML models: they struggle to accurately predict molecules outside of the training conditions. The increase in molecule size allows for more interactions between atoms/functional groups/fragments etc., which the model can not account for.

The best performing ML-based method was submitted by Wang & Riniker[60] using their recently developed molecular dynamics fingerprints method.[61] This operates by saving molecular modelling data as floating-point vectors, which are used as the features for the ML algorithm. Interestingly, they found that using a ML model to predict solvation free energies and then using Equation 2.1 gave more accurate predictions than using the fingerprint data to directly predict logP through ML.[60,61] The simulations were run for 5 ns for each molecule, once in water and once in octanol. The fingerprints include such data as the radius of gyration, solvent-accessible surface area, solute-solute and solute-solvent interaction energy etc. These are combined with simple count descriptors, number of X atoms, number of rotatable bonds etc. The data was then fed into a meta-learner based ML model, which operates by applying many ML models and making its prediction based on the average of them. Using this approach, they obtained a RMSE = 0.53, the 13th highest overall.

Overall it can be seen that currently, at least for molecules that resemble those from the SAMPL6 logP prediction challenge, that QM mechanical approaches are more

accurate than ML approaches. Furthermore, those ML approaches that performed well did so with the use of molecule descriptors gained through either QM or molecule modelling simulation, which adds further computational expense to the endeavour.

Chapter 3

Theory

3.1 Quantum Mechanics[62,63]

Quantum mechanics is applied by viewing particles (such as electrons) as possessing both the characteristics of a particle and of a wave. The wave properties of a particle are described by the wavefunction Ψ, which contains all electronic properties of the particle. Therefore, from Ψ, theoretically any property can be extracted which is why quantum mechanics is of great interest to the field of chemistry. Ψ is not an observable quantity and therefore does not have a physical representation, however, Ψ^2 can be interpreted as the probability density of finding a given electron in a given volume in space.

Ψ is applied through the Schrödinger equation,[64] the full, time-independent form of the which is:

$$\left\{-\frac{h^2}{2m}\nabla^2 + V(\vec{r},t)\right\}\Psi(\vec{r},t) = i\hbar\frac{\partial\Psi(\vec{r},t)}{\partial t} \tag{3.1}$$

Where ∇^2:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{3.2}$$

Equation 3.1 describes a single particle of mass m moving through space in relation to position vector $\vec{r}$, where $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$ and time (t), whilst subject to an external potential $V(\vec{r},t)$. This can be reduced if we consider the equation to be independent of time:

$$\left\{-\frac{h^2}{2m}\nabla^2 + V(\vec{r})\right\}\Psi(\vec{r}) = E\Psi(\vec{r}) \tag{3.3}$$

Where E represents the energy of the particle. The LHS of equation is known as

the Hamiltonian operator $\hat{H}$:

$$\hat{H} = -\frac{h^2}{2m}\nabla^2 + V(\vec{r}) \tag{3.4}$$

Leaving us with the general form of the Schrödinger equation:

$$\hat{H}\Psi = E\Psi \tag{3.5}$$

3.1.1 Operators

Equation 3.5 is known as a partial differential eigenvalue equation. These can be solved using an operator that acts on the equation (an eigenfunction) which returns the function acted upon by some scalar (an eigenvalue). Operators allow for the desired properties to be extracted from the wavefunction, with each property having a complementary operator. In the case of Equation 3.5, $\hat{H}$ is the operator whilst E is the observable.

A more general version of a partial differential eigenvalue equation takes the form of:

$$\hat{O}\Psi = o\Psi \tag{3.6}$$

Where $\hat{O}$ is the operator and o the observable.

The average value of an observable is known as the expectation value. In the case of E, this can be calculated by:

$$E = \frac{\int \Psi^* \hat{H} \Psi d\tau}{\int \Psi^* \Psi d\tau} \tag{3.7}$$

Where Ψ^* indicates that the wavefunction could be a complex number and $d\tau$ indicates an integral over all spatial and spin coordinates. The denominator serves as a normalising function. The Hamiltonian operator contains energy in two parts, kinetic and potential. The kinetic operator is simply:

$$-\frac{h^2}{2m}\nabla^2 \tag{3.8}$$

The potential operator is, for a single electron and single nucleus system:

$$V(\vec{r}) = -\frac{Ze^2}{4\pi\varepsilon_0 r} \tag{3.9}$$

Where Z is the quantity of protons in the system, ε is the dielectric constant, and r is the distance between the electron and the nucleus.

3.1.2 Born-Oppenheimer Approximation

Schrödinger's equations can be solved exactly for some grossly simple systems, such as a particle in a 1-D box or the harmonic oscillator. However, when attempting a three-body problem (such as the helium atom) or higher body orders, exact solutions become impossible, therefore we must apply approximations to get close to exact answers. The Born-Oppenheimer approximation[65] assumes that the position of the nucleus is stationary with respect to the electrons in the system. This is achieved by considering the relative mass ratio of the nucleus and its electrons. As the mass of the nucleus is many times greater than the mass of an electron, and as kinetic energy is defined as $E_K = mv^2/2$, the velocity of the electrons can be viewed as infinitely greater than the velocity of the nucleus. This allows us to assume that electrons can almost instantaneously re-position themselves around the nucleus, therefore we can assume that the kinetic component of the wavefunction concerning the nucleus is equal to 0.

We have seen that the Hamiltonian Equation 3.4 can be divided into kinetic and potential energy terms, this can be further expanded, for an example where there are two (A & B) nuclei (M) and two (i & j) electrons (N):

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \frac{1}{2}\sum_{A=1}^{M}\frac{1}{m_A}\nabla_A^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>1}^{N}\frac{1}{r_{ij}} + \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{R_{AB}} \quad (3.10)$$

The first two terms describe the kinetic energy, while the remaining three describe the potential energy, where R_{AB} denotes the distance between the two nuclei. When applying the Born-Oppenheimer approximation to this system, the second term is removed due to kinetic energy of the nucleus being zero. The last term which describes the potential energy is removed due to the nucleus-nucleus interaction being constant, removing the term's dependence on 3D space. This is known as the electronic Hamiltonian:

$$\hat{H}_{elec} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>1}^{N}\frac{1}{r_{ij}} = \hat{T} + \hat{V}_{Ne} + \hat{V}_{ee} \quad (3.11)$$

Where the subscript Ne describes the nucleus-electron interactions, and ee describes the electron-electron interactions. As the nucleus-nucleus potential energy term is now a constant, it is found to be equal to the energy of the nuclear component:

$$E_{nuc} = \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{r_{AB}} \quad (3.12)$$

The new operator for the electronic version of the Hamiltonian is:

$$\hat{H}_{elec}\Psi_{elec} = E_{elec}\Psi_{elec} \tag{3.13}$$

And the total energy of the system is therefore the sum of the electronic and nuclear energies:

$$E_{tot} = E_{elec} + E_{nuc} \tag{3.14}$$

3.1.3 Variational Principle

To solve the Schrödinger equation for a given molecule, an approximate Hamiltonian operator of the specific system must be developed. Looking at Equation 3.11, we can see that the only variables under the Born-Oppenheimer approximation that are system-specific are the number of electrons and the external potential. Finding these parameters is the first step, the second step is to find the eigenfunctions of the wavefunction (Ψ_i) along with the eigenvalues of the Hamiltonian (E_i). With Ψ_i defined, the Schrödinger equation can be solved to find any number of experimental properties, theoretically. In principle however, this cannot be done except for some simple systems.

Variation principle allows for finding an approximate value that is close to the ground state wavefunction, Ψ_0, which corresponds to the lowest energy state of the system, E_0. This is done by a systematic approach from the equation:

$$E_{trial} = \frac{\langle \Psi_{trial} | H | \Psi_{trial} \rangle}{\langle \Psi_{trial} | \Psi_{trial} \rangle} \geq E_0 \tag{3.15}$$

Where the subscript "trial" denotes that the parameter is an approximate of the real value. As this is a function of a function, these "trial" wavefunctions are termed wavefunctionals. With the variation method, the energy found will always be larger than the true E_0 value. This can be reduced by normalising the wavefunction i.e. when $\int \Psi^2 d\tau = 1$, to:

$$E_{trial} = \left\langle \Psi_{trial} \middle| \hat{H} \middle| \Psi_{trial} \right\rangle \geq E_0 \tag{3.16}$$

Finding the ground state wavefunctional and energy is performed via systematically searching through acceptable N values. As it is known that the trial wavefunctional will always be of greater energy than the ground wavefunction, this makes selection of the most accurate wavefunctional trivial, it is simply the wavefunctional corresponding to the lowest energy.

3.1.4 Hartree-Fock Approximation

Searching though all potential acceptable N values is an infinite task, therefore again, we must search for a reasonable approximation by choosing an appropriate subset of N values. The Hartree-Fock equations offer the simplest approach to this problem. The energy of the most accurate wavefunctional is a minimum value, therefore we can search for it by finding where δE is zero. Furthermore, as the total wavefunction (and therefore also the wavefunctional) must be antisymmetric under the Pauli Exclusion Principle,[66] an antisymmetric product of N single electron wave functions $\chi_i(\vec{x}_i)$, can be taken. This is known as a Slater determinant, Φ_{SD}:

$$\Psi_0 \approx \Phi_{SD} = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\vec{x}_1) & \chi_2(\vec{x}_1) & \cdots & \chi_N(\vec{x}_1) \\ \vdots & \vdots & & \vdots \\ \chi_1(\vec{x}_N) & \chi_2(\vec{x}_N) & & \chi_N(\vec{x}_N) \end{vmatrix} \tag{3.17}$$

By only considering the diagonal elements, this can be simplified:

$$\Phi_{SD} = \frac{1}{\sqrt{N!}} det\left\{\chi_1(\vec{x}_1)\ \ \chi_2(\vec{x}_2)\cdots\chi_N(\vec{x}_N)\right\} \tag{3.18}$$

These one-electron functions $\chi_i(\vec{x}_i)$ are referred to as spin orbitals, comprised of a spatial orbital $\phi_i(\vec{r})$ and one of either $\alpha(s)$ or $\beta(s)$ spin orbitals.

$$\chi(\vec{x}) = \phi_i(\vec{r})\sigma(s),\ \sigma = \alpha, \beta \tag{3.19}$$

The spin functions introduce orthonormality to our equation, as:

$$< \alpha \mid \alpha > \ = \ < \beta \mid \beta >, \ \ and \ \ < \alpha \mid \beta > \ = \ < \beta \mid \alpha > \ = \ 0 \tag{3.20}$$

The variation principle discussed in Section 3.1.3 can now be applied to find the most accurate Slater determinant; the determinant that gives the lowest energy. χ_i is varied until the found energy is a minimum, under the condition that the spin orbitals are orthonormal. Therefore, the energy under the HF approximation is given by:

$$E_{HF} = \left\langle \Phi_{SD} \mid \hat{H} \mid \Phi_{SD} \right\rangle = \sum_i^N (i \mid \hat{h} \mid i) + \frac{1}{2}\sum_i^N \sum_j^N (ii \mid jj) - (ij \mid ji) \tag{3.21}$$

Where the kinetic contribution of the HF Hamiltonian is given by:

$$(i \mid \hat{h} \mid i) = \int \chi_i^*(\vec{x}_1) \left\{ -\frac{1}{2}\nabla^2 - \sum_A^M \frac{Z_A}{r_{1A}} \right\} \chi_i(\vec{x}_1) d\vec{x}_1 \tag{3.22}$$

The coloumbic terms are represented by:

$$(ii \mid jj) = \iint |\chi_i(\vec{x}_1)|^2 \frac{1}{r_{12}} |\chi_j(\vec{x}_2)|^2 \, d\vec{x}_1 d\vec{x}_2 \tag{3.23}$$

And the exchange terms are represented by:

$$(ij \mid ji) = \iint \chi_i(\vec{x}_1)\chi_j^*(\vec{x}_1)\frac{1}{r_{12}}\chi_j(\vec{x}_2)\chi_i^*(\vec{x}_2) d\vec{x}_1 d\vec{x}_2 \tag{3.24}$$

Where Equations 3.23 & 3.24 are the sum of the electronic terms of the HF Hamiltonian.

As stated previously, χ_i must be orthonormal to be valid. This problem can be solved by the inclusion of Lagrangian multipliers, ϵ_i, which are used to represent the ideal spin orbitals. With their addition, the HF equation of spin orbitals corresponding to lowest energy is:

$$\hat{f}\chi_i = \epsilon_i, \quad \chi_i, i = 1, 2, ...N. \tag{3.25}$$

Where $\hat{f}$ is known as the Fock operator. Physically, ϵ_i represent orbital energies, $\hat{f}$ as a one-electron operator is described by:

$$\hat{f}_i = -\frac{1}{2}\nabla_i^2 - \sum_A^M \frac{Z_A}{r_i A} + V_{HF}(i) \tag{3.26}$$

$V_{HF}(i)$ is known as the HF potential: the average repulsive potential felt by the i'th electron from the other electrons in the system. This replaces $1/r_{ij}$ in the original Hamiltonian, going from a two-electron description to a single-electron description. This reduces the complexity of the equation instead by taking an average of experienced electron repulsion. The HF potential allows for a potentially many-body problem, for which no exact solution can be found, to be interpreted by a single body approximation. V_{HF} is equivalent to:

$$V_{HF}(\vec{x}_1) = \sum_j^N \left(\hat{J}_j(\vec{x}_1) - \hat{K}_j(\vec{x}_1)\right) \tag{3.27}$$

As only one electron is now being considered, we can drop i. There are two components to V_{HF}, again split into coloumbic and exchange terms. The coloumbic terms are defined by:

$$\hat{J}_j(\vec{x}_1) = \int |\chi_j(\vec{x}_2)|^2 \frac{1}{r_{12}} \, d\vec{x}_2 \tag{3.28}$$

This coloumbic operator represents the force felt by an electron at $\vec{x}_1$ by another electron in the spin orbital χ_j. This is a local term, as it is applicable only to an electron in the spin orbital χ_i at a position $\vec{x}_1$.

The exchange term:

$$\hat{K}_j(\vec{x}_1)\chi_i(\vec{x}_1) = \int \chi_j^*(\vec{x}_2)\frac{1}{r_{12}}\chi_i(\vec{x}_2)d\vec{x}_2\chi_j(\vec{x}_1) \tag{3.29}$$

The exchange operator $\hat{K}$ represents the exchange interaction between all same-spin electron pairs. As seen from the LHS of the equation, the spin orbital χ_1 depends on all spacial positions of $\vec{x}_1$, and since we integrate over $\vec{x}_2$, it is valid over all space. Therefore it is deemed to be global.

The expectation values of operators $\hat{J}$ & $\hat{K}$ are the solutions to Equations 3.23 & 3.24 respectively.

Although Equation 3.25 at first glance appears to be a eigenvalue problem, where χ_i is the eigenfunction of operator $\hat{f}$, and ϵ_i is the eigenvalue. It is in fact a pseudo-eigenvalue problem, as it cannot be solved in a closed form. This is due to $V_{HF}(i)$ being subject to the effect of all the other electrons in the system averaged over all positions. Once one electron has been solved, its orbital changes, therefore all over orbitals change. This loops continuously as each spin orbital is solved and cannot be closed. Instead, this problem such be solved iteratively using what is known as the self-consistent field (SCF) method.

SCF operates by solving the HF equations for a guess set of orbitals. A slight variation is then made to the orbitals and equations are solved again, this continues until the output is within a predetermined threshold. In "closed-shell systems", systems where there are an even number of electrons such that all electrons are paired, the energy can converge after a small quantity of iterative cycles in a method known as the restricted HF approximation. This method is termed "restricted" as the two spin orbitals are deemed to both occupy the same spatial orbital, each with a spin function (α or β), therefore are degenerate and have the same energy.

"Open-shell" systems are those in which either there is an uneven number of electrons or there are an even number of electron but where not all orbitals are paired. These are a more complicated case but are more rare than their closed counterparts. The most popular solution to open-shell problems is the unrestricted HF. In this method, the α & β orbitals each experience their own external potential: V_{HF}^{α} & V_{HF}^{β} respectively.

3.1.5 Electron Correlation

As discussed in Section 3.1.3, any approximate wavefunctional through the use of the variational principle will have a ground energy higher than that of the true ground energy. The correlation energy[67] E_{HF}^{C} describes the difference between these two energies:

$$E_{HF}^{C} = E_0 - E_H F \tag{3.30}$$

The most significant cause of E_{HF}^{C} is due to electron correlation: instantaneous repulsion between electrons in the systems, which is not considered in the HF equations. This is overestimated as $V_{HF}(i)$ only considers the mean potential exerted by the other electrons in the system, it does not consider them individually and hence they are able to move closer spatially than in a real system. It is termed the dynamical E_{HF}^{C} as the effect depends on the proximity of electron 1 & electron 2 in space, with the dynamical E_{HF}^{C} increasing as the electrons move increasingly closer together.

The non-dynamical E_{HF}^{C} is the next greatest contributor. This is present due to the fact that a Slater determinant can be solved in a non-singular amount of methods, e.g. the restricted and unrestricted HF methods discussed in Section 3.1.4. Under certain conditions, some methods will be worse approximations than others. As can be seen in Figure 3.1, when the distance between hydrogen atoms increases, the restricted HF method begins to perform worse than the unrestricted HF method.

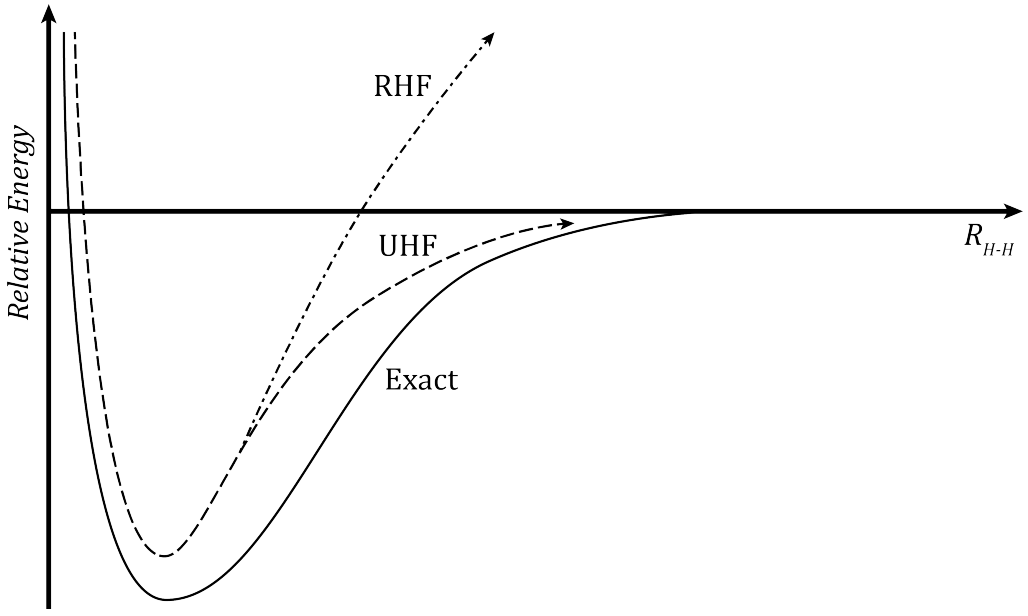

Figure 3.1: Potential energy curves for a H_2 molecule, showing the change exact, restricted, and unrestricted energy with increasing distance between hydrogen atoms.

3.1.6 Electron Density

Electron density is the measure of the probability to find an electron in a given spatial location:

$$\rho(\vec{r}) = N \int \cdots \int \mid \Psi(\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N) \mid^2 ds_1 d\vec{x}_2 \cdots d\vec{x}_N \tag{3.31}$$

$\rho(\vec{r})$ is the electron density: the probability of any electron within N, in the volume $d\vec{r}_1$. This electron can be in any spin, whilst all electrons can be in any spatial position and spin in the wavefunction Ψ. As electrons are indistinguishable, to find the probability of finding any of the N electrons is simply $p(\vec{r})N$. Furthermore, electron density integrates to the number of electrons in the system:

$$\int \rho(\vec{r}) d\vec{r}_1 = N \tag{3.32}$$

Electron density is an observable property and can be directly measured. Due to the positive pull an electron experiences from its nucleus, there is a limit to the distance it can exist from its nucleus, therefore there is an upper limit on the electron density:

$$\lim_{r_{iA} \to 0} \left[\frac{\partial}{\partial r} + 2Z_A \right] \bar{\rho}(\vec{r}) = 0 \tag{3.33}$$

Where $\bar{\rho}(\vec{r})$ is the spherical average of $\rho(\vec{r})$.

Pair density is based on the idea that since electrons are often paired, by finding the density of one of the electrons in the pair we can intrinsically know the density of the other electron of the pair. By knowing the density probability of an electron of spin σ_1 in volume $d\vec{r}_1$, we can find the probability density of an electron with spin σ_2 in volume $d\vec{r}_2$, while all other electrons in the system have arbitrary spins and positions. This is known as pair density:

$$\rho_2 = (\vec{x}_1 \vec{x}_2) = N(N-1) \int \cdots \int \mid \Psi(\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N \mid^2 d\vec{x}_3 \cdots d\vec{x}_N \tag{3.34}$$

Electron density offers a pathway to reduce the computationally intensive electron wave function that depends on $4N$ variables, (three spatial and one spin) by the simpler description given by electron density. The hypothesis being that although the wavefunction does contain all information about the properties of a system, it contains a large quantity of undesirable information and is therefore over-complicated for the purpose of a computational chemist. We sacrifice the all-encompassing knowledge of the wavefunction, but are able to perform a reduced calculation and still obtain the useful information, such as energy, that we seek. This forms the basis

for DFT.

3.1.7 Hohenberg-Kohn Theorems

The first concrete implementation of DFT was introduced by Hohenberg and Kohn, built from the fledgling introduction of electron density's use from the Thomas-Fermi[68,69] model. The first theorem was proof that, through use of electron density, the Hamiltonian could be solved. The first theorem states:

"The external potential $V(\vec{r})$ is a unique functional of electron density $\rho(\vec{r})$; since, in turn $V(\vec{r})$ fixes $\hat{H}$ we see that the full many particle ground state is a unique functional of $\rho(\vec{r})$"

This was found through an *reductio ad absurdum* of two external potentials having the same internal potential. Hence it can be said that the ground electron density, ρ_0, is equivalent to:

$$\rho_0 \Rightarrow N, V(\vec{r}) \Rightarrow N, Z_A, R_A \Rightarrow \hat{H} \Rightarrow \Psi_0 \Rightarrow E_0 \tag{3.35}$$

This theorem proved that there could not be two $V(\vec{r})$ that give the same electron density. Therefore, there is a one-to-one mapping relationship between a specific electron density and a specific $V(\vec{r})$. As $V(\vec{r})$ contains Z_A & R_A, and the integral of ρ_0 contains N, hence ρ_0 contains all information of the Hamiltonian.

E_0 is a functional of ρ_0, therefore all the components (as can be seen in Equation 3.11) of E_0 are also dependent on ρ_0:

$$E_0[\rho_0] = T[\rho_0] + E_{Ne}[\rho_0] + E_{ee}[\rho_0] \tag{3.36}$$

Of these variables, only $E_{Ne}[\rho_0]$ is system dependent, while the other two terms can be grouped into what is known as the Hohenberg-Kohn potential, $F_{HF}[\rho_0]$, as they are not dependent on N, Z_A or R_A. $E_{Ne}[\rho_0]$ can be further expanded, then Equation 3.36 becomes:

$$E_0[\rho_0] = \int \rho_0(\vec{r}) V_{Ne} d\vec{r} + F_{HF}[\rho_0] \tag{3.37}$$

The second theorem proved that electron density obeys the variations principle. This is important as we must be able to prove that the electron density found is indeed the ground electron density. This theorem states that $F_{HF}[\rho]$, which gives the ground state energy, gives the ground state energy of the system only when

supplied with the ground state density:

$$E_0 \leq E[\tilde{\rho}] = T[\tilde{\rho}] + E_{Ne}[\tilde{\rho}] + E_{ee}[\tilde{\rho}] \tag{3.38}$$

As a consequence of this, any trial density ρ_{trial} values must always be higher than the true ground state density ρ_0. Therefore, by varying ρ_{trial}, we can search for a minimum value where $\delta\rho = 0$, at this point we have found ρ_0.

The combination of both Hohenberg-Kohn theorems prove that the wavefunction can in fact be replaced by electron density in the Schrödinger equation. Now a practical implementation of this is needed.

3.1.8 Kohn-Sham Approach

The Kohn-Sham approach set out to find a better representation of the kinetic energy term than the previous work of Thomas-Fermi. They did this using Slater determinants, as discussed in Section 3.1.4. In fact, much of the Kohn-Sham approach built from the Hartree-Fock equations. Slater determinants are used to approximate the full N wavefunction, but can be considered to be an exact representation of the wavefunction for a fictitious system of non-interacting electrons in the V_{HF} potential. This form of the kinetic energy wavefunction is:

$$T_{HF} = -\frac{1}{2}\sum_{i}^{N}\left\langle \chi_i \mid \nabla^2 \mid \chi_i \right\rangle \tag{3.39}$$

The HF spin orbital in Equation 3.39 has a corresponding E_{HF} expectation value of ground state energy:

$$E_{HF} = \min_{\Phi_{SD}\to N}\left\langle \Phi_{SD} \mid \hat{T} + \hat{V}_{Ne} + \hat{V}_{ee} \mid \Phi_{SD} \right\rangle \tag{3.40}$$

As the exact wavefunction of non-interacting electrons are Slater determinants, we can construct a non-interacting reference system, containing a local potential $V_S(\vec{r}_i)$, where subscript S denotes it is of the fictitious system.

$$\hat{H}_S = -\frac{1}{2}\sum_{i}^{N}\nabla_i^2 + \sum_{i}^{N} V_S(\vec{r}_i) \tag{3.41}$$

As this is an non-interacting system there are no terms for describing electron-electron forces. We can construct a Slater determinant of the same form as Equation

3.17 for this system:

$$\Theta_S = \frac{1}{\sqrt{N!}} \begin{vmatrix} \varphi_1(\vec{x}_1) & \varphi_2(\vec{x}_1) & \cdots & \varphi_N(\vec{x}_1) \\ \vdots & \vdots & & \vdots \\ \varphi_1(\vec{x}_N) & \varphi_2(\vec{x}_N) & & \varphi_N(\vec{x}_N) \end{vmatrix} \tag{3.42}$$

Where φ_i is the Kohn-Sham orbitals, similarly to the approach in Section 3.1.4, spin orbitals are found by:

$$\hat{f}^{KS}\varphi_i = \epsilon_i \varphi_i \tag{3.43}$$

Where the one-electron Kohn-Sham operator is defined as:

$$\hat{f}^{KS} = -\frac{1}{2}\nabla^2 + V_S(\vec{r}) \tag{3.44}$$

The fictitious system can be related to the system of interest by choosing a V_S that corresponds to the density resulting from the summation of the moduli of the squared Kohn-Sham orbitals, which is equivalent to the ground state density of the interest system:

$$\rho_S(\vec{r}) = \sum_i^N \sum_S \mid \varphi_i(\vec{r},) \mid^2 = \rho_0(\vec{r}) \tag{3.45}$$

The first step in improving the kinetic energy term calculation was to split the term into those parts which could be calculated exactly, and those which needed to be approximated. The expression to calculate the kinetic energy of the fictitious system of equal density to the real system exactly is:

$$T_S = -\frac{1}{2}\sum_i^N \left\langle \varphi_i \mid \nabla^2 \mid \varphi_i \right\rangle \tag{3.46}$$

Of course, the fictitious and real systems do not have equivalent energies regardless of their equivalent densities. This is resolved through the following equations:

$$F[\rho(\vec{r})] = T_S[\rho(\vec{r})] + J[\rho(\vec{r})] + E_{XC}[\rho(\vec{r})] \tag{3.47}$$

Where $E_{XC}[\rho(\vec{r})]$:

$$E_{XC}[\rho(\vec{r})] \equiv (T[\rho] - T_S[\rho]) + (E_{ee}[\rho] - J[\rho]) = T_C[\rho] + E_{ncl}[\rho] \tag{3.48}$$

Where $J[\rho(\vec{r})]$ is the classical electrostatic electron-electron repulsion term, and E_{XC} is known as the exchange-correlation energy, which contains T_C, a correction kinetic energy for the approximate energy of the true system given by the fictitious system. E_{XC} holds all of the unknown terms of the system: exchange, correlation, parts of

the kinetic energy term etc. E_{ncl} is the non-classical electronic contribution term. Due to the non-interacting nature of the fictitious system, it is a far simpler task to calculate the kinetic energy, then all that is needed to bring the fictitious system to the same exact energy of the real system is to calculate E_{XC}. However, E_{XC} cannot be exactly calculated, we must approximate it through use of the variation principle:

$$E_{XC} = \left(-\frac{1}{2}\nabla^2 + \left[\int \frac{\rho(\vec{r}_2)}{r_{12}} d\vec{(r}_2) + V_{XC}(\vec{r}_1) - \sum_A^M \frac{Z_A}{r_1 A} \right] \right) \varphi_i = \epsilon_i \varphi_i \tag{3.49}$$

Where:

$$V_{XC} \equiv \frac{\delta E_{XC}}{\delta \rho} \tag{3.50}$$

We can then solve this iteratively, just as the HF Equation 3.25, which gives an approximate value of the real system.

3.1.9 Functionals

There are many computational methods to approximately solve the exchange-correlation energy, these are termed functionals. As through DFT, all other parts of $F[\rho(\vec{r})]$ can be solved exactly, the accuracy and speed of your calculation depends solely on the approximation of E_{XC}. Most functional are based on the model of a hypothetical uniform electron gas, "a system in which electrons move on a positive background charge distribution such that the total ensemble is electrically neutral".[62] This is physically represented by the number of electrons and the volume which they occupy both being considered to be infinite, whilst electron density is finite and equal to $N/V = \rho$. This is a system in which exchange and correlation functionals are known almost exactly. This is known as the local density approximation (LDA) and its exchange-correlation energy is:

$$E_{XC}^{LDA}[\rho] = \int \rho(\vec{r}) \varepsilon_{XC}(\rho(\vec{r})) d\vec{r} \tag{3.51}$$

$\varepsilon_{XC}(\rho(\vec{r}))$ represents the exchange-correlation energy per particle of the density, this term can be further split into its exchange and correlation constituents:

$$\varepsilon_{XC}(\rho(\vec{r})) = \varepsilon_X(\rho(\vec{r})) + \varepsilon_C(\rho(\vec{r})) \tag{3.52}$$

The exchange contribution is known, however no exact expression for the correlation contribution is known. We can further break up the equation, instead of considering the electron density, we can consider individual spin densities:

$$\rho(\vec{r}) = \rho_\alpha(\vec{r}) + \rho_\beta(\vec{r}) \tag{3.53}$$

The benefit of having two variables over one is it allows for additional flexibility, it is especially beneficial to the open-shell systems discussed in Section 3.1.4. We can then expand Equation 3.51:

$$E_{XC}^{LDA}[\rho_\alpha, \rho_\beta] = \int \rho(\vec{r})\varepsilon_{XC}(\rho_\alpha(\vec{r}), \rho_\beta(\vec{r}))d\vec{r} \tag{3.54}$$

In cases where $\rho_\alpha(\vec{r}) \neq \rho_\beta(\vec{r})$, termed the spin polarised case, the degree of spin polarisation can be found through the spin-polarisation parameter:

$$\xi = \frac{\rho_\alpha(\vec{r}) - \rho_\beta(\vec{r})}{\rho(\vec{r})} \tag{3.55}$$

Where $\xi = 0$ corresponds to the spin compensated state (equal numbers of each spin state) and $\xi = 0$ corresponds to the fully spin polarised case (all electrons have the same spin).

LDA was initially overlooked in the world of computational chemistry as, of course, electrons are not uniformly distributed through a molecular system. A form more practical to chemistry is known as the generalised gradient approximation:

$$E_{XC}^{GGA}[\rho_\alpha, \rho_\beta] = \int f(\rho_\alpha, \rho_\beta, \nabla\rho_\alpha, \nabla\rho_\beta)\vec{r} \tag{3.56}$$

Much like LDA, E_{XC}^{GGA} can be split into its respective contributions:

$$E_{XC}^{GGA} = E_X^{GGA} + E_C^{GGA} \tag{3.57}$$

E_X^{GGA} can be found as:

$$E_X^{GGA} = E_X^{LDA} - \sum_\sigma \int F(s_\sigma)\rho_\sigma^{4/3}(\vec{r})d\vec{r} \tag{3.58}$$

Where:

$$s_\sigma(\vec{r}) = \frac{|\ \nabla\rho_\sigma(\vec{r})\ |}{\rho_\sigma^{4/3}(\vec{r})} \tag{3.59}$$

3.1.10 Hybrid Functionals

Splitting the exchange-correlation energy into its constituent terms is particularly useful due to two factors:

1. Exchange energy can be calculated exactly
2. And, numerically the exchange energy term has a far greater contribution than the correlation energy

By splitting the terms we can exactly calculate the exchange energy through the use of HF, and then use DFT to approximately calculate the correlation energy:

$$E_{XC} = E_X^{exact} + E_C^{KS} \tag{3.60}$$

This approach is known as hybrid functionals. Both functionals, B3LYP and M11, used in this work fall into this category.

3.1.11 Basis Sets

Basis sets are implementations of either HF or DFT that allow for computationally appropriate methods of solving the equations. At their crux, basis sets are a set of equations that describe the molecular orbitals of the system in question. The most popular type of basis set, and the type that is used in this work, is the Gaussian Type Orbitals:

$$\eta^{STO} = Nx^l y^m z^n exp[-\alpha r^2] \tag{3.61}$$

Where N is a normalisation factor, α represents the size of the orbital (where a small α corresponds to a diffuse orbital and a large α corresponds to a tight orbital). $L = l + m + n$ denotes the extent of the orbitals: $L = 0$ is the s-orbital, $L = 1$ is the p-orbital etc. The popularity for this type of orbital explanation stems from the computational advantage it brings, allowing very efficient algorithms to be used to solve the coulomb and HF-exchange terms.

The number of basis functions used increases the flexibility of the orbital system, at the cost of increased computational cost. The most basic basis functions implement only one function for each orbital, e.g. one for 1s, one for 2s etc. The next step in complexity progression is known as the double-zeta basis set, as one might expect, there are now two functions for each orbital, giving more control on how the orbitals are defined, this can carry on to triple-zeta, quadruple-zeta etc. Polarisability can be accounted for by using the split-valence type, this allows for modelling of orbitals of angular momentum one higher than of those occupied by the atom. The size of these polarised orbitals can be increased through the diffuse terms. Some examples of basis sets used in this work:

1. double-zeta: 6-31G
2. triple-zeta: 6-311G
3. double-zeta split-valence: 6-31G(d,p), where d and p indicates split-valence on non-hydrogen atoms and hydrogen atoms respectively

4. triple-zeta split-valence diffuse: 6-311+G(d), where + and ++ indicates diffuse terms on non-hydrogen atoms and hydrogen atoms respectively

3.1.12 Solvent Models[70]

Until this point the QM discussion has been limited to molecules in the gas phase, however, logP cannot be calculated until the more complicated task of considering solvents is introduced. The most obvious way to model a molecule's behaviour in a particular solvent would be to surround the solute molecule with said solvent, the interactions between the solute and solvent could then be calculated; this is known as explicit solvation. Surrounding a solute molecule in a single shell of solvent molecules would not be an accurate depiction of the real process as more shells are needed, and quickly a computationally inaccessible number is required. A smarter method is through implicit/continuum solvation models, where the space around the solute molecule is modelled as a continuous medium of the properties of the solvent.

The free energy of solvation ΔG_S^0, describes the change in free energy for a molecule A leaving the gas phase and entering a condensed phase. This energy can be found from the equilibrium constant:

$$\Delta G_S^0(A) = \lim [A]_{sol} \to 0 \left\{ -RTln \left. \frac{[A]_{sol}}{[A]_{gas}} \right|_{eq} \right\} \tag{3.62}$$

Where R is the gas constant, T temperature, and subscripts *sol* & *gas* denote the solution and gas phases respectively. The most important physical effects to the solvation process are: electrostatic interactions, cavitation, changes in dispersion, and changes in bulk solvent structure.

The process of solvation is shown in Figure 3.2, a cavity in the solvent is formed to accommodate the solute molecule, the energy cost of this process is termed the cavitation energy. Entropically and enthalpically, this process is unfavourable as this causes a decrease in solvent disorder and solvent-solvent interactions. The solute molecule then enters the cavity, experiencing favorable dispersion interaction with the solvent.

It is important to note that the introduction of solvent changes the solute's electronic structure, for example the dipole moments of a solute are larger when in solution, and therefore the energy of the solute which we desire to calculate through QM methods.

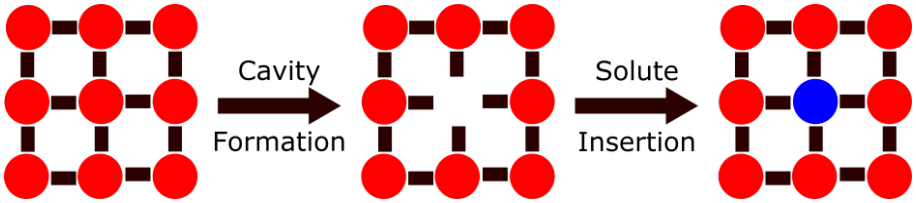

Figure 3.2: Process of solvation. Where red circles denote solvent molecules and the blue circle denotes a solute molecule.

Electrostatics

Electrostatic effects between solute and solvent occur through interaction of their charge distributions. In continuum solvation models, the charge distribution of the solvent is not represented explicitly. Instead, a continuous electric field of the average charge distribution of the solvent, termed the reaction field, is used. The polarisation energy is given by:

$$G = -\frac{1}{2}\int \rho(\vec{r})\phi(\vec{r})d\vec{r} \tag{3.63}$$

Where ρ is the charge density of the solute which is either a continuous function of $\vec{r}$ or as discrete point charges, and ϕ is the electrostatic potential at a point in space. G can be thought to be the energy difference of charging the system in the gas phase and the solution phase, therefore to solve G the electrostatic potential in both phases is required. This can be done through the Poisson equation:

$$\nabla^2\phi(\vec{r}) = -\frac{4\pi\rho(\vec{r})}{\varepsilon} \tag{3.64}$$

Where ε is the dielectric constant. As continuum solvation models operate by explicitly representing the solute and implicitly representing the solvent, the solute is thought to be a pocket of space that interrupts the homogeneous dielectric medium of the solvent. Therefore there are two regions, one outside the cavity and one within:

$$\nabla\varepsilon(\vec{r})\nabla\phi(\vec{r}) = -4\pi\rho(\vec{r}) \tag{3.65}$$

This equation is valid under the condition of zero ionic strength. If electrolytes are present in the solvent, the Poisson-Boltzmann equation is instead used:

$$\nabla\varepsilon(\vec{r})\nabla\phi(\vec{r}) - \varepsilon(\vec{r})\lambda(\vec{r})\kappa^2\frac{k_BT}{q}sinh\left[\frac{q\phi(\vec{r})}{k_BT}\right] = -4\pi\rho(\vec{r}) \tag{3.66}$$

Where k_B is the Boltzmann constant, q is the charge magnitude of the electrolyte ions, λ is a switching function: zero being regions inaccessible to the electrolyte and

one otherwise, and κ^2 is the Debye-Hückle parameter:

$$\kappa^2 = \frac{8\pi q^2 I}{\varepsilon k_B T} \tag{3.67}$$

Where I is the ionic strength of the electrolyte solution. Therefore through either Equations 3.65 or 3.66, the electrostatic potential in solution can be calculated and hence polarisation energy can be computed.

Generalised Born Approximation

In practice, modelling the solvation of a molecule is done through the following process:

1. Creation of a 3D grid.
2. Creation of a cavity by assigning gridpoints and the appropriate dielectric constant.
3. Insertion of a discrete version of the solute molecule. The molecule has no charge and so this is computed solely through dispersion interactions.
4. The solute molecule is then charged. Calculation of electrostatic potential at each grid point through the Poisson-Boltzmann equations.
5. Equation 3.63 can be solved through summation over gridpoints.

3.2 Machine Learning[71]

Machine learning at its core is the process of using past data to make predictions of the future or of unknowns. ML models can be broadly split into two types: regression is quantitative, where the model gives a numerical value for its prediction, and classification is qualitative, where the model assigns a category for its prediction. For the purposes of this work, only regression will be considered.

Data that is supplied to the model is known as inputs, that can be split into predictors/features and past prediction values, whilst the predictions the model makes are the outputs or responses. There are two main styles of learning for ML methods: supervised and unsupervised. In supervised learning, the model learns from labelled training data, meaning that the model knows sample values of the quantity it is trying to predict. In unsupervised learning the modelled is fed unlabelled information, there is no supervision, as it were, from the programmer. The model will find its own relationships from the data it is provided. For the purposes of this work, only

supervised learning will be considered.

3.2.1 Test-Train Split

With any machine learning model, it is important that a portion of the available data for building the model is withheld during training. This is to allow for an unbiased testing of the models predictive power; if the model was allowed to train on the data it was tested on, it would return an accuracy much higher than for external values and no evaluation of the model's "real" predictive power would be gained. This is logically referred to as the test-train split. There is no hard and fast rule for how much of the data should be withheld for testing, however values around 70% training data 30% testing data are common.

3.2.2 Prediction Assessment

A fundamental part of developing a machine learning model is testing how strong the predictions are whilst the model only has access to the training set. Without this ability, there is no way to refine the model towards a better prediction. If significantly large enough data was available, ideally three splits would be used: a training set, a validation set and a testing set, where the validation set would be to test how well a model built on the training set was performing. The model could then be modified and tested again on the validation set to see if improvement was made. The testing set, of course, cannot be used for this purpose as it serves to be a completely independent test of the model. Often however, data is limited, one of the simplest solutions to this is error prediction through cross-validation, which follows the general form:

$$Err = E[L(Y, \hat{f}(X))] \tag{3.68}$$

Where Err is the average error when method $\hat{f}$ is applied to an independent test sample from the joint distribution of inputs, X, and output, Y. $[L(Y, \hat{f}(X))]$ is a loss function of the prediction of Y from the X inputs, minimisation of this value means reduction in the error of prediction.

K-fold cross-validation operates by the testing data being split evenly into K sections, for example K=5, as can be seen in Figure 3.3. The model is trained on $K-1$ sections, and then tested on the remaining section, k. This is then repeated, with each K split being used to test the data: $k = 1, 2, \cdots, K$ and then combined with the error estimates. $K = 5$ or $K = 10$ are typically chosen values. $\kappa : \{1, \cdots, N\} \mapsto \{1, \cdots, K\}$ is an indexing function that indicates the parti-

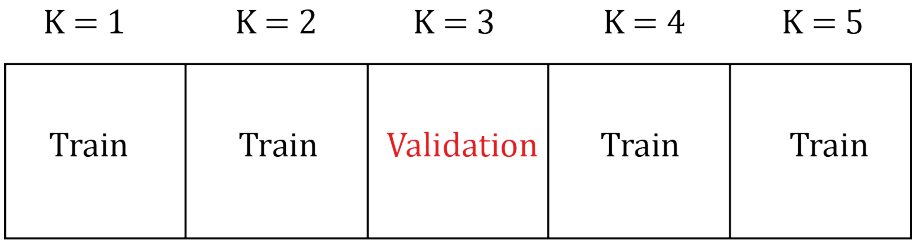

Figure 3.3: Cross validation splitting process and labelling

tion to which observation i is randomly allocated and N the number of input data points. Where $\hat{f}^{-k}$ is the fitted function, computed from the kth section, then cross-validation estimate of prediction error takes the form:

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)) \tag{3.69}$$

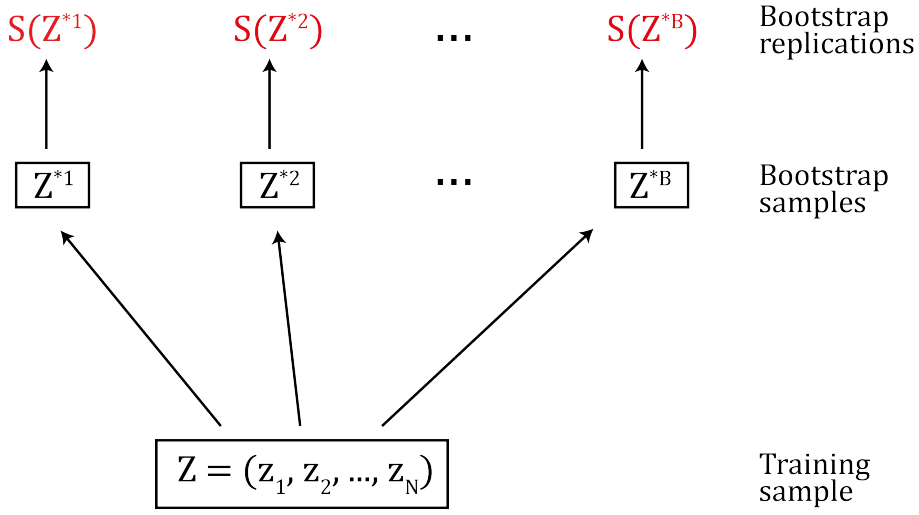

Figure 3.4: Formation of a bootstrap replications from a set of training data.

Another option for testing prediction error is the bootstrap method. Assume the training set of our model is $Z = (z_1, z_2, \cdots, Z_N)$, where $z_i = (x_i, y_i)$, x_i being a constituent input value and y_i a constituent output value. z_i values are selected at random to be a part of a "new" training set of the same size as the original training set. Each z_i can be selected more than once, this is done B times to produce a bootstrap dataset as seen in Figure 3.4. This often results in some z_i being left out of the bootstrap which can be used to test the model. This error can be predicted by:

$$\widehat{Err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i)) \tag{3.70}$$

Where $\hat{f}^{*b}(x_i)$ is the predicted value at x_i. A problem with this approach is that the original training set is being used to test against, and of course, the original and bootstrap sets have shared features. This leads to overfit prediction being over estimated on the positive side. A better method is to only track predictions from bootstrap samples not containing that observation, termed leave-one-out bootstrap error:

$$\widehat{Err} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mid C^{-i} \mid}\sum_{b\in C^{-i}} L(y_i, \hat{f}^{*b}(x_i)) \tag{3.71}$$

Where C^{-i} is the set of indices of samples b that do not contain i, and $\mid C^{-i} \mid$ the number of such samples.

3.2.3 Bagging

Bootstrapping was discussed in Section 3.2.2 in the context of measuring the accuracy of a model, however, it can equally be used to improve the model itself. When used in this context, it is referred to as bagging: where the data the model is trained on is selected randomly from the full dataset, and the same data point can be selected more than once. Assume a model has been fitted to $Z = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ which results in prediction $\hat{f}(x)$. Bagging averages this prediction over a set of bootstrap samples. The model is fitted to each Z^{*b} to gain prediction $\hat{f}^{*b}(x)$. This estimate is therefore given by:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \tag{3.72}$$

Bagging can be shown to reduce the variance of a model, furthermore by introducing duplicate samples randomly, it adds more variance to the dataset which helps reduce overfitting.

3.2.4 Random Forest

Bagging works very well for high-variance, low-bias ML models, such as decision trees. Decision trees operate by fitting the features to simple, branching constraints, as can be seen in Figure 3.5. An algorithm is built that chooses which variable to use to split, and also what value to make the split point at. Assuming we partition into M regions $R_1, R_2, \cdots, R_M$ and model the response in each region as c_m:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{3.73}$$

If the minimisation criterion is the sum of squares $\sum(y_i - f(x_i))^2$, then the best

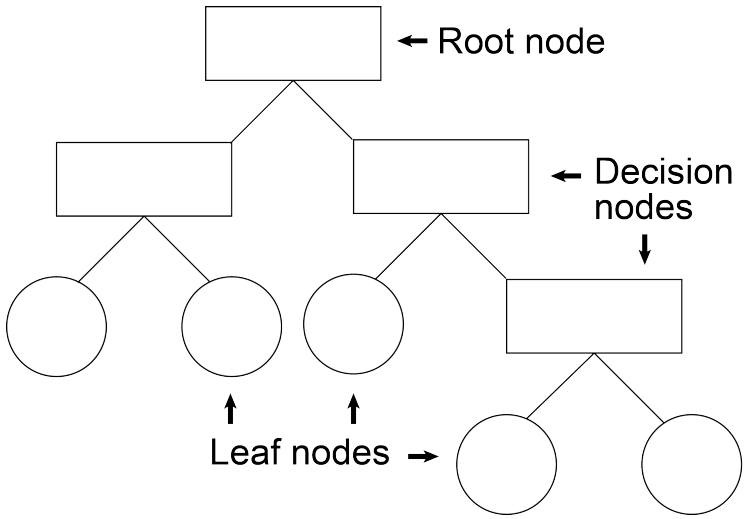

Figure 3.5: Graphical representation of a decision tree model, outlining the structure of root, decision and leaf nodes.

value of $\hat{c}_m$ is:

$$\hat{c}_m = ave(y_i \mid x_i \in R_m) \tag{3.74}$$

Where *ave* is the average. Finding the best binary partition is performed through a greedy algorithm. If j is a splitting variable and s a split point:

$$R_1(j,s) = \{X \mid X_j \leq s\} \; and \; R_2(j,s) = \{X \mid X_j > s\} \tag{3.75}$$

Then, find j and s that solve:

$$\min j, s \left[\min c_1 \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min c_2 \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \tag{3.76}$$

And the inner minimisation is solved by:

$$\hat{c}_1 = ave(y_i \mid x_i \in R_1(j,s)) \; and \; \hat{c}_2 = ave(y_i \mid x_i \in R_2(j,s)) \tag{3.77}$$

The best choice of s is computationally very fast, and therefore iterating to find the optimal (j, s) pair is feasible. After the first split is completed, the process can be repeated for each of the consequent branches. If this were to continue on indefinitely it would result in a grossly overfitted model, therefore tree size is a tuning parameter that controls the number of branches in the model. This is usually performed by stopping the growth of the tree after some minimum node size is reached. Other tuning parameters exist:

1. Total number of trees in the forest.

2. Minimum number of samples required to split each node.

3. Minimum number of samples required for leaf node formation, which has a similar effect on tree size.

Random forest is a modification of the bagging process that builds a large "forest" of unrelated decision trees and makes an average prediction from the ensemble of trees. The reason that this is an effective process is that it allows the averaging of many noisy but mostly unbiased models. Decision trees are intrinsically noisy models and so benefit from averaging.

A Random Forest model is generated by creating a bootstrap sample from $b = 1, 2, \cdots, B$, with a $Z*$ the same size as N from the training data. A decision tree T_b is then grown from each b by recursively repeating the following for each terminal node until the minimum node size n_{min} is reached:

1. Choose m variables randomly from the potential variables
2. Calculate the best (j, s) amongst m
3. Form two daughter nodes

The output of the ensemble of trees is then $\{T_b\}_1^B$. A prediction at point x can then be made:

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{3.78}$$

3.3 Molecular Descriptor Calculation[72]

Molecular descriptors are defined as the "final result of a logical and mathematical procedure, which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardised experiment".[73] These are necessary when it comes to feeding molecular information to a computer system to build predictions as a computer can only understand numerical information. Some examples of molecular descriptors can be seen in Figure 3.6.

Molecular descriptors are usually classified by the number of dimensions used to find the descriptor. 0D descriptors can be found knowing only the atom composition of the molecule, e.g. molecular weight, number of x atoms, number of bonds etc. 1D descriptors take into account how these atoms are bonded together to give fragment counts, e.g. number of terminal x atoms, number of x functional groups etc. 2D descriptors evaluate topological factors such as shape indices, topological polar surface areas etc. 3D descriptors consider the geometrical shape of the molecule, e.g.

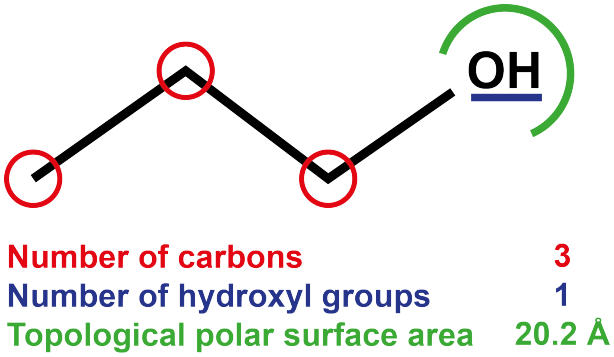

Figure 3.6: Depiction of some general molecular descriptors for a propanol molecule

radius of gyration, radial distribution function etc. So called 4D descriptors take into account 3D coordinates plus a sampling of conformations of the molecule in question.

There are various software that have been developed for the calculation of molecular descriptors. DRAGON[48] is perhaps the largest and most well known, claiming the ability to calculate over 5000 descriptors, however, it is a proprietary software. The software used in this work was Mordred,[74] a more recent development having been released in 2018. This python-based software operates by taking the SMILES[75] string input which contains the atoms and their molecular arrangement, then computing and outputting up to 1825 descriptors, if both 2D and 3D are calculated.

Chapter 4

Computation Method

4.1 Database

A database of independent experimental logP, hydration free energy and solvation free energy values was required. The logP values were sourced from the Sangster database,[3] the hydration free energies from the FREESOLV database,[76] and the solvation free energies from the Minnesota Solvent database,[77] all of which had their experimental values measured at 298.15 K. Only molecules which had an entry in all three databases were kept, in the end this resulted in 107 molecules, see the Appendix for information on acquiring this data.

4.2 First Principles Calculations

As can be seen from Equation 4.1, logP can be calculated from the gas constant (R), temperature (T) and the Gibbs free energy of transfer between octanol and water (ΔG_{OW}). This value can be difficult to calculate directly through QM methods, instead the Gibbs free energy of hydration (ΔG_H) and the Gibbs free energy of solvation (ΔG_S) can be found, as $\Delta G_{OW} = -(\Delta G_S - \Delta G_H)$. This relationship can be seen in Figure 4.1.

$$logP = -\frac{\Delta G_{OW}}{RT(ln10)} \tag{4.1}$$

$$logP = -\frac{(\Delta G_S - \Delta G_H)}{RT(ln10)} \tag{4.2}$$

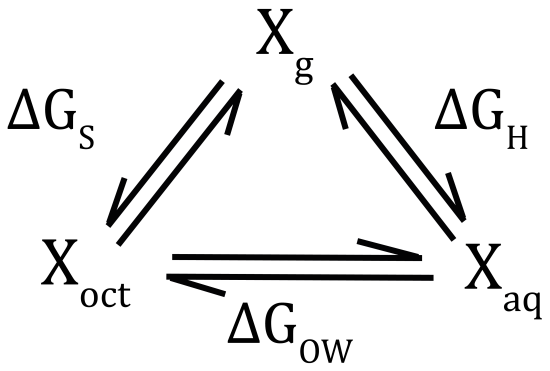

Figure 4.1: Thermodynamic energy cycle involving Gibbs free energy of transfer between octanol and water (ΔG_{OW}), Gibbs free energy of hydration (ΔG_H), and Gibbs free energy of solvation (ΔG_S)

4.3 Quantum Mechanical

The QM calculations were performed using the Gaussian 16 software.[78] The input files used were supplied with the Minnesota Solvent Database. Geometry optimisation and energy calculation was performed for each molecule using all combinations of the HF,[79] M11,[80] & B3LYP[81,82] methods and the 6-31G(d), 6-311G(d), 6-311G(d,p), & 6-311+G(d,p)[83] basis sets. Each molecule was geometry optimised in the gaseous state, the aqueous state, and in octanol for each level of theory. Where solvents were involved, the continuum SMD[84] was used. LogP values were calculated from the difference in energy in octanol and water. Hydration and solvation free energies were calculated from the difference in energy in gas and the respective solvent.

4.4 Molecular Descriptors

Molecular descriptors for use in the ML models were computed through use of the Mordred program.[74] The SMILES strings were supplied to the program and 2D molecular descriptors were requested. For some descriptors there were molecules in the dataset that had a N/A value, in these cases the descriptor was removed for all molecules. This left 1122 molecular descriptors per molecule.

4.5 Machine Learning

Implementation of the Random Forest models was accomplished through use of the sci-kit learn[85] python package. The data was split into training and testing data through a 75/25 split, and the results were averaged over 100 of these splits to reduce

the variability of the split impacting on the prediction results. A reproducibility seed was used to allow for fair comparison between ML models, it ensures that for a given seed the molecules in the testing and training sets are the same and the produced Random Forest is identical. This is needed so when adding extra model complexity such as optimisation, the new results are directly comparable to the old. The seed used was "121212", for each test train split the seed was multiplied by the number of the split i.e. split 1 had seed "121212", split 2 had seed "242424", etc. Without this measure all test train splits would contain the same molecule split. Each model had 500 trees in the forest, this value has been found in literature to be sufficient for most models,[86] and gives a good ratio of accuracy against computational expense.

Hyperparameter optimisation was added to the model, this could have been implemented in one of two ways. The optimisation can be done once for all the data, or optimisation can be done after every test train split. In theory, optimising for each specific split should give a marginally better improvement to the performance of the model so this was chosen. However, it should be noted that this style of optimisation comes with a greatly increased computational expense, with 100 more optimisations being performed in the case of this model. Optimisation was performed with five-fold cross validation through a grid search, the hyperparameters tested were: maximum features to consider when splitting a node, maximum node depth, minimum samples required to split a node, and minimum samples to form a leaf node.

Recursive feature elimination was also performed with five-fold cross validation for each test train split. Originally, features were removed one-by-one, this allowed for all features to be ranked from best to worst. However, this proved to be a too computationally expensive approach. Instead, 100 features were removed at a time until at least 400 remained, then 25 until at least 200 remained, then 10 until at least 100 remained, then features were removed one-by-one. This is acceptable as the features removed early in the feature elimination are of least importance, and therefore are not of much interest in understanding the model. Tests were performed to ensure that there were not features of significant performance above 100 remaining features. In practice, more than the minimum number of 100 features remained before the one-by-one feature removal occurred. The full script including the optimisation and RFE can be found in the Appendix.

For each test train split, the R^2, RMSE, bias, and SDE were calculated. These error values were averaged over all 100 splits to find the overall error for R^2, RMSE, bias, and SDE. To find the ML model's prediction of logP for individual molecules,

the predictions made on the unseen test molecules were recorded for each test train split and the average of these was taken as the overall ML model prediction for that molecule. This was done to allow the logP prediction for a molecule to be plotted against the experimental value for the molecule. The random nature of the test train split meant that some molecules were in the testing set more often than others.

Chapter 5

Results and Discussion

5.1 Database

After merging the three experimental databases and keeping only molecules that had a entry in each database (i.e. a hydration energy, a solvation energy, and a logP value) a total of 107 molecules remained. MW and logP distributions for the molecules can be found in Figure 5.1 and 5.2 respectively. The atomic makeup of the dataset included carbon, hydrogen, oxygen, nitrogen, sulphur, chlorine, and bromine atoms. The types of molecules included are alkanes, alkenes, alkynes, aromatics, alcohols, phenols, ethers, furans, carboxylic acids, sulphides, etc. Although there is a decent variety in functional groups and molecule types, overall, the molecules in the database can be seen in Figure 5.1 to be rather small in terms of size, when compared to more drug-like molecules which tend to have MWs of around 200 to 600 Daltons.[87] Predictions of drug-like molecules would give more practical results to the thesis as those are molecules which logP prediction are of industrial interest. The problem is two-fold, firstly the requirement of having three separate experimental data points. Whilst there is a plethora of logP freely accessible for many small and large compounds, hydration and solvation experimental free energies are more scarce, especially for larger molecules. Secondly, a decently large database is required to create a well-performing ML model. These two factors together resulted in having a database of smaller, less drug-like molecules.

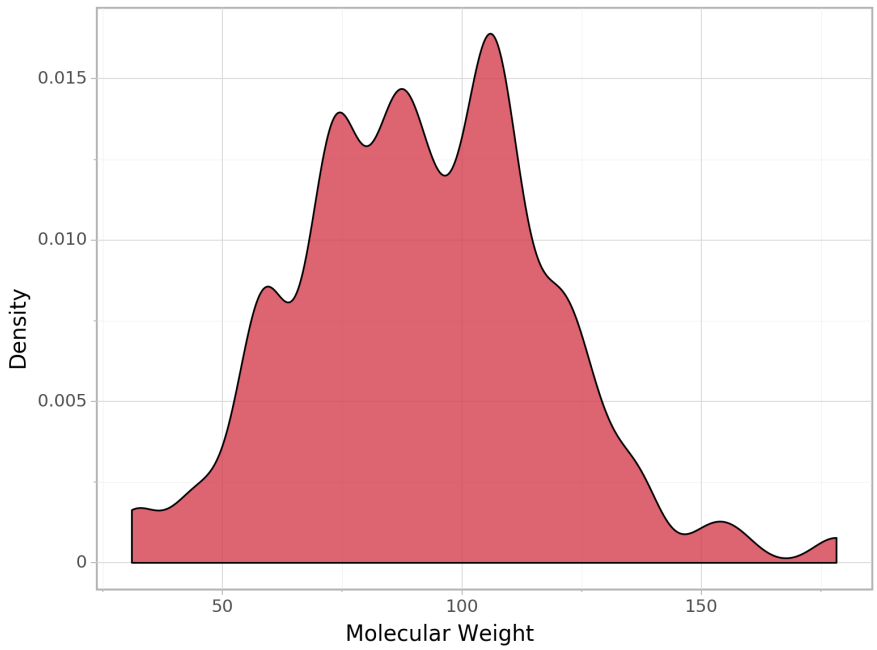

Figure 5.1: MW distribution of the 107 database molecules, MWs are measured in Daltons.

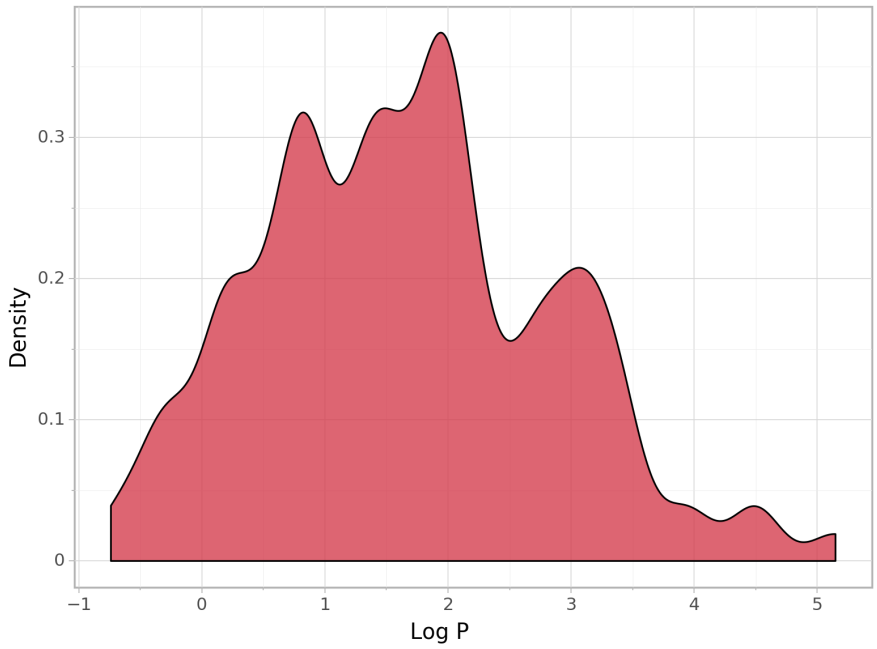

Figure 5.2: LogP distribution of the 107 database molecules

A comparison of MW against logP values for the molecules in the database was ran, to see if a trend would be apparent, which is presented in Figure 5.3. Indeed, it would appear that as MW of a molecule increases, so too does its logP value. Therefore as molecules in the dataset increase in size, they tend to increasingly favour the octanol phase, indicating an increase in lipophilicity. This could be explained by

an increase in aliphatic chain size of larger molecules, as octanol itself has a long aliphatic chain, therefore like dissolving in like.

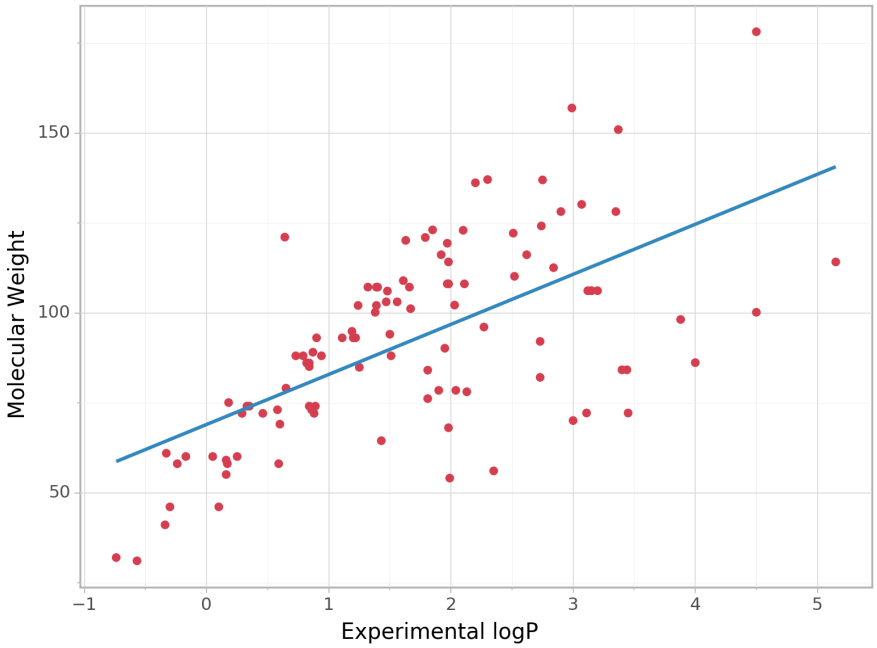

Figure 5.3: Trend of logP against MW, showing that logP increases with increasing MW.

5.2 Calculation via Experimental Free Energies

As experimental values of hydration and solvation free energies were obtained, computation of logP from the thermodynamic energy cycle (Section 4.2) could be calculated. The results of these calculations can be seen in Table 5.1 and Figures 5.4, 5.5, & 5.6. As can be seen from the graphs and errors, calculation from experimental free energies gave highly precise and accurate predictions for logP, hydration free energies and solvation free energies. The bias is low and equally overestimates and underestimates, combined with the low SDE we can conclude that with any two of the experimental values, the third can be almost exactly calculated, at least with this dataset.

Table 5.1: Errors for calculated logP, hydration free energies and solvation free energies from experimental data. Hydration free energy (ΔG_H) and solvation free energy (ΔG_S) are in kcal/mol and all values calculated at 298.15 K

Type	R^2	RMSE	Bias	SDE
LogP	0.993	0.101	0.066	0.097
Hydration	0.996	0.138	0.090	0.130
Solvation	0.994	0.138	0.090	0.130

The highly accurate nature of these calculations raised suspicions on the independency of the gathered experimental data: did one of the databases calculate their experimental values through use of the other two experimental values? If this was the case, it would explain the accuracy of my own calculations, as the data would no longer be independent. All three of the databases state that their values are experimentally generated, and on inspection of each database's sources this could not be disproved. As far as I am aware all experimental values are indeed independent.

If calculation of logP can be done so exactly through first principles calculation and experimental hydration and solvation free energy values, it leads to questioning why this is not the default method for experimental evaluation of logP. I believe this is a matter of practicality and not accuracy. Experimental hydration and solvation free energies are difficult to measure experimentally,[88] more so than the logP shake-flask method due to the very low vapour pressures most organic molecules exhibit. Furthermore, this would require the measurement of two independent experimental values, therefore doubling the workload when compared to the shake-flask method. The limiting factor in the discovery of experimental logP values is not the availability of compounds to be tested, but in the rate they can be measured. This is why we see the use of the shake-flask method instead.

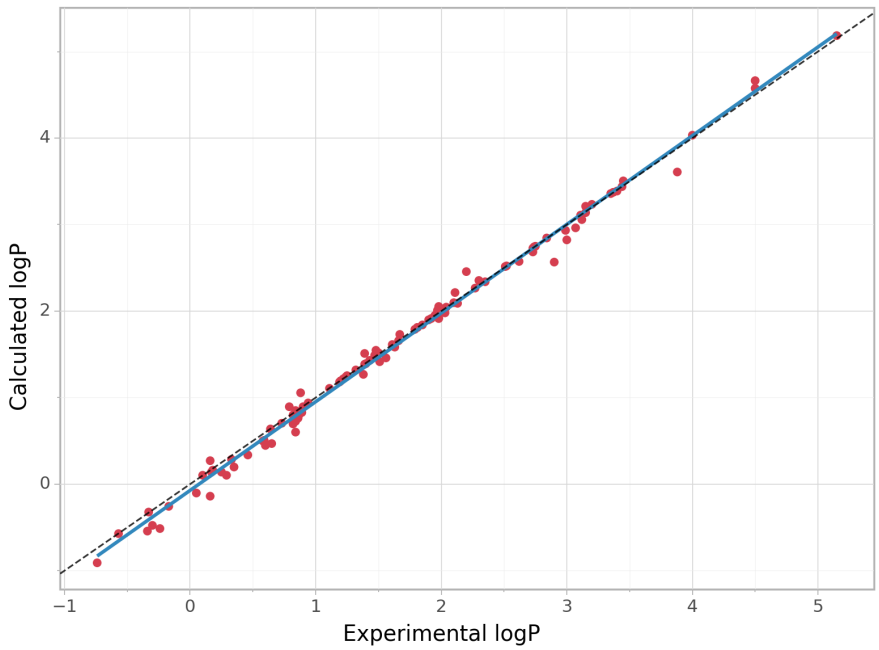

Figure 5.4: LogP calculated via experimental hydration and solvation free energies against experimental logP, where the black dotted line (y=x) represents perfect prediction. Calculated at 298.15 K

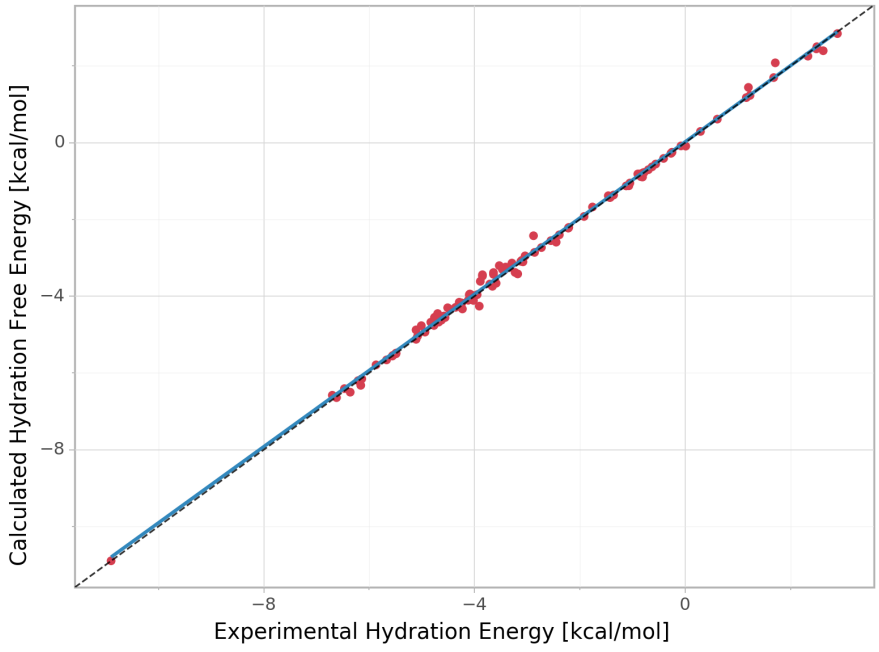

Figure 5.5: Hydration free energy calculated via experimental values. Hydration free energy (ΔG_H) is in kcal/mol and calculated at 298.15 K

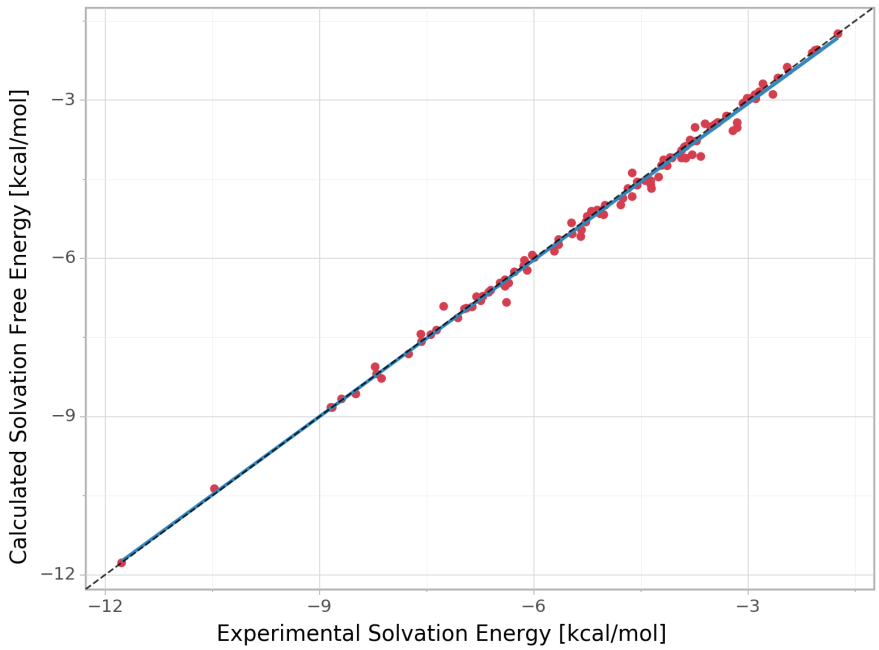

Figure 5.6: Solvation free energy calculated via experimental values. Solvation free energy (ΔG_S) is in kcal/mol and calculated at 298.15 K

5.3 Quantum Mechanical

The results for the calculation of logP through QM methods can be seen in Table 5.2, these were obtained from the octanol and water QM calculations. As can be seen from both the table and from Figure 5.7, the method used makes a small but noticeable difference to the accuracy of the prediction when using the 6-31G(d) basis set. Under this condition, the hybrid functionals perform worse than HF, though it should be noted this is a very slight difference that is well within the SDE of all three methods. The prediction between the three methods can be seen to be tight for most values, however for some values a more noticeable spread can be observed. In general for those cases, B3LYP tends to over-predict logP values compared to the other two methods, whilst HF and M11 are over and under predicted equally. Despite the seemingly systematic over-prediction of the B3LYP functional, it still performed slightly better than the M11 functional. As the complexity of the orbital representation increases, we observe that the choice of method makes less and less impact on the error, until at the most complex basis set, 6-311+G(d,p), there is almost no difference in both R^2 and RMSE between the three methods.

Table 5.2: Errors of calculated logP values via QM means, for combinations of HF, M11, & B3LYP methods, with 6-31G(d), 6-311G(d), 6-311G(d,p), & 6-311+G(d,p) basis sets. Calculated at 298.15 K.

Theory	R^2	RMSE	Bias	SDE
HF/6-31G(d)	0.876	0.427	0.357	0.299
HF/6-311G(d)	0.869	0.438	0.369	0.302
HF/6-311G(d,p)	0.874	0.430	0.363	0.298
HF/6-311+G(d,p)	0.880	0.418	0.354	0.293
M11/6-31G(d)	0.867	0.440	0.369	0.304
M11/6-311G(d)	0.867	0.440	0.369	0.304
M11/6-311G(d,p)	0.872	0.433	0.363	0.302
M11/6-311+G(d,p)	0.881	0.418	0.352	0.294
B3LYP/6-31G(d)	0.869	0.438	0.357	0.310
B3LYP/6-311G(d)	0.869	0.438	0.357	0.310
B3LYP/6-311G(d,p)	0.866	0.442	0.359	0.313
B3LYP/6-311+G(d,p)	0.880	0.419	0.343	0.302

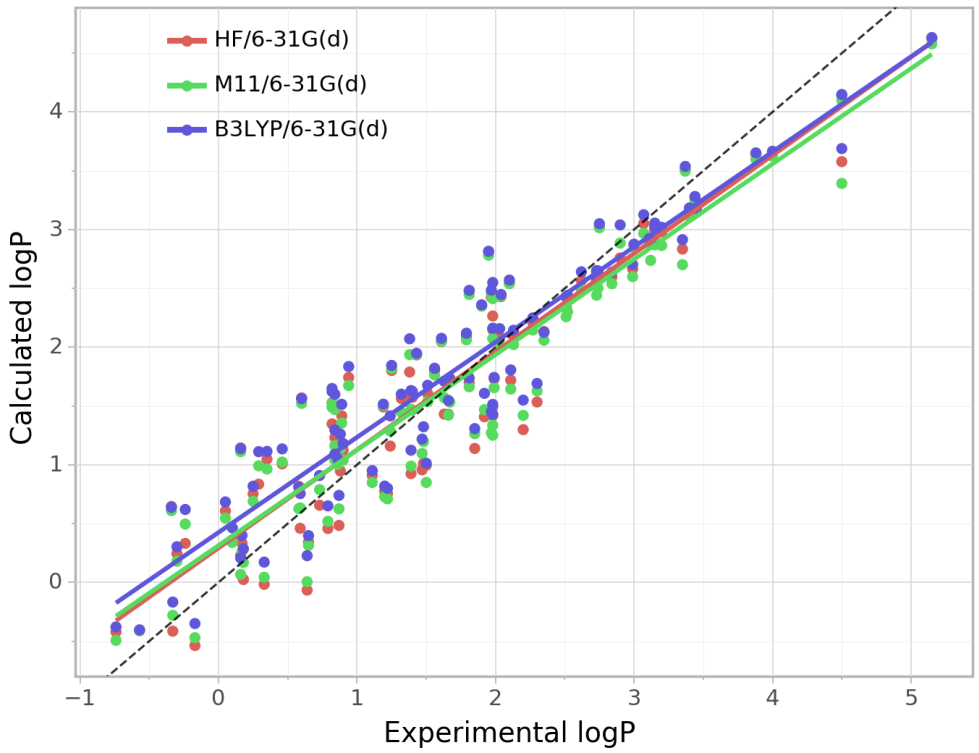

Figure 5.7: Experimental logP values against QM calculated values for the HF, M11, & B3LYP methods with the 6-31G(d) basis set. 6-31G(d) was chosen as results vary the most across all three QM methods with this basis set, and therefore the most graphically noticeable difference. Calculated at 298.15 K.

The basis set appears to have a more noticeable impact on the accuracy of logP prediction. Across the two hybrid functionals, it can be seen that as the complexity of the basis sets increases, the accuracy of logP predictions also increases. This is true for all cases except going from 6-31G(d) to 6-311G(d), in this case the QM calculated free energies in the octanol and water phases are exactly the same for these basis sets. What we can conclude from this is either: for my molecules in solution, these hybrid functionals do not benefit from the increase to the triple-zeta and the additional orbital flexibility it brings, or there was a bug in the software that could not be traced. When the polarisation of the hydrogen atoms is included, we observed a noticeable increase in accuracy. The largest increase in accuracy can be seen when the addition of the non-hydrogen diffuse function is included. Adding a diffuse term to the hydrogen atoms was also considered, however by this point it was clear that changes to the methods and basis sets were making negligible improvements to the calculation of logP. When comparing all basis sets, it can be inferred that the 6-311+G(d,p) is the best performing, however with the small differences experiences this can be not conclusively proven.

When observing the effect of basis set complexity on the HF calculations however, the results are not as expected. The inclusion of the triple-zeta worsens the prediction errors, this is slightly improved by the inclusion of hydrogen atom polarisation however the prediction is still worse than the 6-31G(d) basis set. Finally, when the polarisation function is added, the HF results fit perfectly with the other methods. Although this trend can be observed, it should be noted that errors only slightly worsen and well within the SDE, it is possible that this is merely a statistical anomaly. If this is not the case, it would appear that the HF method struggles more than the other methods without use of the larger orbital that the polarisation function supplies. Figure 5.8 shows how logP predictions varied as the basis set changed for the HF method, when compared to change due to methods (Figure 5.7) the change per atom is less noticeable. It can be observed that the 6-311G(d) function was the most likely to under-predict relative to the other basis sets, this systemic error perhaps explains why it had the least accurate prediction error at this level of theory. The 6-311+G(d,p) basis set had the trend line with the closest gradient to 1. This gives confidence to the computed errors, 6-311+G(d,p) does indeed appear to give the most accurate results.

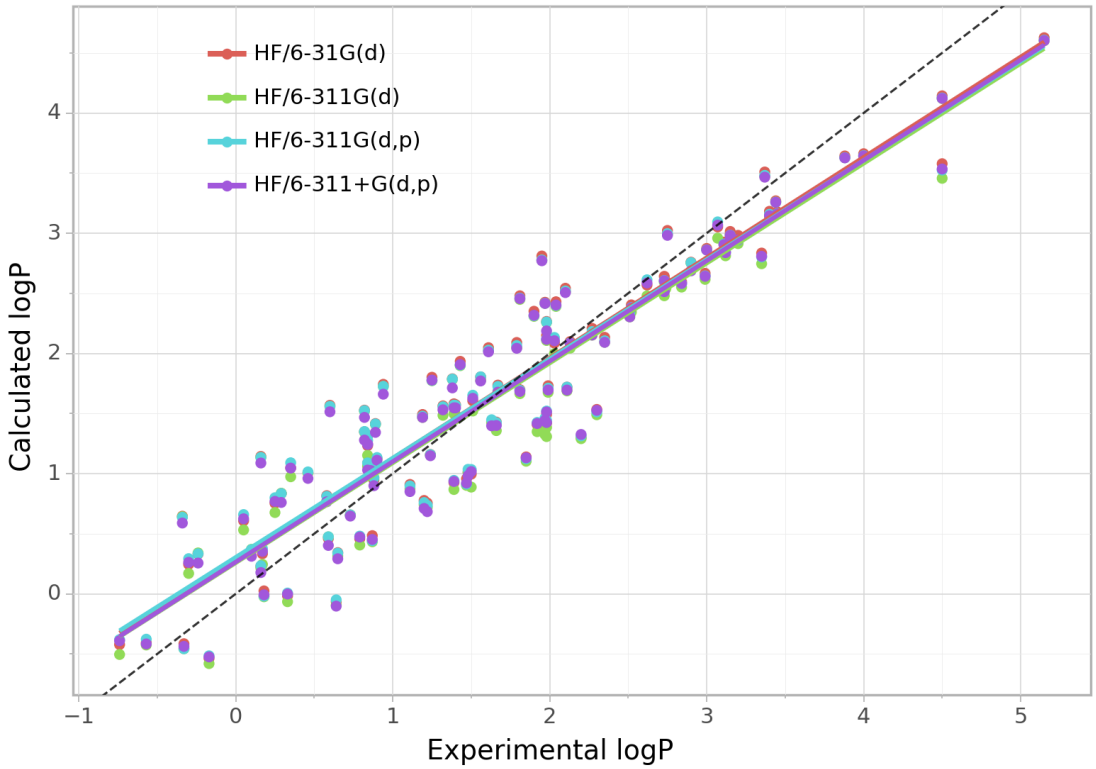

Figure 5.8: Experimental logP values against QM calculated values for the HF method with the 6-31G(d), 6-311G(d), 6-311G(d,p), 6-311+G(d,p) basis sets. The HF method was chosen as it experiences the greatest deviance across all four basis sets and therefore the most graphically noticeable difference. Calculated at 298.15 K.

When comparing across all the trial methods and basis sets, the M11/6-311+G(d,p) is the most accurate. However, the HF and B3LYP methods are also very comparable when using the 6-311+G(d,p) basis set. Although M11/6-311+G(d,p) can be concluded to be the most accurate, it is also one of the most computationally expensive options tested. When comparing the improvement the M11/6-311+G(d,p) level of theory makes over the HF/6-31G(d), the improvement is only very marginal. Therefore, the best level of theory is debatable, there is a marginal increase in accuracy going from HF/6-31G(d) to M11/6-311+G(d,p), which comes with a marginal increase in computational time. I conclude that if you have a small number of molecules that you wish to compute the logP values of, I would recommend the M11/6-311+G(d,p) level of theory, especially if you want to use the values in further calculations or models where accurate input values matter. However, if your aim is to find logP values over a large search space, such as finding potential drug molecules, the benefit of increased accuracy is less important and a faster speed of calculation is a far more important factor. This is all under the caveat that, as discussed in Section 5.1, my database of molecules have low MWs. One would expect

that as the size of the molecule increases, the gap in prediction between the two levels of theory would increase, at that point the sacrifice in accuracy may be great enough to warrant the computational expense.

Figure 5.9 shows the predicted logP values when using the M11/6-311+G(d,p) level of theory, divided into groups. It is observed that some groups have very linear predictions that run parallel to the $y = x$ line, namely alkanes, alkenes, carboxylic acids, and cycloalkanes. In these cases all prediction members belong to very similar chemical families, taking the carboxylic acids for example, the predicted values are for aceticacid, propanoicacid, butanoicacid, pentanoicacid, & hexanoicacid: these molecules only differ by the length of the aliphatic chain. In examples like the carboxylic acids, where the aliphatic chain is increased by one carbon unit at a time, logP increases by a discrete factor with each addition. This could allow for the possibility for predicting other members of the family without having to run the QM calculations. This would be most useful in cases where large search spaces of molecules were to be predicted, in theory you could skip the calculation of x many molecules in the family, i.e. calculate methane, propane, & pentane, and then use those values to estimate ethane & butane. There is also the possibility of using a correction factor for these groups that is specific to each chemical family, in order to bring results closer to experimental. The ketone group is a good example of how this linear prediction line breaks down when molecules are not very closely related, the linear line belongs to acetone, 2-butanone, 2-pentanone, 2-hexanone, & 2-heptanone, whilst the outlier is acetophenone. The use of skipping molecules in a family and the use of a correction factor that is family-specific starts to lose practicality when considering larger, drug-like molecules that are complex in structure and possess multiple functional groups.

Another interesting trend that can be seen from Figure 5.9 is that aromatic compounds seem to be systematically under-predicted, especially aromatic compounds with added functional groups. These systems allow for electrons to move far away from their nucleus, this could signify that the orbital representation used in the 6-311+G(d,p) is not large enough to correctly model these molecule types. Whilst this is not always the case, some predictions in the aromatic amine group can be seen to be over-predicted, this knowledge could allow for a slight indiscriminate correction factor to be applied to all aromatic/aromatic+functional group compounds, to bring results closer to experimental values. If this trend exists for the aromatics, it is very possible that it exists for other groups, discovery of more of these systematic trends could allow for computationally inexpensive accuracy increases.

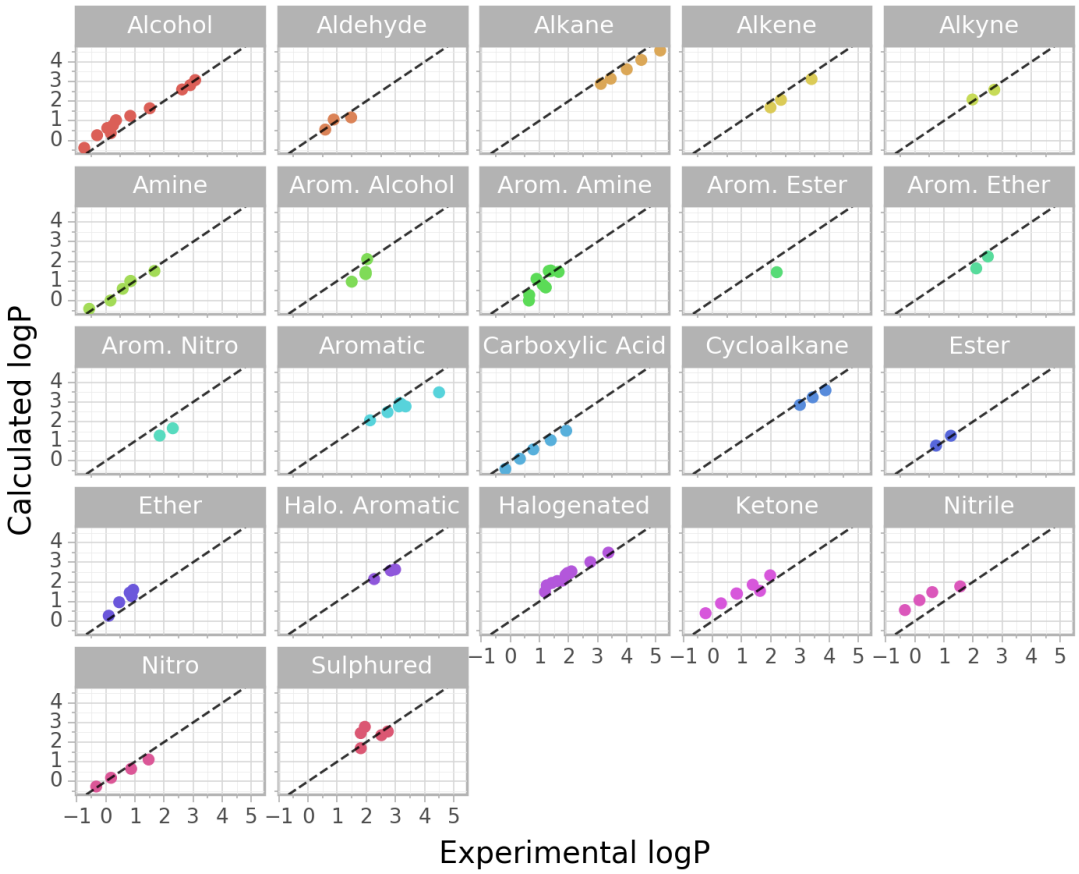

Figure 5.9: Experimental logP values against QM calculated values for the M11/6-311+G(d,p) level of theory, separated by general molecule type. Calculated at 298.15 K.

Table 5.3: Errors of calculated hydration free energy values via QM methods, for combinations of HF, M11, & B3LYP methods and 6-31G(d), 6-311G(d), 6-311G(d,p), & 6-311+G(d,p) basis sets. No values are given for the hybrid functionals with the 6-31G(d) basis set due to technical failure of the software. Hydration free energy (ΔG_H) is in kcal/mol and calculated at 298.15 K

Theory	R^2	RMSE	Bias	SDE
HF/6-31G(d)	0.853	0.896	0.654	0.468
HF/6-311G(d)	0.762	1.139	0.927	0.279
HF/6-311G(d,p)	0.814	1.007	0.730	0.474
HF/6-311+G(d,p)	0.705	1.267	0.946	0.373
M11/6-31G(d)	N/A	N/A	N/A	N/A
M11/6-311G(d)	0.810	1.018	0.888	0.229
M11/6-311G(d,p)	0.895	0.754	0.622	0.367
M11/6-311+G(d,p)	0.797	1.052	0.913	0.218
B3LYP/6-31G(d)	N/A	N/A	N/A	N/A
B3LYP/6-311G(d)	0.952	0.510	0.398	0.352
B3LYP/6-311G(d,p)	0.925	0.638	0.527	0.361
B3LYP/6-311+G(d,p)	0.935	0.596	0.457	0.387

The results of the QM-calculated hydration and solvations free energies can be found in Tables 5.3 & 5.4 respectively. The hydration free energies were found from the water and gas calculations, and the solvation free energies from the octanol and gas calculations. The most immediately noticeable observation is the failure of the 6-31G(d) basis set for hybrid functions at predicting free energy. This error is held completely within the gaseous calculations, systematically for each molecule the gaseous energy calculations is roughly off by a factor of 10. This is interesting as no such error exists for the molecules calculated in solvents for those levels of theory. When calculating logP, the gas phase energies cancel out, this allows logP to be calculated accurately which can be seen in Table 5.2. The error is far greater when bromide atoms are considered, with predictions that are off by a factor of 10,000. This fact by itself is explainable: the bromide atom is too large for the 6-31G(d) basis set to accurately represented. What is strange is how this problem does not appear anywhere near to the same extent when calculating values in solvent. I believe that the molecules in the dataset should be able to be correctly predicted at these levels of theory, therefore it is concluded that these results are due to a bug with the QM software used that could not be tracked down.

Table 5.4: Errors of calculated solvation free energy values via QM methods, for combinations of HF, M11, & B3LYP methods and 6-31G(d), 6-311G(d), 6-311G(d,p), & 6-311+G(d,p) basis sets. No values are given for the hybrid functionals with the 6-31G(d) basis set due to technical failure of the software. Solvation free energy (ΔG_S) is in kcal/mol and calculated at 298.15 K

Theory	R^2	RMSE	Bias	SDE
HF/6-31G(d)	0.626	1.094	0.858	0.359
HF/6-311G(d)	0.512	1.251	0.992	0.266
HF/6-311G(d,p)	0.569	1.176	0.894	0.376
HF/6-311+G(d,p)	0.373	1.418	1.067	0.037
M11/6-31G(d)	N/A	N/A	N/A	N/A
M11/6-311G(d)	0.578	1.164	0.980	0.204
M11/6-311G(d,p)	0.686	1.003	0.810	0.347
M11/6-311+G(d,p)	0.481	1.290	1.031	-0.060
B3LYP/6-31G(d)	N/A	N/A	N/A	N/A
B3LYP/6-311G(d)	0.762	0.873	0.704	0.377
B3LYP/6-311G(d,p)	0.738	0.916	0.743	0.364
B3LYP/6-311+G(d,p)	0.662	1.042	0.838	0.214

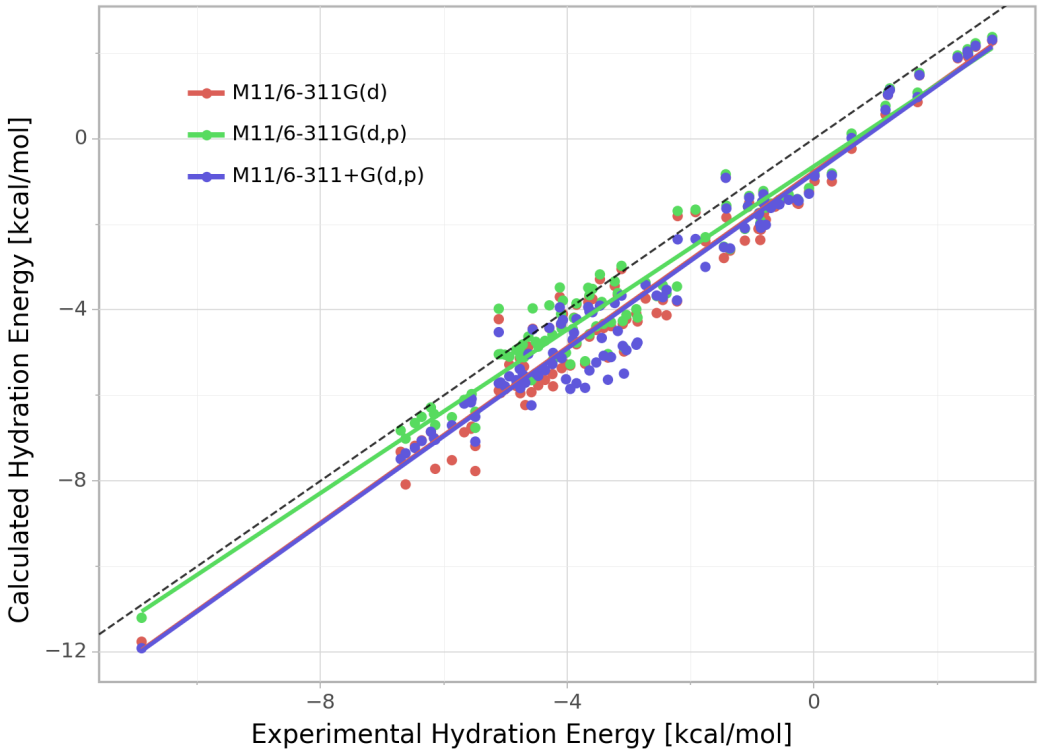

Figure 5.10: QM-calculated hydration free energies for the M11/6-311+G(d,p) level of theory. Hydration free energy (ΔG_H) is in kcal/mol and calculated at 298.15 K

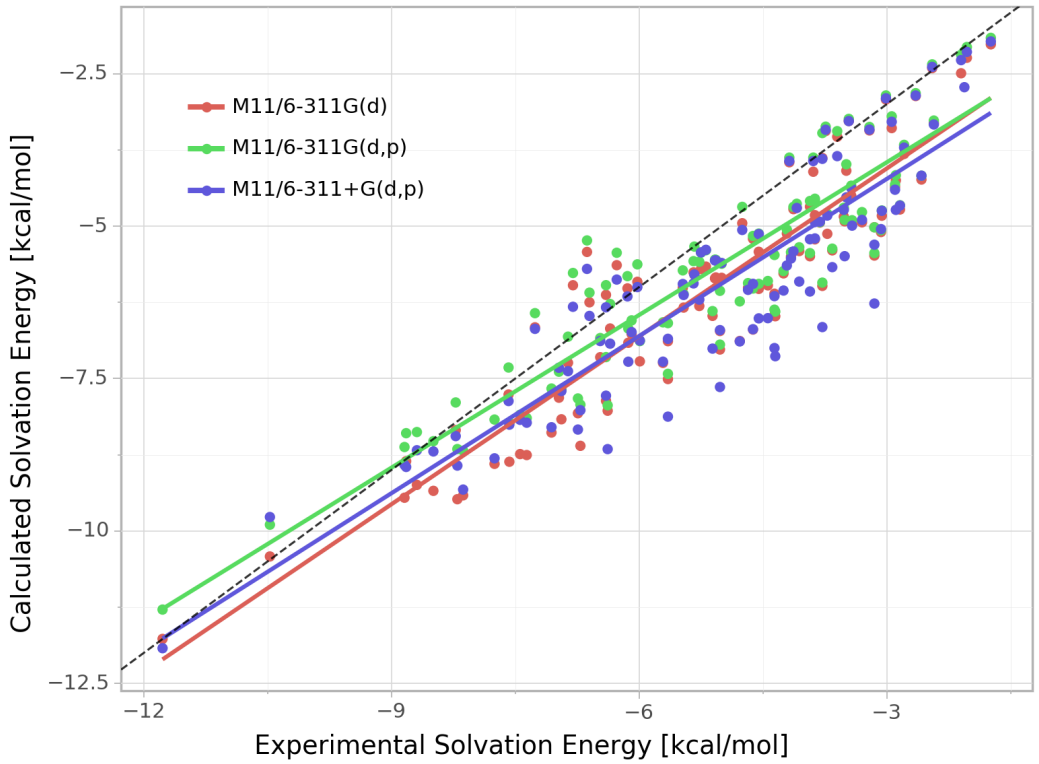

Figure 5.11: QM-calculated solvation free energies for the M11/6-311+G(d,p) level of theory. Solvation free energy (ΔG_S) is in kcal/mol and calculated at 298.15 K

Comparing the hydration and solvation predictions, it can be observed that across the levels of theory, hydration free energies are calculated substantially more accurately than the solvation free energies. This is as expected, water is a highly important molecule to computational chemistry, and indeed chemistry as a whole, therefore correct modelling of water as a solvent has been extensively developed in comparison to octanol. Another factor that would lead to worse predictions for solvation free energy is that in the partition of octanol and water, the octanol layer is termed "wet" as there is a significant amount of water present, 27% by mole fraction.[89] However, in my QM calculations the octanol solvent is modelled as pure, this could be remedied by the addition of explicit water molecules around the solute. This was not attempted due to the greatly increased computational expense such a modification would cause. From this observation, we can conclude that the error in our logP calculation comes from the solvation free energy more so than the hydration free energy. To further improve the QM logP predictions, finding a model that represents octanol more realistically would be a good first priority. Comparing the three methods, it can be seen that B3LYP is the best-performing for both hydration and solvation free energy across every basis set. The B3LYP hydration free energy calculations are especially good, having a RMSE of approximately half of the other methods with equivalent basis sets. Interestingly, the change in errors for the different levels of theory are far more significant for the free energies calculations than for the logP calculation seen in Table 5.2. Furthermore, these errors are higher than the logP calculation, this is strange as you would expect the logP calculation to inherit both the hydration and solvation errors. LogP is calculated through the difference of hydration and solvation free energies, this suggests that there is some systematic error held within the free energy calculations which is removed by cancellation of errors. Viewing Figures 5.10, & 5.11, we can see this is indeed the case, there is a strong negative bias to both sets of calculations. These graphs also help explain the free energy trend in the basis sets. With the logP calculations, as basis set complexity increased there was a steady increase in the accuracy of finding logP. However, overall for the free energy calculations the most complex basis set 6-311+G(d,p) was the worst performing. Since logP is calculated from the difference of the hydration and solvation free energy, it seems intuitive that the more accurate these free energy values are the better the logP prediction. However, the results prove otherwise. Although, as can be seen in Figure 5.10, the M11/6-311G(d,p) trend line is much closer to the line of $y = x$ than for M11/6-311+G(d,p). Observing the errors in Table 5.3, we can see that the more complex basis set has a lower SDE, despite its much higher bias value. Therefore it can be concluded that QM free energy predictions do not need to be accurate in relation to experimental values. The more important factors are that the predictions have a low SDE, and that if the predictions are shifted via a

large bias, that the bias is consistent between both the hydration and solvation free energies. This is why even though B3LYP was much more accurate at calculating experimental hydration and solvation free energies compared to the M11 method, it did not predict logP with more accuracy. One further observation that can be found from Figures 5.10 & 5.11 is the characteristic slope of the logP prediction trendline, where low logP values are over-predicted and high values under-predicted, comes mostly from the solvation calculations.

Table 5.5: QM-calculated logP, hydration free energy and solvation free energy with correction factor applied errors for the M11/6-311+G(d,p) level of theory. Hydration free energy (ΔG_H) and solvation free energy (ΔG_S) are in kcal/mol and all values calculated at 298.15 K

Theory	R^2	RMSE	Bias	SDE
Hydration Original	0.797	1.052	0.913	0.218
Hydration Corrected	0.948	0.563	0.427	0.380
Solvation Original	0.481	1.290	1.031	-0.060
Solvation Corrected	0.780	0.900	0.729	0.369
LogP Original	0.881	0.418	0.352	0.294
LogP Corrected	0.866	0.436	0.349	0.314

This clear and one directional shift in the bias, in combination with a small SDE, leads to the possibility of applying some form of correction factor to the calculated free energy values, to see if this leads to a better prediction of logP. The most simple form of correction would be to set the c value in the $y = mx + c$ line of the QM predictions to the same as the c value of the experimental hydration free energy $y = mx + c$ line. As there was no training/testing data split, the correction factor would be applied to the data it was trained on, therefore the corrected values cannot be deemed valid. This was applied to the M11/6-311+G(d,p) level of theory as an initial test to observe if prediction correction could be of benefit, and the results can be found in Table 5.5 and Figures 5.14, 5.12, & 5.13. For the hydration free energy this form of correction leads to a far better prediction, due to the trend line and line of $y = x$ having approximately equal slopes. For the solvation free energy we can see from the errors a noticeable improvement to the predicted values, however this is far from the perfect solution. Due to the difference in the trend line's and the $y = x$ line's slopes, this form of correction over-corrects the lower free energy values and under-corrects the higher free energy values. This is why in conjuncture with the point of logP being found from the difference of free energy values discussed in the previous paragraph, there isn't any improvement found in the prediction

of logP despite the improvements to free energy predictions. A better correction factor for solvation would be one that took into account its free energy value, made smaller changes to low values, and larger changes to higher values. As there was no proper training/testing split and these correction factors were only being pursued as a proof of concept, a more complicated correction method was not attempted. While ultimately correction of the c term did not lead to better logP prediction, the effect it had on hydration free energy predictions is a good showcase that this form of data improvement could lead to better QM predictions. To investigate if this would have any practical value, the correction factor would have to be found from experimental data from molecules that were not being predicted via the QM method. The QM data would have to be split into a testing and a training set to understand if the correction factor was of value to data outside that it was created on, and this would cause dilution of the data used to make the QM models. As my dataset was relatively small with 107 molecules, I did not pursue the evaluation of correction factors further.

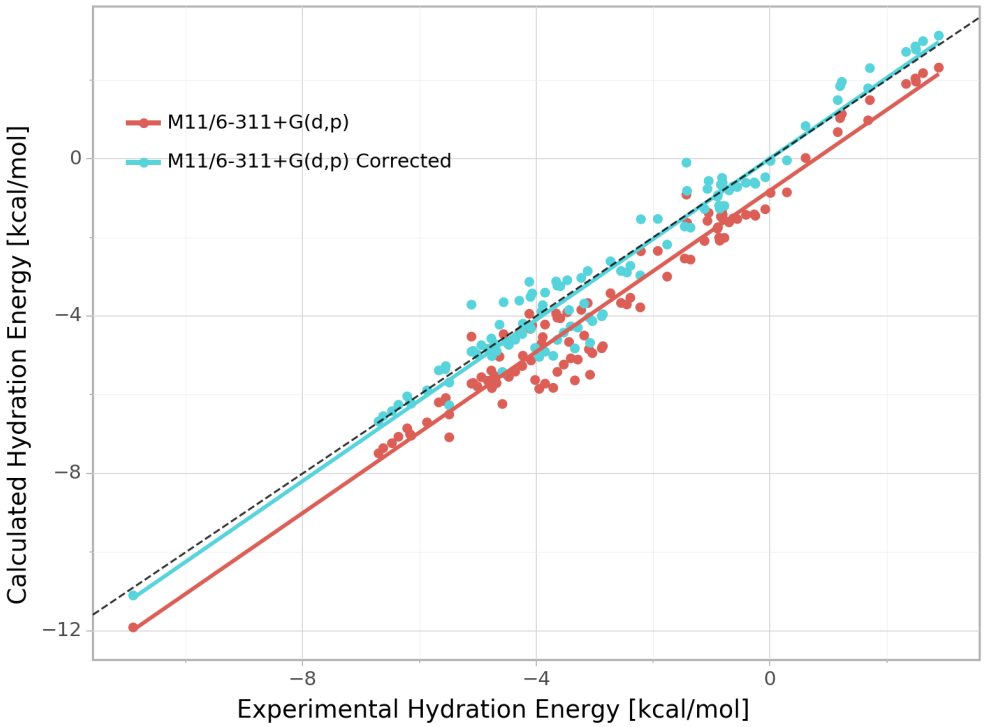

Figure 5.12: QM-calculated hydration free energies for the M11/6-311+G(d,p) level of theory with correction factor applied. Hydration free energy (ΔG_H) is in kcal/mol and calculated at 298.15 K

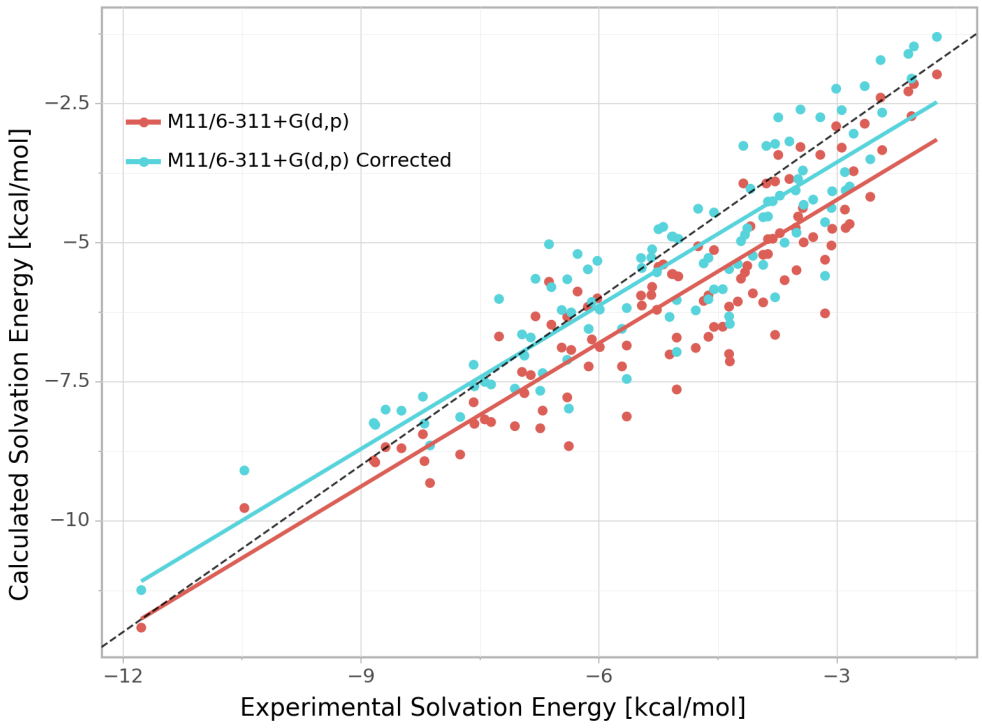

Figure 5.13: QM-calculated solvation free energies for the M11/6-311+G(d,p) level of theory with correction factor applied. Solvation free energy (ΔG_S) is in kcal/mol and calculated at 298.15 K

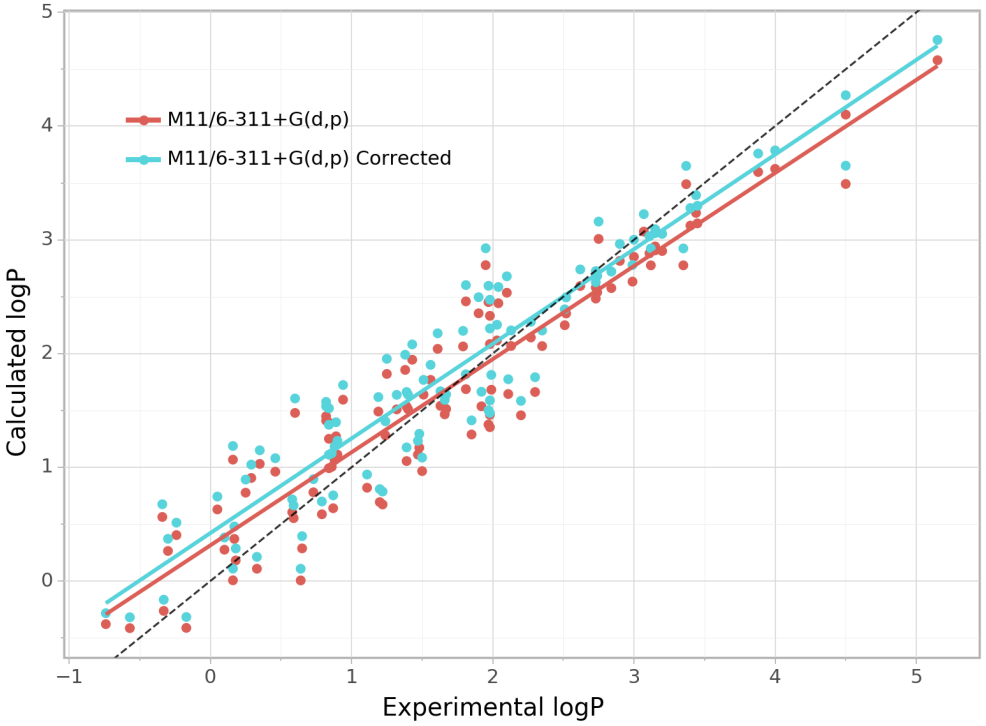

Figure 5.14: QM-calculated logP values for the M11/6-311+G(d,p) level of theory with correction factor applied. Calculated at 298.15 K

5.4 Machine Learning

The results for the ML logP calculations can be seen in Table 5.6 and in Figure 5.16. First, a Random Forest model using the default hyperparameters selected by the sci-kit learn package was performed. From Table 5.6 we can see that this model was already performing better than all of the QM models found in Table 5.2.

Table 5.6: LogP machine learning prediction errors using Random Forest. Calculated for 298.15 K, errors averaged over 100 resamples.

Model	R^2	RMSE	Bias	SDE
RF	0.890	0.383	0.296	0.293
RF + Opt	0.893	0.378	0.297	0.287
RF + Opt + RFE	0.904	0.357	0.277	0.278

Table 5.7: Optimisation hyperparameters percentage chosen. Those features that were not chosen for any of the 100 test train splits are not shown.

Max Features		Max Depth					Min Samples Split				Min Samples Leaf	
SQRT	1/3	5	8	10	13	15	2	3	4	5	1	2
17%	83%	19%	43%	14%	17%	7%	56%	34%	9%	1%	97%	3%

Next, hyperparameter optimisation was performed for each resample, the percentage that each feature was chosen across all 100 test train splits is shown in Table 5.7. In terms of maximum features to consider when splitting a node, one third of the total number of features was chosen more often than the square root of the total features. Literature agrees that for regression one third is the best performing value,[86] so this was to be expected. The binary logarithm of the features was never selected, if I were to improve the efficiency of optimisation I would remove this value from the grid search. However, the square root of the features was still selected a statistically important number of times, therefore its inclusion optimisation should be kept. The maximum depth of the forest is the parameter that shows the greatest variance in both the numerical range of the value and spread of different values chosen. It can be seen the value of 8 was the most chosen, but 5, 10, and 13 were all chosen with around half as much frequency as 8. The values of 2, 3, and 20 were not chosen for any split, to improve optimisations speed I would remove these values. Max depth appears to be a parameter where a large number of values to be tested is required, a change to a stepwise increase in the values trialled would likely be of benefit. To try improving the optimisation I would try all values between 4 and 15, although this would give a significant increase to the computation expense of the optimisation. If optimisations began to take too long to run, one option would

be to use a randomised grid search, these operate by not trying every combination but by randomly picking options until x many combinations have been tried. This is not as ideal as the full grid search, but averaged over the 100 test train splits it should have almost the same effect. The minimum number of samples required to split a node most often chose the lowest value possible of 2, with 3 and 4 having decreasingly less frequency. A value of 5 was only chosen once and 7 and 10 were never selected so I would recommend these 3 values for removal. For minimum number of samples that must be described by a leaf (final) node, almost unanimously a value of 1 was chosen, I would remove the values of 2, 3, and 5 in future models. From Table 5.6 we can see that inclusion of optimisation did come with a modest increase in prediction power of the model, but nothing significant. This agrees with literature that Random Forests are not highly dependent on the hyperparameters used,[86] if computational power was a limiting factor, optimisation could be done only once for the full dataset, or removed entirely without much loss in performance.

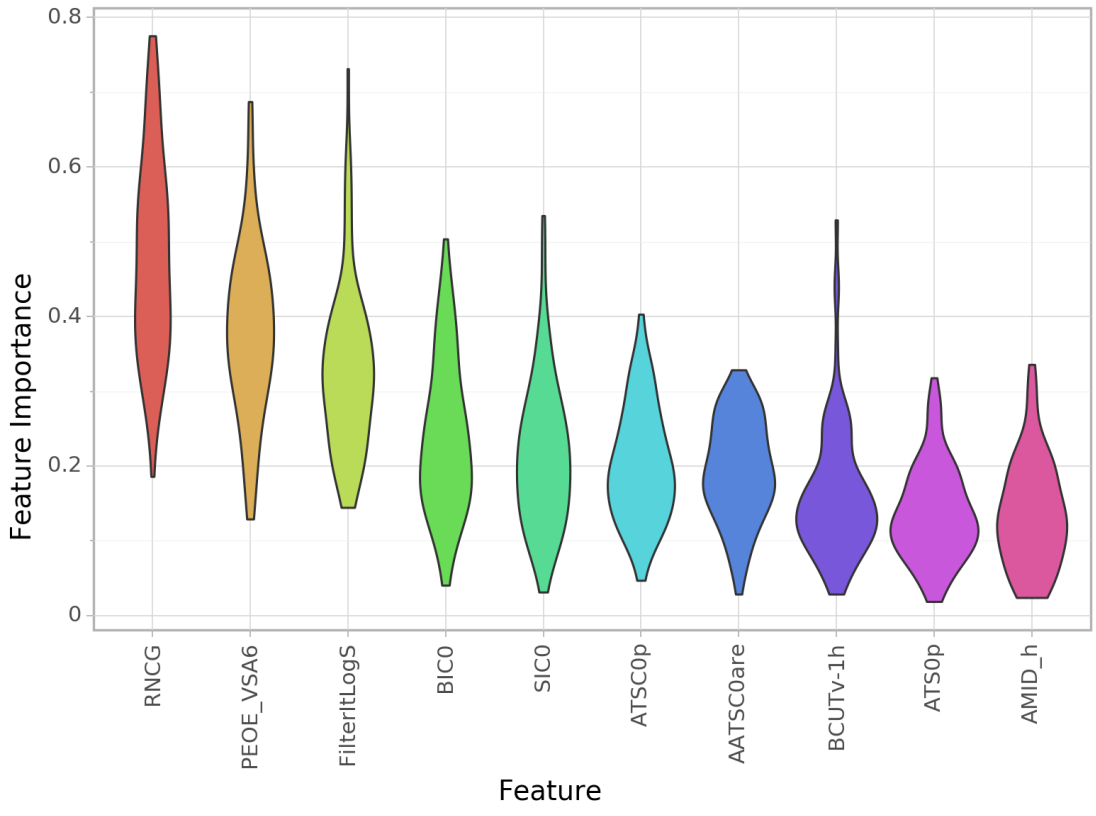

Figure 5.15: Graph of features (in this case molecular descriptors) against their importance values across all 100 test train splits. The graph shows the 10 features that had the highest average importance, with importance decreasing from left to right. A feature importance of 1 would represent that the feature had 100% weighting towards the prediction of logP.

Finally, the model was improved one last time through the use of recursive feature elimination (RFE). When looking across all test train splits, a total of 407 features were kept at least once, leaving 715 features that were never used in any of the RFE models and therefore 715 features that had no relevance to the value of logP. On average 62 features remained after RFE, with the maximum remaining being 231 and the minimum remaining just 2 features. Vector normalisation was performed on feature importance values across each split, so values from splits that contain different amounts of remaining features are equivalent. These two values were the most commonly selected features, appearing in all 100 splits: a computational prediction of solubility (FilterItLogS), which is expected as logP and solubility are intrinsically linked, and relative negative charge (RNCG), which describes how negative charge is distributed throughout a molecule. With just these two descriptors a model was produced with a RMSE of 0.632 log units, this was the worst performing split overall, however it is impressive that with only two descriptors, a prediction that is accurate to almost half a logP unit could be made. The 10 most important features to the calculation of logP can be found in Figure 5.15. We can see that RNCG is the most important feature overall, however for some splits it had much higher importance than others. RNCG is a good way of representing delocalised electrons, perhaps for splits where lots of aromatics were in the testing split RNCG was of much greater importance than others. This trend of the most important descriptors having a large range of importance values can be seen for the top 5 features, suggesting that flexible features that describe the full range of molecule types in your database are the most useful. The second most important feature was PEOE_VSA, this value represents the total van der Waals surface area which is an important attribute for cavitation energy calculation in the process of solvation. The next two are BIC0 and SIC0, these features are similar in that they are used to describe the structure of the molecule. The 6th, 7th and 9th most important descriptors belong to the Autocorrelation of Topological Structure family, which is a method of describing a property across the topological area of the molecule. The two properties being measured with this method are: "p", polarisability which is of course highly relevant to solvation, and "are", the Allred-Rochow electronegativity which measures the electrostatic force exerted from the effective nuclear charge on the valence electrons. Interestingly, with this last term we are approaching the realm of QM calculations, as it is estimated from Slater's rules. Although we have not explicitly included any QM data, it is present and of high importance in the ML prediction. This is a testament to the importance of QM calculations in the realm of logP and solvation calculation. BCUTv-1h is another method of measuring van der Waals volume. The final descriptor AMID_h is a measure of the number of hydrogens present, a simple count is the 10th most important feature. This

seems out of place amongst much more complicated and intricate features, however H-bonding is very important to hydration of a solute and with H being the most common atom amongst all molecules it is perhaps understandable why this feature was so important to the calculation of logP. Although, relatively few hydrogens in the molecules will have the ability to form H-bonds, so this does not seem to be the best descriptor to describe a molecule's H-bonding potential. The top 10 important features that were just discussed seem to be related: they are important to the calculation of solvation free energy. From this we can infer that just as with the first principle calculations, where we calculated solvation free energy in both octanol and water and then calculated logP, the ML algorithm appears to be taking a similar approach. As can be seen from Table 5.6, the removal of noisy features gave the greatest increase in performance and lead to the best performing ML logP prediction model.

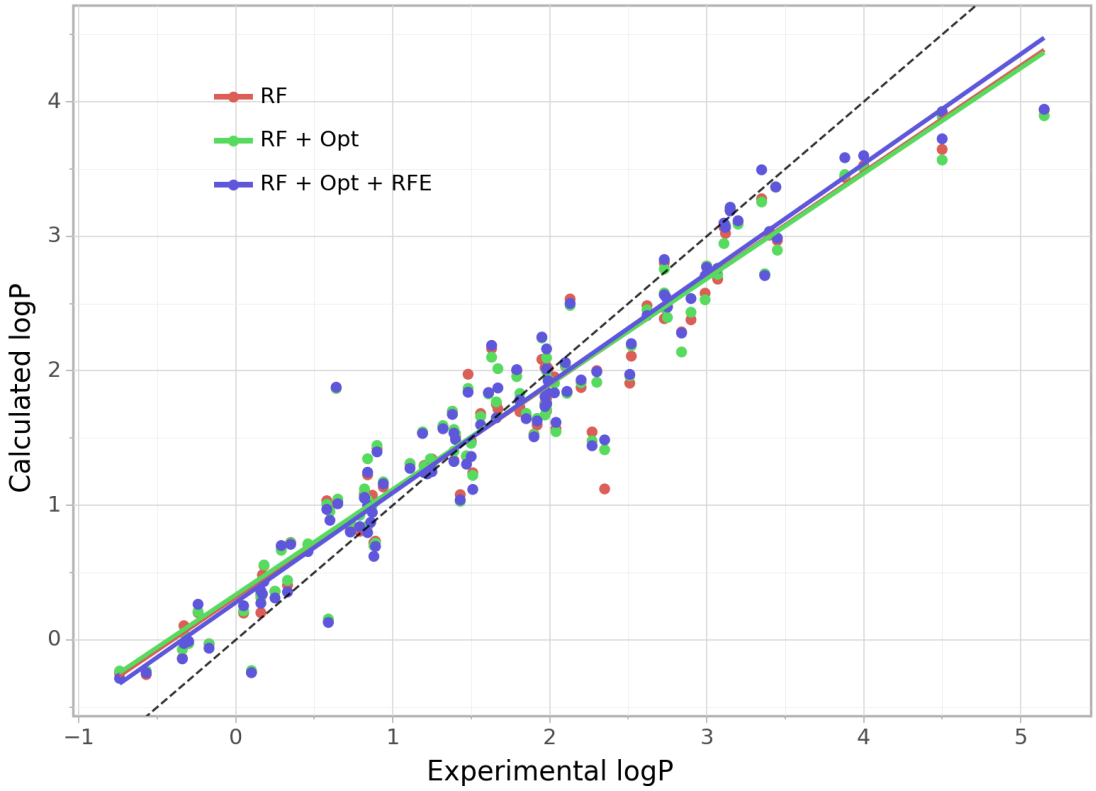

Figure 5.16: Experimental logP values against ML-calculated values, showing the Random Forest model, the Random Forest model with hyperparameter optimisation, and the Random Forest model with hyperparameter optimisation and recursive feature elimination. Calculated for 298.15K

Figure 5.16 shows the average predictions of each molecule's logP value for the 100 test sets of each model. Due to the random nature of the split, some molecules

had more data points than others, but all models had the same number of tests per molecule due to the predefined seed used. On average, each molecule had around 25 measurements, with the highest being 31 and the lowest being 16. Although the data size for finding these average values is on the low end for precisely calculating a molecule's logP value, this method is being used as a way of seeing general trends in logP predictions through ML methods and therefore is acceptable. Viewing Figure 5.16, there is no apparent trend across any of the three models tried. With the example of the QM predictions and the basis sets, it was clear from the graph that 6-311+G(d,p) performed slightly better across all molecules. With the ML predictions, results are more varied molecule per molecule, with some molecules being very similar across the three models and some having largely different predictions. This is to be expected, in the case of QM models the results are found through very repeatable calculations, improving the complexity of the basis set for example will improve results across all molecules. In the case of ML models, the results are much more random, depending on the test train split, the hyperparameters, the features removed, and the construction of the decision trees. These lead to more inconsistent improvements to the model with the addition of further complexity. An interesting similarity between the QM and ML results is the similarity of the trend lines: when logP is low the models tend to over-predict the value and when logP is high they tend to under-predict.

Figure 5.17 shows the averaged predicted values separated by molecule group for the Random Forest model with optimisation and RFE. When compared to the same graph for the QM predictions (Figure 5.9), there are fewer linear trends within the groups, this is most apparent with the alkane group. Interestingly for the groups that the QM model performed well in, the ML model also performs well, like the carboxylic acids, cycloalkanes and nitro groups. These groups had molecules that were very simple, with one functional group and an aliphatic chain, therefore it seems that these simple molecules are well represented through QM models and through molecular descriptors alike. It appears that the increase in accuracy of the ML model over the QM is in the bias of the predictions, this can be seen most clearly in the aromatic nitro, carboxylic acid, and nitrile groups. These groups have predictions that resemble a line and in these cases the groupings can be seen to be shifted closer to the $y = x$ line. The ML model also performs much better at predicting the values for the sulphured group, which was one that the QM particularly struggled with. However the ML predictions can be seen to be less reliable as a whole as it is more prone to outlier values. This can be best seen comparing the ML and QM alkane, alkene, ether, and halogenated aromatics groups. While the ML model outperforms the QM model on overall errors, for some groups in particular the ML

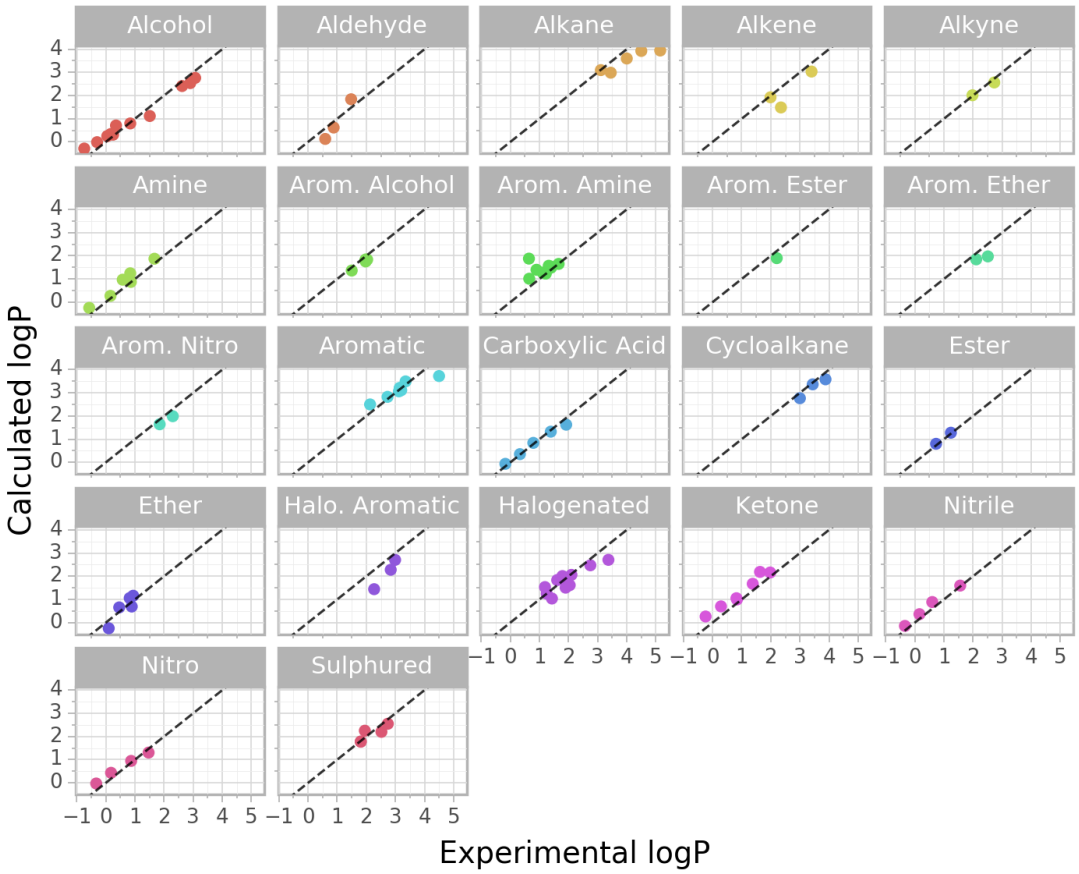

Figure 5.17: Experimental logP values against ML-calculated values separated by group, for the Random Forest with optimisation and recursive feature elimination model. Calculated for 298.15K

seems to struggle more than the QM. These groups are long chain alkanes, halogenated aromatics, and halogenated groups. Whilst there are relatively few datapoints for the long chain alkanes with only 3 molecules, considering the halogenated and halogenated aromatics groups together there are 14 molecules, enough to consider this result to be statistically significant. To improve the model, either more molecular descriptors that described the effect of halogen atoms should be included or the accuracy of calculated descriptors that already are present in the database could be improved. When discussing the linear nature of prediction for many of the groups in Section 5.3, it was suggested that a correction factor could be applied on a group basis to improve results. I believe this could also be applied to these ML results, however with less success due to the tendency to have more outlier values.

In the SAMPL6 challenge it was found that ML models that used QM-calculated molecular descriptors were the most successful, this was discussed in Section 2.5. To investigate this further, it was decided to include the quantum mechanically cal-

Table 5.8: LogP machine learning prediction errors with the addition of QM-calculated hydration and solvation free energies. Calculated using optimisation and RFE for 298.15 K

Model	R^2	RMSE	Bias	SDE
RF + Opt + RFE	0.904	0.357	0.277	0.278
With QM Data	0.904	0.358	0.278	0.278

culated free energies of the molecules in the water and octanol solvents from the M11/6-311+G(d,p) theory. The difference of these values gives the free energy of transfer between octanol and water, from which logP can be directly calculated. We know that these values lead to a fairly accurate prediction of logP with a RMSE of 0.418. Therefore it was expected that the inclusion of these features would lead to an increase in the performance of the model. As can be seen from Table 5.8, there was statically no difference made with the inclusion of QM-calculated free energies. Looking at the RFE results, hydration free energy was included as a descriptor 68 times out of 100 test train splits. It is clear therefore that the ML algorithm recognised the significance of hydration free energy effect on the calculation of logP, but its feature importance was never significant. In the case of solvation free energy it was included a mere 6 times, always paired with inclusion of hydration energy, even though theoretically it has the same importance as the hydration free energy. In fact, it could be argued that solvation free energy is more important, as most logP values in the dataset are above 0 log units, and therefore partition more in the octanol phase. I believe this can be explained by how RFE operates, it first fits a model using all features and gives each feature a value depending on the impact it had on the value of the prediction. It then removes the least impactful feature and starts the process over again, until the last feature is above a certain significance value threshold. In practice, this means that features are considered individually when it comes to RFE removal. As can be seen from Tables 5.3 & 5.4, the hydration free energy values were more accurately predicted than the solvation free energies. This could explain why hydration free energy remained more often than solvation free energy, it has a higher accuracy and therefore performed better when considered on an individual level than solvation. I believe these results could be improved via two options. I chose the M11/6-311+G(d,p) free energy values as this was the model that gave the best logP calculation, with the thought process that the model could use the combination of the two values. This proved not to be the case for the vast majority of splits, therefore if I gave the model the difference of the free energies from the M11/6-311+G(d,p) calculation, both values would be represented together as the free energy of transfer. Alternatively, since it is not logP but free energy values I am supplying, the better level of theory to use would be

the B3LYP/6-311+G(d,p) which better performed on the accuracy of free energy calculations. However, there is a more simple explanation that could also be possible. With the large number of molecular descriptors given to the model, it may be able to calculate the hydration and solvation error to the same or better degree of accuracy that the QM predictions that I have supplied. Although it does not have the free energies explicitly, the model already contains the information through a combination of other descriptors.

Table 5.9: Hydration and solvation free energies machine learning prediction errors using Random Forest. Calculated using optimisation and RFE at 298.15 K

Model	R^2	RMSE	Bias	SDE
Hydration	0.913	0.686	0.430	0.494
Solvation	0.750	0.906	0.658	0.457

To test if the hypothesis of the ML model already contained the free energies implicitly, I used the ML model to find the hydration and solvation free energies directly. These results can be seen in Table 5.9, as with the QM results found in Tables 5.3 & 5.4, these values were found less accurately than the logP values. We can see that indeed the ML model is able to calculate both hydration and solvation free energy with more accuracy than the M11/6-311+G(d,p) level of theory that we supplied the model with in the above paragraph, explaining why the addition of QM free energy data did not improve the ML models logP prediction power. Interestingly, these prediction values are quite similar to the QM values, the hydration free energy is calculated significantly better than the solvation free energy. On viewing Figures 5.18 & 5.19 it is observed that the slope of the hydration trend line is much closer to that of the $y = x$ line than the solvation free energies trend line, much like the QM results. When this was discussed in Section 5.3, it was believed to be caused by the solvation model for water being more developed than octanol's. The mirror nature of the QM and ML results could be explained by the molecule descriptors also being more developed for water. Many of the descriptors are calculated via mathematical models, therefore it would not be an unreasonable assumption when these models were trained, aqueous accuracy was prioritised over accuracy in solvents. Or perhaps, there is something intrinsic to the octanol molecule: its shape, the duality of non-polar and polar areas, greater size and therefore increase in conformers etc. that makes representation of octanol difficult in comparison to the smaller and more symmetric water molecule. Further investigation would be required to give a concrete explanation.

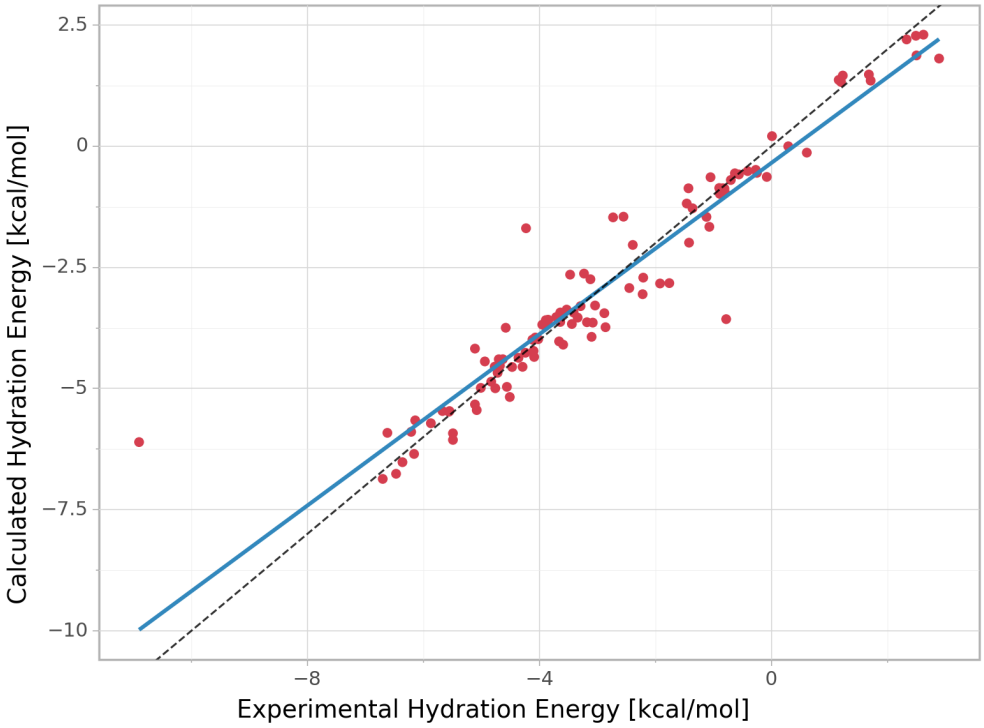

Figure 5.18: ML-calculated hydration free energies using optimisation and RFE. Hydration free energy (ΔG_H) is in kcal/mol and calculated at 298.15 K

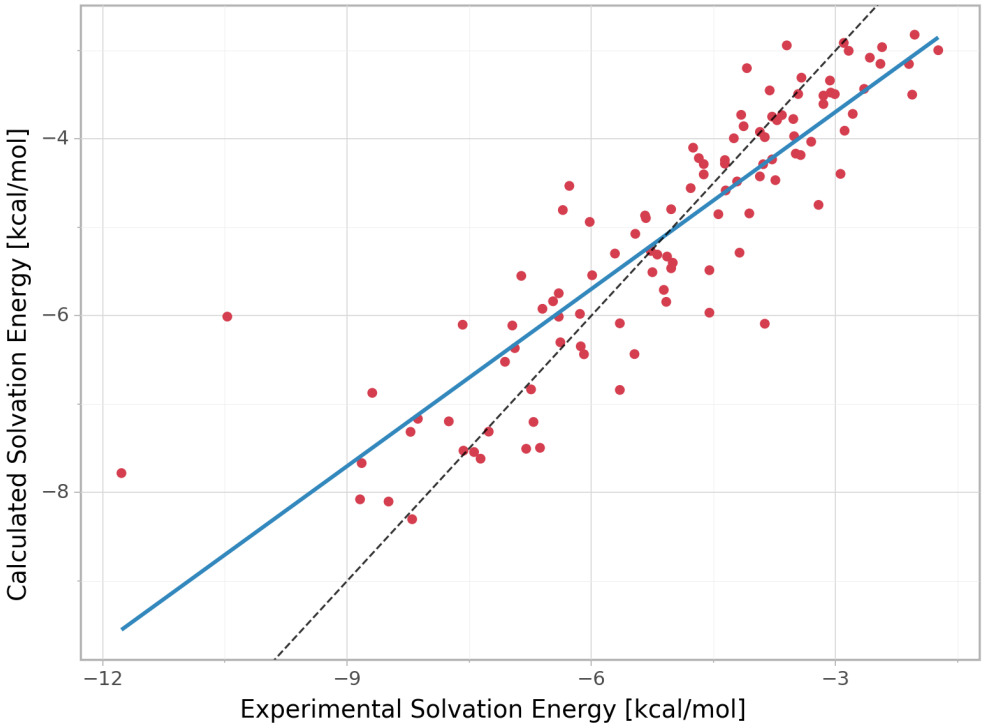

Figure 5.19: ML-calculated solvation free energies using optimisation and RFE. Solvation free energy (ΔG_S) is in kcal/mol and calculated at 298.15 K

Chapter 6

Chemical Engineering

There are a number of links from the research presented in this thesis to real world applications that apply to the chemical engineering sector. LogP has already been established in Chapter 2 to be indispensable in the modelling of potential drug molecules, and this also applies to the agrochemical and environmental sector to the same degree. However, the most broad and multidisciplinary physical property in regards to the chemical engineering domain predicted in this work is that of solvation free energy. Solvation free energy is directly involved in any process where the interaction of gas and solution is experienced, this is such a common occurrence in large scale chemical processing that solvation free energy has use in almost every chemical engineering sector. Solvation free energy can also be used to calculate other properties such as pKa, which can be used in the design of other chemical engineering processes.

6.1 In the Engineering Context

The Gibbs free energy of solvation has been defined by Ben-Naim in his book on Solvation Thermodynamics,[90] as the work required to move a molecule from a fixed position in an ideal gas to a fixed position in a solution whilst under constant temperature and pressure. The use of solvation free energy in the chemical engineering context can be seen in the design of absorption towers, which are most commonly used in the removal of impurities in a gas stream.

Henry's law constants (K_H) are partition coefficients much like with logP, K_H is a ratio of the concentration of a molecule between the liquid and the gaseous phase:

$$K_H = \frac{C_{g(i)}}{C_{l(i)}} \tag{6.1}$$

Where $C_{g(i)}$ & $C_{l(i)}$ are the equilibrium molar concentrations of species i in the

gaseous and liquid phases, respectively. K_H can be directly calculated through solvation free energy through the simple equation:

$$\Delta G_{solv} = RTln(K_H) \tag{6.2}$$

6.1.1 Absorption/Stripping[91]

Henry's constants are perhaps most frequently used in the process of absorption/stripping, where absorption is the process of transfer of gas molecules into the liquid phase and the opposite process is known as stripping.

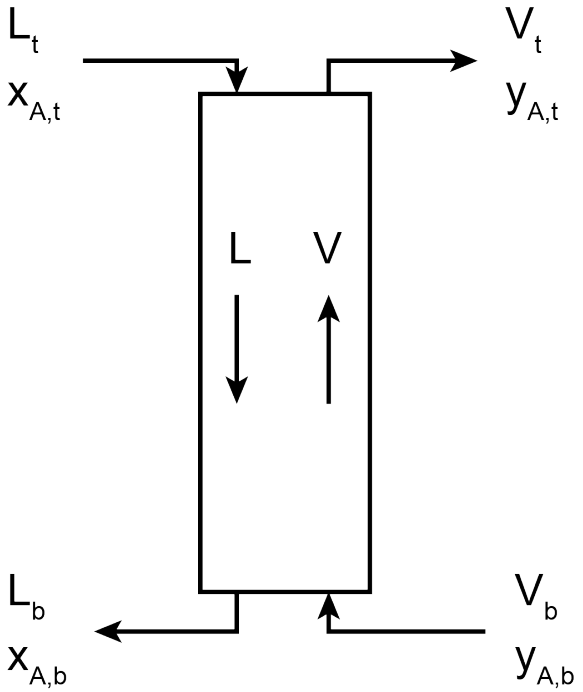

Figure 6.1: Schematic diagram of a counter-current absorption process.

Figure 6.1 shows a diagrammatical representation of the adsorption process for a counter-current system, for this system the overall mass balance is given by:

$$L_b + V_t = L_t + V_b \tag{6.3}$$

Where L is liquid flow rate, V is vapour flow rate, and subscripts t & b represent the top and bottom of the absorption tower respectively.

The component balance for species A is then given by:

$$L_b x_{A,b} + V_t y_{A,t} = L_t x_{A,t} + V_b yA, b \tag{6.4}$$

Where y_A & x_A is the mole fraction of A in the vapour and liquid phases, respectively. This can be represented on a solute-free basis through use of the solute-free concentrations:

$$\bar{X}_A = \frac{x_A}{1 - x_A} \tag{6.5}$$

$$\bar{Y}_A = \frac{y_A}{1 - y_A} \tag{6.6}$$

If the gas supplying the solute to be absorbed is completely insoluble in the solvent and the solvent is completely nonvolatile, then gas and solvent rates are constant through the absorber:

$$\bar{L}\bar{X}_{A,b} + \bar{V}\bar{Y}_{A,t} = \bar{L}\bar{X}_{A,t} + \bar{V}\bar{Y}_{A,b} \tag{6.7}$$

Where $\bar{L}$ denotes the flow rate of the nonvolatile and $\bar{V}$ denotes the flow rate of the carrier gas. This equation can be applied to any section of the tower, if applied to the top of the column the operation line is found:

$$\bar{Y}_{A,t} = \frac{\bar{L}}{\bar{V}}\bar{X}_{A,t} + \left(\bar{Y} - \frac{\bar{L}\bar{X}_A}{\bar{V}}\right) \tag{6.8}$$

Where $\bar{X}_A$ & $\bar{Y}_A$ are the mole ratios of A in the liquid and vapour phase, respectively. The operation line, which is a straight line possessing a slope of $\hat{L}/\hat{V}$, can be drawn onto a mole ratio diagram as seen in Figure 6.2. From the XY diagram, the number of stages of the column can be determined by stepping between the mole fraction curve and the operations line.

Through use of Henry's law constants, key parameters in the computation of the equations in this section can be carried out:

$$K_H C_A = P_A = K x_A \tag{6.9}$$

Where C_A is the molar concentration of A, P_A is the partial pressure of A, & K is the vapour-liquid equilibrium constant.

6.1.2 In practice

Absorption/stripping is most often used in industry as a means of purifying a gas stream for safe emission to the atmosphere, notably CO_2 containing streams. CO_2 has gained the largest attention out of all greenhouse gases due to the sheer quantities released: the concentration in the atmosphere went from 300 ppm pre-industrial revolution to 400 ppm.[92] There has been governmental and industrial push to find

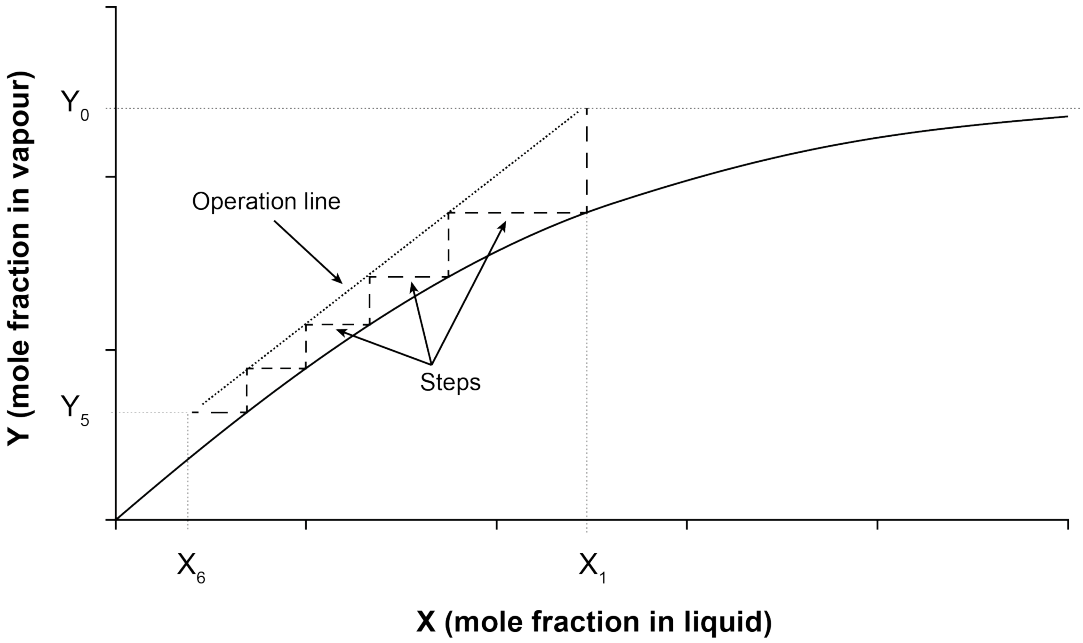

Figure 6.2: XY diagram for a fictitious system. Operation line is plotted and the process of stepping off the XY line to the operation line is shown.

an economically sustainable way to reduce the levels of CO_2 released. Methods of capturing carbon post-combustion are the most flexible, as they can be installed on existing plant designs. Currently, it is widely believed that this technology is still in its infancy, and that economical implementation is still not feasible. It is thought however that post-combustion capture through the use of chemical adsorption towers will be the first practical implementation.[93] The most commonly used solvent in this style of absorption towers is monoethanolamine, while effective at absorbing CO_2, the presence of other flue gases such as SO_2 & NO_2 will cause the formation of salts.[94] Furthermore, the presence of oxygen will cause increased corrosion of the equipment[94] and degradation of the amine solvent. These limitations require significant pre-processing of the flue gas entering the system. The CO_2 capture process operates by the gases being passed through an absorber at 40-60°C, then pumped to a stripper that operates at 100-120°C and 1.5-2 atm. The heat required by the reboiler of the stripper tower is the major energy penalty of the process.[93] A large concern of the implementation of such capture methods in power plants is the reduction in energy output of the installation they cause. A review found that this penalty was approximately 10%, due to the reboiler and gas compression energy demands.[95] Another concern is the storage of CO_2 which is naturally gaseous and therefore occupies a large volume, and compression of gas always comes associated with an energy cost. How the gas will be stored is still up for debate, currently the most favoured option is geological storage.[96] This operates by injecting the gas into depleted oil and gas fields, such as the Sleipner and Snøhvit

projects in Norway. The advantage of using existing petroleum fields is the knowledge that they are closed systems, else the previously stored hydrocarbons would have escaped. The transportation and pumping of the gas into such deposits of course comes at a significant energy cost.[96] This method of storage should be seen as an undesirable but necessary transition stage towards the use of stored carbon in a cyclic economy, where it is used to make carbon-based products.[96] Currently, due to the high energy, transportation, and storage costs associated with the capture of CO_2 gas, it is not an economically viable process.

The ability to lessen the emissions from a process is of great value to a variety of factors that affect chemical engineering design. There is always environmental concerns that exert pressure both during design and post-implementation of a plant. A design has a number of environmental constraints placed on it by numerous sources: there is government legislation that will dictate the concentrations and yearly amounts of chemicals that can be released to the atmosphere. With CO_2 emissions, there are policies in place such as the EU Emissions Trading System, where companies are given an allowance of CO_2 they can freely release without fines. Companies who discharge less than their allowance can sell their excess cap to other companies. Therefore, it is important to run cost-benefit analyses on emissions, especially with newly built facilities, as it can be economical to implement a more expensive plant design that minimises CO_2 release in order to sell excess emissions cap. There is also public perception to be considered, as environmental concern grows amongst the populous, there is ever greater pressure on companies to promote green processes and lessen their environmental impact. Following on from public pressures, companies will often self-implement policies and goals to lower their carbon footprint in order to promote their brand in the public eye, which therefore puts additional pressure on plant design.

Although absorption has been touched upon in depth in this section, there are a variety of other chemical engineering processes that are equally reliant on solvation free energy and the properties that can be calculated from it, such as reactor design (pKa), distillation (K_H), and flash drums (K_H).

6.2 Solvation Free Energy Computation

6.2.1 Experimental

Solvation free energy can be directly measured by allowing a small amount of a given solvent to dissolve into an aqueous solution, the solution is left to allow the

vapour-liquid equilibrium to balance in a closed vessel at constant temperature.[97] Calculation of solvation free energy is found through the Henry's law constant determination method:

$$C_g = C_{aq_0} \frac{K_H(V_g/V_{aq})}{K_H(V_g/V_{aq}) + 1} \tag{6.10}$$

Where C_g is the concentration of the solute in the gaseous phase, C_{aq_0} is the initial concentration of the solute in the aqueous layer, and V_g & V_{aq} is the gaseous and aqueous volume respectively. From finding the Henry's law constants, solvation free energy can be calculated through use of Equation 6.2. This method suffers most notably when attempting to measure solutes that possess low volatilities (which is the case for drug-like molecules[97]), as only a low concentration of solute is present in the gaseous phases which is difficult to measure. This experimental method is very similar to the shake-flask method of logP measurement discussed in Section 2.2. A solute has to be disolved in a solvent, sufficient time for equilibrium to be reached is required etc. Therefore, it suffers from similar limitations of low throughput and automation difficulties.

6.2.2 Computational Calculation

Many of the reasons why computational calculation of logP is advantageous in comparison to its experimental counterparts equally applies to solvation free energy in the engineering focus. The acquisition of physical property data is required before computation of any chemical engineering process can take place. Often, it is mandatory to find not just one data point but many for each property in order to assure accuracy. Whilst handbooks and databanks of experimental properties do exist, they are often outdated or limited, requiring extensive literature searches. And for some rarer or newly developed chemicals, no experimental data has been generated at all. Without proper physical data, pilot plant studies and design cannot be undertaken. It is possible that this data can be acquired through experimental testing, however this is a long and resource intensive process. The availability of physical property models to computationally estimate this data is therefore of paramount importance to the chemical engineering world. Solvation free energy can be used to calculate a variety of physical properties that are required to design chemical processes: Henry's law constants, acid-base dissociation constants, aqueous and solvent solubilities etc. If these properties can be calculated without the need of experimental generation then a much larger catalogue of solutes and solvents can be simulated and tested for a chemical engineering process. This allows for a much larger number of potential chemicals to be tested for a process than would be feasible if experimental data was required for each compound. This opens the possibility for the optimal chemical

(best atom economy, lowest energy requirement, safest, etc.) to be chosen.

Calculation of solvation free energy has been approached in many computational methods, one of which is through empirical use of cheminformatics and molecular descriptors. The simplest model is known as the group contribution method. This can be done in either an atomistic or a fragment approach similar to those discussed in Section 2.3, where solvation free energy is equal to the sum of the atom/fragments:

$$\Delta G_{solv} = \sum a_i x_i + c \tag{6.11}$$

Where a_i & c are regression coefficients, x represents the number of instances of that atom/fragment, and the intercept is found through multilinear regression. This style of empirical calculation is found to work well for small molecules, however, much like ML models, they do not perform well on molecules that are dissimilar to those the model was trained on.[97] These methods do benefit from very low computational expense, therefore can be useful when a large database of potential solvents for a process needs to be screened. But when looking to accurately model a process that could cost million of pounds to produce, they fail to meet the standard.

Currently in the chemical processing simulation world, in programs such as Aspen Plus,[98] estimation of chemical physical properties where no experimental data is available is done through the use of empirical/semi-empirical methods such as UNIFAC.[99] UNIFAC is a group contribution method that operates via estimating the activity coefficients of the molecules in the system, these coefficients are used to quantify how far from ideal conditions the real system deviates. The activity coefficients are related to Henry's law constants by:

$$\gamma_A = \frac{x_A P}{y_A K_H} \tag{6.12}$$

Where γ_A is the activity coefficient of A. We have discussed the limitations of such empirical-based calculation models on data outside of their creation range. Therefore such models are only reliable for determining physical properties of common chemicals. For a company to find a competitive edge, the estimation of new and novel compounds is a necessity, so the need for accurate methods to measure such molecules is apparent.

As has been shown in this work, solvation free energy can be measured through use of QM calculation. The obvious question raised then is whether this style of solvation free energy estimation be used to more accurately simulate chemical processes.

This style of approach is more computationally expensive than the empirical style calculations, however this comes with more accurate results. The background of these styles of calculations was covered in Section 3.1.12. In summary, the solvent can be simulated either though explicit solvent molecules being placed around the solute, or through use of an implicit continuum solvation model, where the solute is surrounded by the properties of the solvent. Explicit models have the potential to be more accurate, however for a truly representative model, 100s or 1000s of solvent models have to be calculated which is far more computationally intensive than its more commonly used implicit variant. Through the SAMPL4 hydration challenge,[100] many attempts were made to find the solvation free energy for a number of previously unknown solutes in the aqueous phase. One of the best performing of these was a QM method utilising an implicit solvation model with functional group specific correction factors, which had a RMSE = 1.23 kcal/mol. This level of error gives rise to an error of roughly 1 log units in calculation of physical properties such as pKa, which is too high of an error for accurate simulation. A study evaluated the performance of the QM COSMO-RS model, UNIFAC, and two other solvation models: the Modified Separation of Cohesive Energy Density (MOSCED) and Abraham methods at estimation of activity coefficients.[101] It was found that the MOSCED model outperformed the others with an average relative deviation of calculated to experimental values of 16.2%, meanwhile the COSMO-RS model had a value of 182% as it struggled to accurately describe long-range interactions. It should be noted, however, that the COSMO-RS method used was not a direct calculation of solvation free energy to find activity coefficients, COSMO-RS was used to find descriptors that were then used to compute the activity coefficient. It would appear that in terms of accurate activity coefficient calculations, QM mechanical methods currently do not have the same level of accuracy as more established methods.

There is a lack of research into the accuracy of QM calculations of solvation free energy that are then used to find activity coefficients. In this work, using the B3LYP/6-311+G(d,p) level of theory, hydration energies were able to be found as accurate as RMSE $\approx$ 0.6, this gives rise to a pKa RMSE $\approx$ 0.40 log units. While this level of error is not ideal when it comes to accurate simulation of a chemical process, it does show that QM models have the potential to have fairly accurate predictions for systems that they describe well, such as the small molecules in my dataset. Development of QM methods is being furthered every year, at some point it does seem likely that these methods will be more accurate than the current database estimation methods. There is also a lack of research on utilising ML methods to calculate chemical physical properties in comparison to methods such as UNIFAC. I have shown in this work that ML methods can be used to accurately

calculate hydration energies with a RMSE ≈ 0.7, whilst this was not more accurate than the best performing QM methods, it was better than the majority of the QM methods tested. I believe that computational calculation of solvation free energy could be used to better improve physical property estimation of chemical processing simulation software, however the research interest is currently not sufficient to truly assess this claim.

Chapter 7

Conclusions and Further Work

For the QM calculation of logP, the complexity of the basis set was seen to make small, but noticeable improvements to the errors. Choice of functional used made a far smaller impact on the performance of the prediction, when the best performing basis set was used, no statistically significant difference was observed between the three functionals. Overall, the M11/6-311+G(d,p) was deemed to be the best level of theory. When QM logP predictions were viewed on a molecular group basis, it was found that molecules from the same family would have a linear prediction line which opens the possibly for family-based correction factors. When observing QM predictions of hydration and solvation values, it was found that changes to the level of theory used had a much more noticeable impact on the results. QM calculations of the individual energies performed significantly worse than logP predictions, with hydration calculation performing better than solvation. It was found that more accurate predictions of hydration and solvation energies does not necessarily lead to better logP predictions. This is due to the nature of logP being equivalent to the difference of the two energies, if the prediction of both the hydration and solvation values is shifted by the same amount, then this will not be reflected in the logP prediction. This allowed the M11 functional to outperform the B3LYP, despite B3LYP giving closer values to the experimental energy values. The Random Forest ML method was able to predict logP values with a high degree of accuracy without any optimisation or RFE. The addition of those two extra layers of complexity returned a noticeable improvement to the accuracy of the predictions, however, both individually came with a large increase in the computational expense of the model. When looking at the predictions for molecular groups, some linear trends for molecules of the same family were observed, but to a much lesser extent than that of the QM methods. Addition of QM-calculated free energies to the training data of the model proved to have no impact on the ML predictions, this could have been due to the QM data not being accurate enough, or not being fed to the ML model in the best form. The ML algorithm was also able to make reasonable predictions for

the calculation of the both the energy values. For the methods tried on the dataset particular to this work, ML predictions outperformed their QM counterparts.

In Section 2.5 of the Literature Review, a state-of-the-art comparison of QM and ML logP prediction methods showed that, currently, QM methods performed better than ML methods. However, from the results of my own predictions, ML proved to be the more accurate method. Fundamentally, I believe the contradictory results are due to the molecules in the dataset. In my database, all of my molecules were very similar in size and the variety in types/functional groups was low, ML models perform well if they are used on molecules similar to those they are trained on. In my case due to the relatively uniform nature of my database, this was the case. This is well exemplified when viewing the SAMPL6 submission by Patel el al. that was discussed in Section 2.5, they trained their ML model using the same database as this work, although with different molecules. Their RMSE on the SAMPL6 molecules was approximately 70% higher than the RMSE found from in-house testing, due to the difference of their training molecules compared to the molecules they were predicting. From my results and literature, we can conclude that ML learning outperforms QM when there is sufficient enough background data for the model to be built from. In the cases of larger and more complex molecules, data acquisition becomes more difficult, in this instance QM predictions should be able to outperform ML models.

The ultimate aim of this thesis was to evaluate which computational method performed the best for calculation of logP values. Although my results showed ML outperforming QM in terms of accuracy, I believe this was due to the small nature of the molecules, and for a larger, more drug-like set of molecules, QM calculations will lead to the most accurate prediction of logP. QM also has the ability to utilise correction factors to further improve the accuracy of its calculations more so than ML. These statement are backed up by the results discussed in the Literature Review. However, QM predictions have a large drawback: they are far more computationally expensive than a prediction from a pre-existing ML model. ML models can take a long time to train, but once trained all that has to be calculated is the molecular descriptors for the molecules to be predicted, which is a quick process. With QM calculations, the time taken can range from minutes to hours, and the time taken increases dramatically with the size of the molecule. Practically, which is the best computational model depends on what the use is. If you wish to obtain the most accurate logP value possible for a small number of molecules, QM should be your choice. However, if you aim to search a search space containing millions, or billions of compounds to find potential drug molecules, then computational speed

could be deemed more important than the accuracy of the result. This could even be done in two stages, ML predictions to narrow down to very likely candidates, after which more extensive QM calculations can be run.

In terms of future work of the project, a number of potential methods of improvement can be found from the SAMPL6 challenge as discussed in Section 2.7. The solvent model used for the QM calculations could be improved through trialling the COSMO-RS model. The def2-SVP basis was also highlighted to perform well for QM predictions of logP. My QM model did not take into account the "wet" nature of octanol (in the octanol-water partition some percentage of the octanol layer is comprised on water molecules), entries from the SAMPL6 challenge that compared both dry and wet octanol predictions on identical models showed a RMSE increase of around 0.05-0.1 logP units for the wet models. More ML models, such as Neural Networks, could have been trialled to see if a different method would outperform Random Forest, or to use an ensemble ML approach, where the average of many different ML models is taken. More types of QM data could have been fed to the ML program to see if this would improve accuracy, descriptors such as HOMO/LUMO energies, dipole moments etc. instead of solely calculated energies. And ultimately, the most important improvement that could be made to this research in terms of practicality would be to test these results on a database of more drug-like molecules.

Bibliography

(1) S. Amézqueta, X. Subirats, E. Fuguet, M. Rosés and C. Ràfols, *Liquid-Phase Extraction*, Elsevier, London, 2020, pp. 183–208.

(2) J. Kujawski, H. Popielarska, A. Myka, B. Drabińska and M. K. Bernard, "The log P Parameter as a Molecular Descriptor in the Computer-aided Drug Design", *Computational Methods in Science and Technology*, 2012, **18**, 81–88.

(3) J. Sangster, "Octanol Water Partition Coefficients of Simple Organic Compounds", *Journal of Physical and Chemical Reference Data*, 1989, **18**, 1111–1229.

(4) L. Di and E. H. Kerns, *Drug-Like Properties*, Elsevier, London, 2016, pp. 39–50.

(5) J. E. Rice, *Organic Chemistry Concepts and Applications for Medicinal Chemistry*, Elsevier, London, 2014, pp. 85–92.

(6) R. N. Smith, C. Hansch and M. M. Ames, "Selection of a reference partitioning system for drug design work", *Journal of Pharmaceutical Sciences*, 1975, **64**, 599–606.

(7) C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings", *Advanced Drug Delivery Reviews*, 1997, **23**, 3–25.

(8) V. Poongavanam, B. C. Doak and J. Kihlberg, "Opportunities and guidelines for discovery of orally absorbed drugs in beyond rule of 5 space", *Current Opinion in Chemical Biology*, 2018, **44**, 23–29.

(9) S. D. Krämer, H. E. Aschmann, M. Hatibovic, K. F. Hermann, C. S. Neuhaus, C. Brunner and S. Belli, "When barriers ignore the "rule-of-five"", *Advanced Drug Delivery Reviews*, 2016, **101**, 62–74.

(10) P. Matsson, B. C. Doak, B. Over and J. Kihlberg, "Cell permeability beyond the rule of 5", *Advanced Drug Delivery Reviews*, 2016, **101**, 42–61.

(11) B. G. Giménez, M. S. Santos, M. Ferrarini and J. P. Dos Santos Fernandes, "Evaluation of blockbuster drugs under the rule-of-five", *Pharmazie*, 2010, **65**, 148–152.

(12) M. J. Waring, J. Arrowsmith, A. R. Leach, P. D. Leeson, S. Mandrell, R. M. Owen, G. Pairaudeau, W. D. Pennie, S. D. Pickett, J. Wang, O. Wallace and A. Weir, "An analysis of the attrition of drug candidates from four major pharmaceutical companies", *Nature Reviews Drug Discovery*, 2015, **14**, 475–486.

(13) I. Kola and J. Landis, "Can the pharmaceutical industry reduce attrition rates?", *Nature Reviews Drug Discovery*, 2004, **3**, 711–715.

(14) S. Morgan, P. Grootendorst, J. Lexchin, C. Cunningham and D. Greyson, "The cost of drug development: A systematic review", *Health Policy*, 2011, **100**, 4–17.

(15) Matej Mikulic, *Global Pharmaceutical Industry - Statistics Facts*, tech. rep., Statista, 2019.

(16) T. M. Letcher, *Thermodynamics, Solubility and Environmental Issues*, Elsevier, London, 2007, pp. v–vi.

(17) T. M. Letcher, *Thermodynamics, Solubility and Environmental Issues*, Elsevier, London, 2007, pp. 3–16.

(18) R. C. Kirkwood, "Recent developments in our understanding of the plant cuticle as a barrier to the foliar uptake of pesticides", *Pesticide Science*, 1999, **55**, 69–77.

(19) F. Kerler and J. Schoenherr, "Accumulation of Lipophilic Chemicals in Plant Cuticles: Prediction From Octanol/Water Partition Coefficients", *Arch. Environ. Contam. Toxicol*, 1988, **17**, 1–6.

(20) Welfare. Secretary's Commission on Pesticides and Their Relationship to Environmental Health, *Report of the Secretary's Commission on pesticides and their relationship to environmental health*, tech. rep., US Department of Health, Education and Welfare, 1969.

(21) K. Chamberlain, A. A. Evans and R. H. Bromilow, "1-Octanol/Water Partition Coefficient and pKa for Ionisable Pesticides Measured by a pH-Metric Method", *Pesticide Science*, 1996, **47**, 265–271.

(22) Y. Ran, Y. He, G. Yang, J. L. Johnson and S. H. Yalkowsky, "Estimation of aqueous solubility of organic compounds by using the general solubility equation", *Chemosphere*, 2002, **48**, 487–509.

(23) A. Sabljić, H. Güsten, H. Verhaar and J. Hermens, "QSAR modelling of soil sorption. Improvements and systematics of log KOC vs. log KOW correlations", *Chemosphere*, 1995, **31**, 4489–4514.

(24) R. J. Serne, R. L. Treat and R. O. Lokken, *Development of a technical approach for assessing environmental release and migration characteristics of Hanford Grout*, tech. rep., Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), 1985.

(25) G. Briggs, Predicting uptake and movement of agrochemicals from physical properties, talk given at the SCI Meeting on uptake of agrochemicals and pharmaceutical, London, UK, 1997.

(26) C. M. Tice, "Selecting the right compounds for screening: does Lipinski's Rule of 5 for pharmaceuticals apply to agrochemicals?", *Pest Management Science*, 2001, **57**, 3–16.

(27) E. D. Clarke and J. S. Delaney, "Physical and Molecular Properties of Agrochemicals: An Analysis of Screen Inputs, Hits, Leads, and Products", *CHIMIA*, 2003, **57**, 731–734.

(28) E. D. Clarke, "Beyond physical properties-Application of Abraham descriptors and LFER analysis in agrochemical research", *Bioorganic and Medicinal Chemistry*, 2009, **17**, 4153–4159.

(29) G. Hao, Q. Dong and G. Yang, "A Comparative Study on the Constitutive Properties of Marketed Pesticides", *Molecular Informatics*, 2011, **30**, 614–622.

(30) Grand View Research, *Agrochemicals Market Analysis By Product, By Application, By Region, And Segment Forecasts, 2018 - 2025*, tech. rep., 2017.

(31) F. Vlahovic, S. Ivanovic, M. Zlatar and M. Gruden, "Density functional theory calculation of lipophilicity for organophosphate type pesticides", *Journal of the Serbian Chemical Society*, 2017, **82**, 1369–1378.

(32) W. Klein, W. Kördel, M. Weiß and H. J. Poremski, "Updating of the OECD Test Guideline 107 "partition coefficient N-octanol/water": OECD Laboratory Intercomparison Test on the HPLC method", *Chemosphere*, 1988, **17**, 361–386.

(33) OECD, "Test No. 107: Partition Coefficient (n-octanol/water): Shake Flask Method", *OECD Guidelines for the Testing of Chemicals*, 1995, **1**.

(34) S. K. Poole and C. F. Poole, "Separation methods for estimating octanol-water partition coefficients", *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, 2003, **797**, 3–19.

(35) S. Lane, "Coupled chromatography-mass spectrometry techniques for the analysis of combinatorial libraries", *Handbook of Analytical Separations*, 2000, **1**, 127–161.

(36) L. Hitzel, A. P. Watt and K. L. Locker, "An increased throughput method for the determination of partition coefficients", *Pharmaceutical Research*, 2000, **17**, 1389–1395.

(37) J. De Bruijn, F. Busser, W. Seinen and J. Hermens, "Determination of octanol/water partition coefficients for hydrophobic organic chemicals with the "slow-stirring" method", *Environmental Toxicology and Chemistry*, 1989, **8**, 499–512.

(38) C. Barzanti, R. Evans, J. Fouquet, L. Gouzin, N. M. Howarth, G. Kean, E. Levet, D. Wang, E. Wayemberg, A. A. Yeboah and A. Kraft, "Potentiometric determination of octanol-water and liposome-water partition coefficients (log P) of ionizable organic compounds", *Tetrahedron Letters*, 2007, **48**, 3337–3341.

(39) N. A. Marine, S. A. Klein and J. D. Posner, "Partition Coefficient Measurements in Picoliter Drops Using a Segmented Flow Microfluidic Device", *Analytical Chemistry*, 2009, **81**, 1471–1476.

(40) Y. Dohta, T. Yamashita, S. Horiike, T. Nakamura and T. Fukami, "A System for LogD Screening of 96-Well Plates Using a Water-Plug Aspiration/Injection Method Combined with High-Performance Liquid Chromatography-Mass Spectrometry", *Analytical Chemistry*, 2007, **79**, 8312–8315.

(41) R. Mannhold, G. I. Poda, C. Ostermann and I. V. Tetko, "Calculation of molecular lipophilicity: State-of-the-art and comparison of log P methods on more than 96,000 compounds", *Journal of Pharmaceutical Sciences*, 2009, **98**, 861–893.

(42) A. K. Ghose and G. M. Crippen, "Atomic Physicochemical Parameters for Three-Dimensional-Structure-Directed Quantitative Structure-Activity Relationships. 2. Modeling Dispersive and Hydrophobic Interactions", *Journal of Chemical Information and Computer Sciences*, 1987, **27**, 21–35.

(43) T. Cheng, Y. Zhao, X. Li, F. Lin, Y. Xu, X. Zhang, Y. Li, R. Wang and L. Lai, "Computation of octanol-water partition coefficients by guiding an additive model with knowledge", *Journal of Chemical Information and Modeling*, 2007, **47**, 2140–2148.

(44) A.J. Leo, *CLOGP*, version 4.9, Daylight Chemical Information, California, 2011.

(45) A. K. Ghose, V. N. Viswanadhan and J. J. Wendoloski, "Prediction of hydrophobic (lipophilic) properties of small organic molecules using fragmental methods: An analysis of ALOGP and CLOGP methods", *Journal of Physical Chemistry A*, 1998, **102**, 3762–3772.

(46) M. H. Abraham, "Scales of solute hydrogen-bonding: their construction and application to physicochemical and biochemical processes", *Chemical Society Reviews*, 1993, **22**, 73–83.

(47) I. Moriguchi, S. Hirono, Q. Liu, I. Nakagome and Y. Matsushita, "Simple Method of Calculating Octanol/Water Partition Coefficient.", *Chemical & Pharmaceutical Bulletin*, 1992, **40**, 127–130.

(48) A. Mauri, V. Consonni, M. Pavan and R. Todeschini, "DRAGON software: An easy approach to molecular descriptor calculations", *Communications in Mathematical and in Computer Chemistry*, 2006, **56**, 237–248.

(49) Y. C. Lo, S. E. Rensi, W. Torng and R. B. Altman, "Machine learning in chemoinformatics and drug discovery", *Drug Discovery Today*, 2018, **23**, 1538–1546.

(50) I. V. Tetko and V. Y. Tanchuk, "Application of associative neural networks for prediction of lipophilicity in ALOGPS 2.1 program", *Journal of Chemical Information and Computer Sciences*, 2002, **42**, 1136–1145.

(51) I. V. Tetko, "Neural network studies. 4. Introduction to associative neural networks", *Journal of Chemical Information and Computer Sciences*, 2002, **42**, 717–728.

(52) K. S. Rogers and A. Cammarata, "A molecular orbital description of the partitioning of aromatic compounds between polar and nonpolar phases", *BBA - Biomembranes*, 1969, **193**, 22–29.

(53) G. Klopman and L. D. Iroff, "Calculation of partition coefficients by the charge density method", *Journal of Computational Chemistry*, 1981, **2**, 157–160.

(54) N. Bodor, Z. Gabanyi and C. K. Wong, "A New Method for the Estimation of Partition Coefficient", *Journal of the American Chemical Society*, 1989, **111**, 3783–3786.

(55) H. Chuman, A. Mori, H. Tanaka, C. Yamagami and T. Fujita, "Analyses of the partition coefficient, log P, using ab initio MO parameter and accessible surface area of solute molecules", *Journal of Pharmaceutical Sciences*, 2004, **93**, 2681–2697.

(56) A. Klamt and G. Schüürmann, "COSMO: A new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient", *Journal of the Chemical Society, Perkin Transactions 2*, 1993, **0**, 799–805.

(57) M. Işık, T. D. Bergazin, T. Fox, A. Rizzi, J. D. Chodera and D. L. Mobley, "Assessing the accuracy of octanol-water partition coefficient predictions in the SAMPL6 Part II logP Challenge", *Journal of Computer-Aided Molecular Design*, 2020, **34**, 335–370.

(58) D. Guan, R. Lui and S. Matthews, "LogP prediction performance with the SMD solvation model and the M06 density functional family for SAMPL6 blind prediction challenge molecules", *Journal of Computer-Aided Molecular Design*, 2020, DOI: 10.1007/s10822-020-00278-1.

(59) P. Patel, D. M. Kuntz, M. R. Jones, B. R. Brooks and A. K. Wilson, "SAMPL6 logP challenge: machine learning and quantum mechanical approaches", *Journal of Computer-Aided Molecular Design*, 2020, DOI: 10.1007/s10822-020-00287-0.

(60) S. Wang and S. Riniker, "Use of molecular dynamics fingerprints (MDFPs) in SAMPL6 octanol–water log P blind challenge", *Journal of Computer-Aided Molecular Design*, 2019, DOI: 10.1007/s10822-019-00252-6.

(61) S. Riniker, "Molecular Dynamics Fingerprints (MDFP): Machine Learning from MD Data to Predict Free-Energy Differences", *Journal of Chemical Information and Modeling*, 2017, **57**, 726–741.

(62) W. Koch and M. C. Holthausen, *A Chemist's Guide to Density Functional Theory*, Wiley, New Jersey, 2001.

(63) A. R. Leach and A. R. Leach, *Molecular modelling: principles and applications*, Pearson education, New York, 2001.

(64) E. Schrödinger, "An undulatory theory of the mechanics of atoms and molecules", *Physical Review*, 1926, **28**, 1049–1070.

(65) M. Born and R. Oppenheimer, "Zur Quantentheorie der Molekeln", *Annalen der Physik*, 1927, **389**, 457–484.

(66) W. Pauli, "Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren", *Zeitschrift für Physik*, 1925, **31**, 765–783.

(67) P. O. Löwdin, "Quantum theory of many-particle systems. III. Extension of the Hartree-Fock scheme to include degenerate systems and correlation effects", *Physical Review*, 1955, **97**, 1509–1520.

(68) L. H. Thomas, "The calculation of atomic fields", *Mathematical Proceedings of the Cambridge Philosophical Society*, 1927, **23**, 542–548.

(69) E. Fermi, "Un metodo statistico per la determinazione di alcune priorieta dell'atome", *Rend. Accad. Naz. Lincei*, 1927, **6**, 32.

(70) C. J. Cramer, *Essentials of Computational Chemistry*, John Wiley & Sons, Ltd, West Sussex, England, 2nd, 2004, ch. 11, pp. 429–456.

(71) T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, New York, 2009.

(72) J. Gasteiger and T. Engel, *Chemoinformatics: a textbook*, Wiley-VCH, New Jersey, 2003, p. 649.

(73) R. Todeschini and V. Consonni, *Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references*, John Wiley & Sons, New Jersey, 2009, vol. 41.

(74) H. Moriwaki, Y. S. Tian, N. Kawashita and T. Takagi, "Mordred: A molecular descriptor calculator", *Journal of Cheminformatics*, 2018, **10**, 4.

(75) D. Weininger, "SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules", *Journal of Chemical Information and Computer Sciences*, 1988, **28**, 31–36.

(76) D. L. Mobley and J. P. Guthrie, "FreeSolv: A database of experimental and calculated hydration free energies, with input files", *Journal of Computer-Aided Molecular Design*, 2014, **28**, 711–720.

(77) P. Winget, D. M. Dolney, D. J. Giesen, C. J. Cramer and D. G. Truhlar, "Minnesota solvent descriptor database", *Dept. of Chemistry and Supercomputer Inst., University of Minnesota, Minneapolis, MN*, 1999, **55455**.

(78) M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C.

Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman and D. J. Fox, *Gaussian~16 Revision C.01*, Gaussian Inc., Connecticut, 2016.

(79) D. R. Hartree, "The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods", *Mathematical Proceedings of the Cambridge Philosophical Society*, 1928, **24**, 89–110.

(80) R. Peverati and D. G. Truhlar, "Improving the accuracy of hybrid meta-GGA density functionals by range separation", *Journal of Physical Chemistry Letters*, 2011, **2**, 2810–2817.

(81) A. D. Becke, "Density-functional thermochemistry. III. The role of exact exchange", *The Journal of Chemical Physics*, 1993, **98**, 5648–5652.

(82) C. Lee, W. Yang and R. G. Parr, "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density", *Physical Review B*, 1988, **37**, 785–789.

(83) R. Ditchfield, W. J. Hehre and J. A. Pople, "Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules", *The Journal of Chemical Physics*, 1971, **54**, 724–728.

(84) A. V. Marenich, C. J. Cramer and D. G. Truhlar, "Universal solvation model based on solute electron density and on a continuum model of the solvent defined by the bulk dielectric constant and atomic surface tensions", *Journal of Physical Chemistry B*, 2009, **113**, 6378–6396.

(85) Scikit-learn Developers, "Scikit-learn: Machine learning in python", *Journal of Machine Learning Research*, 2011, **12**, 2825–2830.

(86) P. Probst, M. Wright and A.-L. Boulesteix, "Hyperparameters and Tuning Strategies for Random Forest", *WIREs Data Mining and Knowl Discov.*, 2019, DOI: `https://doi.org/10.1002/widm.1301`.

(87) C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings", *Advanced Drug Delivery Reviews*, 2001, **46**, 3–26.

(88) G. Duarte Ramos Matos, D. Y. Kyu, H. H. Loeffler, J. D. Chodera, M. R. Shirts and D. L. Mobley, "Approaches for Calculating Solvation Free Energies and Enthalpies Demonstrated with an Update of the FreeSolv Database", *Journal of Chemical and Engineering Data*, 2017, **62**, 1559–1569.

(89) B. E. Lang, "Solubility of water in octan-1-ol from (275 to 369) K", *Journal of Chemical and Engineering Data*, 2012, **57**, 2221–2226.

(90) A. Ben-Naim, *Solvation Thermodynamics*, Springer US, New York, 1987.

(91) R. K. Sinnott and G. Towler, *Chemical Engineering Design*, Elsevier Ltd, London, 2013.

(92) C. H. Yu, C. H. Huang and C. S. Tan, "A review of CO2 capture by absorption and adsorption", *Aerosol and Air Quality Research*, 2012, **12**, 745–769.

(93) M. Wang, A. Lawal, P. Stephenson, J. Sidders and C. Ramshaw, "Post-combustion CO2 capture with chemical absorption: A state-of-the-art review", *Chemical Engineering Research and Design*, 2011, **89**, 1609–1624.

(94) R. M. Davidson and IEA Coal Research. Clean Coal Centre., *Post-combustion carbon capture from coal fired plants : solvent scrubbing*, IEA Clean Coal Centre, London, 2007, p. 58.

(95) K. Goto, K. Yogo and T. Higashii, "A review of efficiency penalty in a coal-fired power plant with post-combustion CO2 capture", *Applied Energy*, 2013, **111**, 710–720.

(96) K. Armstrong and P. Styring, "Assessing the Potential of Utilization and Storage Strategies for Post-Combustion CO2 Emissions Reduction", *Frontiers in Energy Research*, 2015, **3**, 8.

(97) E. L. Ratkova, D. S. Palmer and M. V. Fedorov, "Solvation Thermodynamics of Organic Molecules by the Molecular Integral Equation Theory: Approaching Chemical Accuracy", *Chemical Reviews*, 2015, **115**, 6312–6356.

(98) Aspen Technology, *Aspen Plus 11*, version 11, Massachusetts.

(99) A. Fredenslund, R. L. Jones and J. M. Prausnitz, "Group-contribution estimation of activity coefficients in nonideal liquid mixtures", *AIChE Journal*, 1975, **21**, 1086–1099.

(100) D. L. Mobley, K. L. Wymer, N. M. Lim and J. P. Guthrie, "Blind prediction of solvation free energies from the SAMPL4 challenge", *Journal of Computer-Aided Molecular Design*, 2014, **28**, 135–150.

(101) T. Brouwer and B. Schuur, "Model Performances Evaluated for Infinite Dilution Activity Coefficients Prediction at 298.15 K", *Industrial and Engineering Chemistry Research*, 2019, DOI: `10.1021/acs.iecr.9b00727`.

Word Count

```
File: main.tex
Encoding: utf8
Sum count: 23986
Words in text: 22302
Words in headers: 148
Words outside text (captions, etc.): 1114
Number of headers: 61
Number of floats/tables/figures: 35
Number of math inlines: 330
Number of math displayed: 92
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  0+10+0 (1/0/0/0) _top_
  158+1+0 (1/0/0/0) Chapter: Abstract
  0+1+0 (1/0/0/0) Chapter: Introduction
  487+2+0 (1/0/0/0) Section: Thesis Motivation
  374+2+0 (1/0/1/0) Section: Thesis Layout
  0+2+0 (1/0/0/0) Chapter: Literature Review} \label{lit:chap:lit_review
  98+4+23 (1/1/0/0) Section: The Importance of LogP} \label{lit:sect:importance
  472+2+85 (1/3/0/0) Subsection: Pharmaceutical Industry} \label{lit:sect:pharm
  610+3+24 (1/1/0/0) Subsection: Environmental and Agrochemical
  471+4+10 (1/1/0/0) Section: Laboratory Measurement of LogP}\label{lit:sect:la
  229+2+0 (1/0/0/0) Section: Cheminformatical Methods} \label{lit:sect:comp_met
  180+1+0 (1/0/1/0) Subsection: Atomistic
  82+1+0 (1/0/0/0) Subsection: Fragment
  423+4+0 (3/0/1/0) Subsection: Property Based
  322+3+0 (1/0/1/0) Section: Quantum Mechanical Measurement
  1230+7+14 (1/1/5/1) Section: Quantum Mechanical and Machine Learning Method C
  0+1+0 (1/0/0/0) Chapter: Theory
  176+4+0 (1/0/13/5) Section: Quantum Mechanics\autocite{Koch2001ATheory, leach
  186+1+0 (1/0/9/4) Subsection: Operators
```

340+2+0 (1/0/4/5) Subsection: Born-Oppenheimer Approximation
242+2+0 (1/0/8/2) Subsection: Variational Principle} \label{theory:sect:varia
768+2+0 (1/0/41/13) Subsection: Hartree-Fock Approximation} \label{theory:sec
204+2+20 (1/1/7/1) Subsection: Electron Correlation
311+2+0 (1/0/13/4) Subsection: Electron Density
330+2+0 (1/0/28/4) Subsection: Hohenberg-Kohn Theorems
435+2+0 (1/0/14/12) Subsection: Kohn-Sham Approach
309+1+0 (1/0/9/9) Subsection: Functionals
82+2+0 (1/0/0/1) Subsection: Hybrid Functionals
302+2+0 (1/0/7/1) Subsection: Basis Sets
644+7+17 (3/1/17/6) Subsection: Solvent Models\autocite{Cramer2004Essentials
182+3+0 (1/0/0/0) Section: Machine Learning \autocite{hastie2009elements}
111+2+0 (1/0/0/0) Subsection: Test-Train Split
473+2+17 (1/2/30/4) Subsection: Prediction Assessment} \label{theory:sect:pre
127+1+0 (1/0/4/1) Subsection: Bagging
375+2+16 (1/1/23/6) Subsection: Random Forest
277+5+10 (1/1/3/0) Section: Molecular Descriptor Calculation \autocite{Gaste
0+2+0 (1/0/0/0) Chapter: Computation Method
86+1+0 (1/0/1/0) Section: Database
64+3+24 (1/1/7/1) Section: First Principles Calculations} \label{comp:sect:FP
122+2+0 (1/0/0/0) Section: Quantum Mechanical
64+2+0 (1/0/0/0) Section: Molecular Descriptors
583+2+0 (1/0/2/0) Section: Machine Learning
2+3+0 (1/0/0/0) Chapter: Results and Discussion
317+1+31 (1/3/0/0) Section: Database} \label{randd:sect:database
351+5+96 (1/3/5/0) Section: Calculation via Experimental Free Energies
3002+2+463 (1/8/22/0) Section: Quantum Mechanical} \label{randd:sect:QM
2865+2+232 (1/5/9/0) Section: Machine Learning
147+2+0 (1/0/0/0) Chapter: Chemical Engineering
134+4+0 (1/0/6/2) Section: In the Engineering Context
291+3+32 (1/2/14/7) Subsection: Absorption/Stripping\autocite{Sinnott2013Chem
719+2+0 (1/0/14/0) Subsection: In practice
0+4+0 (1/0/0/0) Section: Solvation Free Energy Computation
188+1+0 (1/0/4/1) Subsection: Experimental
1192+2+0 (1/0/7/2) Subsection: Computational Calculation
1119+4+0 (1/0/0/0) Chapter: Conclusions and Further Work
0+1+0 (1/0/0/0) Chapter: Appendix
46+4+0 (1/0/0/0) Section: Scripts and Raw Data

Word count of the biblography could not be computed using LaTeX, the biblography was loaded into Microsoft Word which returned a word count of 2942. This brings the combined word count of the thesis to 26935.

Appendix

Scripts and Raw Data

Due to the nature of the COVID situation at the time of writing this thesis, a physical copy of the scripts and raw data files could not be supplied. They can therefore be acquired at request from my project supervisor David Palmer.

Random Forest with Optimisation & RFE Script

```
 1 import pandas as pd
 2 import numpy as np
 3
 4 # Using Skicit-learn, import train test split, RF, random
   cross validation optimisation, recursive feature
   elimination, and metrics
 5 from sklearn.model_selection import train_test_split
 6 from sklearn.ensemble import RandomForestRegressor as rfr
 7 from sklearn.model_selection import RandomizedSearchCV as
   rcv
 8 from sklearn.feature_selection import RFECV
 9 from sklearn import metrics
10
11 # Seed for reproducable results
12 seed = 121212
13 np.random.seed(seed)
14
15 # Defining variables for averages
16 avgr2 = 0
17 avgrmse = 0
18 avgmae = 0
19 avgsdep = 0
20
21 # Defining hyperparameters
22 # max number of features to consider at each split
23 max_features = ['sqrt', 'log2', 0.333]
24 # max number of levels in tree
25 max_depth = [2, 3, 5, 8, 10, 13, 15, 20]
26 # Minimum number of samples to spilt a node
27 min_samples_split = [2, 3, 4, 5, 7, 10]
28 # Minimum number of samples required at each leaf node
29 min_samples_leaf = [1, 2, 3, 5]
30
31 grid_param = {'max_features': max_features,
32               'max_depth': max_depth,
33               'min_samples_split': min_samples_split,
34               'min_samples_leaf': min_samples_leaf}
35
36 # Set how many times the RF model will run
37 runtimes = 100
38
39 # Set up dataframe for storing data
40 stored_preds = pd.read_csv('predictions_template.csv')
41 stored_opt = pd.DataFrame()
42 stored_import = pd.DataFrame(pd.read_csv('features_template
   .csv'))
43 stored_metrics = pd.DataFrame()
44
45 # Running model for n times
```

```
for n in range(runtimes):
    # Read in data as pandas dataframe
    features = pd.read_csv('mordredRF.csv')

    # Label data
    labels = np.array(features['logP'])

    # Label features
    # Remove logP
    features = features.drop('logP', axis=1)

    # Saving feature names for later use
    feature_list = list(features.columns)

    # Convert to numpy array
    features = np.array(features)

    # Split the data into training and testing sets, 75/25
split
    train_features, test_features, train_labels,
test_labels = train_test_split(features, labels, test_size=
0.25, random_state=seed * n)

    # Save test feature ID's
    test_names = test_features[:, 0]
    # Drop ID's from features
    test_features = np.delete(test_features, 0, 1)
    train_features = np.delete(train_features, 0, 1)
    feature_list = np.delete(feature_list, np.where(
feature_list == "logP"))
    feature_list = np.delete(feature_list, np.where(
feature_list == "ID"))
    test_features = pd.DataFrame(test_features, columns=
feature_list)

    # Instantiate model
    rf = rfr(n_estimators=500, random_state=seed * n)

    # Run optimisation
    rf_opt = rcv(estimator=rf, param_distributions=
grid_param, n_iter=1000, cv=5, verbose=1, random_state=seed
 * n, n_jobs=-1, scoring='neg_mean_squared_error')

    # Train model on training data
    rf_opt.fit(train_features, train_labels)

    # Update model with optimised hyperparameters
    best_params = rf_opt.best_params_
    rf.set_params(**best_params)
```

```
 87
 88      # Saving optimisation results
 89      best_params = pd.Series(best_params)
 90      best_params = best_params.to_frame().transpose()
 91      stored_opt = stored_opt.append(best_params)
 92
 93      # Recursive feature elimination cross validation with
    optimised hyperparameters, first with 100 removed per step
     till at least 400 remain... until one-by-one removal
 94      rf_opt_RFEC = RFECV(estimator=rf, step=100,
    min_features_to_select=400, cv=5, scoring='
    neg_mean_squared_error', verbose=1, n_jobs=-1)
 95      train_features = rf_opt_RFEC.fit_transform(
    train_features, train_labels)
 96      feature_list = feature_list[rf_opt_RFEC.support_]
 97
 98      rf_opt_RFEC = RFECV(estimator=rf, step=25,
    min_features_to_select=200, cv=5, scoring='
    neg_mean_squared_error', verbose=1, n_jobs=-1)
 99      train_features = rf_opt_RFEC.fit_transform(
    train_features, train_labels)
100      feature_list = feature_list[rf_opt_RFEC.support_]
101
102      rf_opt_RFEC = RFECV(estimator=rf, step=10,
    min_features_to_select=100, cv=5, scoring='
    neg_mean_squared_error', verbose=1, n_jobs=-1)
103      train_features = rf_opt_RFEC.fit_transform(
    train_features, train_labels)
104      feature_list = feature_list[rf_opt_RFEC.support_]
105
106      rf_opt_RFEC = RFECV(estimator=rf, step=1, cv=5,
    scoring='neg_mean_squared_error', verbose=1, n_jobs=-1)
107      train_features = rf_opt_RFEC.fit_transform(
    train_features, train_labels)
108      feature_list = feature_list[rf_opt_RFEC.support_]
109
110      # Predict logP
111      rf.fit(train_features, train_labels)
112      test_features = test_features[feature_list]
113      predictions = rf.predict(test_features.values)
114
115      # Store predictions
116      pred = np.stack((test_names, predictions), axis=-1)
117      pred = pd.DataFrame(data=pred,
118                          index=np.array(range(1, 28)),
119                          columns=np.array(range(1, 3)))
120      pred = pred.rename(columns={1: "ID", 2: "Pred"})
121      stored_preds = pd.merge(stored_preds, pred, on="ID",
    how='left').fillna(0)
```

```
122
123     # Store recursive feature rankings and number of
    features remaining per split
124     import_feat = pd.DataFrame(feature_list).T
125     importance = pd.Series(rf.feature_importances_)
126     importance.name = 1
127     import_feat = import_feat.append(importance)
128     import_feat.columns = import_feat.iloc[0]
129     import_feat = import_feat.drop(import_feat.index[0])
130     frames = [stored_import, import_feat]
131     stored_import = pd.concat(frames)
132
133     # Metrics
134     r2 = metrics.r2_score(test_labels, predictions)
135     rmse = np.sqrt(metrics.mean_squared_error(test_labels
    , predictions))
136     mae = metrics.mean_absolute_error(test_labels,
    predictions)
137     sdep = rmse - (mae ** 2)
138
139     # Add to average metrics
140     avgr2 = avgr2 + r2
141     avgrmse = avgrmse + rmse
142     avgmae = avgmae + mae
143     avgsdep = avgsdep + sdep
144     metrics_values = pd.DataFrame([r2, rmse, mae, sdep]).T
145     stored_metrics = stored_metrics.append(metrics_values)
146     print('For run number', n, 'R2:', r2, 'RMSE:', rmse, '
    Bias:', mae, 'SDEP:', sdep)
147
148     # Store values to csv
149     stored_preds.to_csv("stored_predictions.csv")
150     stored_opt.to_csv("stored_opt.csv")
151     stored_import.to_csv("stored_import.csv")
152     stored_metrics.to_csv("stored_metrics.csv")
153
154 # Calculate average metrics
155 avgr2 = avgr2 / runtimes
156 avgrmse = avgrmse / runtimes
157 avgmae = avgmae / runtimes
158 avgsdep = avgsdep / runtimes
159
160 metrics_values = pd.DataFrame([avgr2, avgrmse, avgmae,
    avgsdep]).T
161 stored_metrics = stored_metrics.append(metrics_values)
162 print('Average: R2:', avgr2, 'RMSE:', avgrmse, 'Bias:',
    avgmae, 'SDEP', avgsdep)
163
164 # Final storage of values to csv
```

```
stored_preds.to_csv("stored_predictions.csv")
stored_opt.to_csv("stored_opt.csv")
stored_import.to_csv("stored_import.csv")
stored_metrics.to_csv("stored_metrics.csv")
```