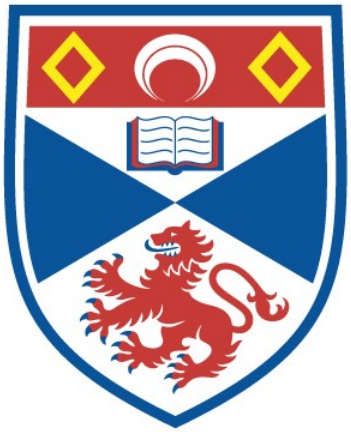
HYDRATE CRYSTAL STRUCTURES, RADIAL DISTRIBUTION FUNCTIONS, AND COMPUTING SOLUBILITY

Rachael Elaine Skyner

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2017

Hydrate Crystal Structures, Radial Distribution Functions, and Computing Solubility

Rachael Elaine Skyner

University of St Andrews

This thesis is submitted in partial fulfilment for the degree of PhD

at the

University of St Andrews

27th March 2017

For Dad, and in memory of Bill Skyner

The two greatest men I will ever know.

"How can you do chemistry on a computer?
Don't you need a lab, and chemicals?" *~ Bill Skyner, 2013*

1. Candidate's declarations:

I, **Rachael Elaine Skyner**, hereby certify that this thesis, which is approximately **36,000** words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in **September 2013** and as a candidate for the degree of **PhD** in **March 2017**; the higher study for which this is a record was carried out in the University of St Andrews between **2013** and **2017**.

I, **Rachael Elaine Skyner,** received assistance in the writing of this thesis in respect of **grammar and spelling**, which was provided by **Jonathan Colburn**.

Date: **27/03/2017** Signature of candidate: _____

2. Supervisor's declaration:

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of **PhD** in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date: **27/03/2017** Signature of supervisor: _____

3. Permission for publication:

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

PRINTED COPY
a) No embargo on print copy

ELECTRONIC COPY
a) No embargo on electronic copy

ABSTRACT AND TITLE EMBARGOES

If you have selected an embargo option indicate below if you wish to allow the thesis abstract and/or title to be published. If you do not complete the section below the title and abstract will remain embargoed along with the text of the thesis.

a) I agree to the title and abstract being published YES

Date: **27/03/2017** Signature of candidate: _____

Signature of supervisor: _____

Abstract

Solubility prediction usually refers to prediction of the intrinsic aqueous solubility, which is the concentration of an unionised molecule in a saturated aqueous solution at thermodynamic equilibrium at a given temperature. Solubility is determined by structural and energetic components emanating from solid-phase structure and packing interactions, solute–solvent interactions, and structural reorganisation in solution. An overview of the most commonly used methods for solubility prediction is given in **Chapter 1**.

In this thesis, we investigate various approaches to solubility prediction and solvation model development, based on informatics and incorporation of empirical and experimental data. These are of a knowledge-based nature, and specifically incorporate information from the Cambridge Structural Database (CSD).

A common problem for solubility prediction is the computational cost associated with accurate models. This issue is usually addressed by use of machine learning and regression models, such as the General Solubility Equation (GSE). These types of models are investigated and discussed in **Chapter 3**, where we evaluate the reliability of the GSE for a set of structures covering a large area of chemical space. We find that molecular descriptors relating to specific atom or functional group counts in the solute molecule almost always appear in improved regression models.

In accordance with the findings of **Chapter 3**, in **Chapter 4** we investigate whether radial distribution functions (RDFs) calculated for atoms (defined according to their immediate chemical environment) with water from organic hydrate crystal structures may give a good indication of interactions applicable to the solution phase, and justify this by comparison of our own RDFs to neutron diffraction data for water and ice. We then apply our RDFs to the theory of the Reference Interaction Site Model (RISM) in **Chapter 5**, and produce novel models for the calculation of Hydration Free Energies (HFEs).

Acknowledgements

First, and foremost, I would like to thank my supervisors, Dr. John Mitchell and Dr. Colin Groom. I would particularly like to thank them for the three-hour grilling that they gave me in June 2013 (otherwise known as an interview, apparently). Other than being great at extremely long interviews, they have also both been a massive support, professionally and personally, throughout my PhD. Without them, you wouldn't have to endure reading this thesis, as the work leading to it and the actual writing of it would not have been possible. I also have to give John extra-special thanks for his incredible proof-reading skills. I'm now a full convert to the Oxford comma.

Next on the list are my long-suffering family, who call me the eternal student (rude). Thank you, Dad, for financially supporting me when I've spent all of my money on food or alcohol over the last 8 years, and for listening to me, and for all of the small things that seem like nothing at the time, but mean everything. Thank you, Joanne for listening to me when I moan about work, and for looking after my Dad when I'm not there. Thank you to my younger brothers and sister for not understanding a single thing I do, forcing me to find other ways of having fun when I'm visiting. Thank you to my grandma (and to my grandad, to whom this thesis is dedicated), who also doesn't understand what I do, but constantly tells me she's proud, and reminds me that I'm the favourite grandkid, no matter what I do.

Thank you to all of the academics and collaborators who have spent hour upon hour with me, talking crystallography, informatics, and solubility. I am sure that if it weren't for a lot of those conversations and conference questions, the ideas presented in this work would never have popped in to my head, or developed in the way they did. Special mentions to: everyone at CCDC (thanks for the money, too!), Dr. David Palmer and Dr. Maxim Fedorov (Strathclyde), and Dr. Tasnim Munshi (Bradford/Lincoln).

Some final thanks go to all of my fellow-nerds (past and present) in 150, all of their respective supervisors, if not least for giving me a break at Thursday coffee, and also to my friends outside of the realms of Chemistry (i.e. inside of the realms of the fine drinking establishments of St Andrews). Particularly: My "old flat-husband" - Jose Garrido Torres, "Mum and Dad" - Dr. Rosie Alderson and Dr. James (Jimmy) McDonagh, "Daughter" – (almost Dr.) Ava Sih-Yu Chen, "Rain Pants" - Dr. Leo Holroyd, and "Lucy" – Dr. Luke Crawford, who have all been like an unconventional, extremely stressed out, and extraordinarily intelligent family to me during my time at St Andrews.

Table of Contents

Chapter 1
Introduction

*Parts of this chapter are published in; Skyner et al. Phys. Chem. Chem. Phys., 2015, **17**, 6174-6191*

1.1 Overview

Poor aqueous solubility is a major cause of attrition (failure) in the pharmaceutical development process and remains a vital property to quantify in the development of agrochemicals, and in the identification and quantification both of metabolites and of potential environmental contaminants. It is estimated that around 70% of pharmaceuticals in development are poorly soluble, with 40% of those currently approved also being poorly soluble[1,2]. Solubility is determined by structural and energetic components emanating from solid phase structure and packing interactions, in addition to relevant solute–solvent interactions and structural reorganisation in solution. This chapter focuses on the methods currently available to model the solution phase and to predict solubility for a wide range of applications, including ligand binding, molecular property prediction and molecular design[3].

Accurate and timely prediction of solubility could save time and money in drug development, agrochemical development and environmental monitoring. An early-stage analysis of drug and agrochemical candidates allows organisations to focus on those molecules most likely to meet their required solubility criteria. Many models exist in this area, with differing levels of accuracy, physical interpretability, and calculation time.

Quantitative Structure Activity Relationship (QSAR) and Quantitative Structure Property Relationship (QSPR) models are very successful in this field, providing good predictive results at a reasonably low computational cost. These models, however, tend to be limited to molecules similar to those used in their training set. Moreover, these models lack a full physical interpretation, although some do allow assessments of descriptor importance that can perhaps to some extent be physically interpreted.

Several fitted or derived general equations, which take only a few pieces of empirical data as arguments, have also been produced. One of the most successful is the General Solubility Equation

(GSE),[4] taking the melting point and the base ten logarithm of the partition coefficient (log *P*; partition coefficient for neutral molecules in octanol and water) as empirical input.

The field has also seen the revival of old ideas as new automated data driven design protocols, such as Matched Molecular Pair Analysis (MMPA)[5]. MMPA allows one to acquire previously 'unknown' data from existing data sets by exploring how a single molecular change can impact a particular property or activity of interest. We now see large scale data mining following these kinds of protocols, consortia such as SALT MINER, and programs developed by individual companies such as GSK's BioDig.[6,7]

The methods mentioned thus far are often the preferred choice for industrial investigation into solubility, for example for drug-candidate screening, as pharmaceutical companies are primarily interested in a 'rough-idea' of how soluble a compound is. However, if a precise value or perhaps a mechanistic view of solubility is required, physics based approaches to solubility may be preferably applied. These methods vary greatly in complexity.

Classical simulations can encompass simple Molecular Dynamics (MD), studying the interactions between solute and solvent, to more complex perturbations of solutes from the solution phase to the gas phase. Recent advances have seen a new generation of polarisable force fields emerging with a greater capacity to account for changes in the electronic charge distribution. Many of these force fields utilise multipole moments, as opposed to point charges, to capture the anisotropy of the atomic charge distribution. Force fields such as Atomic Multipole Optimised Energetics for Biomolecular Applications (AMOEBA) have been used to study the solvation dynamics of ions.[8] Newer, polarisable force fields, such as the Quantum Chemical Topology Force Field (QCTFF), use multipolar electrostatics calculated based on quantum chemical topology, supplemented with machine learning (Kriging) to model the system. This force field has been used to model amino acids with small water clusters.[9] Some force fields can be mixed with a quantum chemical core region in mixed Quantum Mechanics–Molecular Mechanics (QM/MM) approaches.

Other common models include those representing the solvent as a continuous field with no explicit solvent coordinates. In most cases, these models come at much higher computational cost than their informatics counterparts, and often at lower accuracy. However, if such a method were feasible and accurate enough to predict solubility, it would not have a domain of applicability restricted by the molecules within a training set and would also be physically interpretable. Thus, there is a continuing search for such physical methods. These methods have proven useful for modelling or approximating the solution phase, hence their applications are diverse and widespread outside of solubility prediction.

1.1.1 Thermodynamics and solubility

A solution is considered as an equilibrium between solute and solvent, reaching equilibrium when the number of molecules transferred from the solution to a non-solute state is equal to the transfer of molecules from a non-solute state to solution, *i.e.* when the forward rate is equal to the backward rate and both phases are in equilibrium. Solubility is a quantitative term, most simply describing the amount of a substance that will dissolve in a given amount of solvent, and is a property of thermodynamic equilibrium. A second process involved in solvation is dissolution; a kinetic term describing the rate at which a substance is transferred from a non-solute phase into

solution. Solubility and dissolution are fundamental terms describing the process of solvation, and are related by the Noyes–Whitney equation;[10]

$$\frac{dW}{dt} = \frac{kA(C_s - C)}{L} \quad [1.1]$$

where dW/dt is the rate of dissolution, A is the solute surface area in contact with the solvent, C is the instantaneous solute concentration in the bulk solvent, C_s is the diffusion layer solute concentration (given from the solubility of the molecule with the assumption that the diffusion layer is saturated), k is the diffusion coefficient, and L is the diffusion layer thickness.

As solubility is a thermodynamic term, it is inherently affected by factors such as temperature and pressure, as well as ionisation, solid state effects, and gaseous partial pressure for solvated gases.

pH is considered to have a significant effect on solubility, as many organic molecules can behave as weak acids or weak bases, due to ionisable basic or acidic functional groups, with polarisation of ionisable groups in solution increasing or decreasing the overall solubility. The pH of the aqueous solution in which such molecules are dissolved determines whether the molecule exists primarily in its neutral or ionised form. The charged form of a molecule is more soluble, and thus the aqueous solubility of a substance is pH-dependent.[11] This dependence is described by the Henderson–Hasselbalch (HH) equations as follows;

$$\log S_{total}^{acidic} = logS_0 + \log\left(1 + 10^{pH - pk_a}\right)$$

$$\log S_{total}^{basic} = logS_0 + \log\left(1 + 10^{pk_a - pH}\right) \quad [1.2]$$

where S_{total} is the equilibrium (thermodynamic) solubility, log S_0 is the intrinsic solubility, defined as the solubility of an unionised species in a saturated solution, pK_a is the negative logarithm of the ionisation constant of the molecule, and the final term on the right hand side is the solubility of the ionised form.[11] The HH relationship can be utilised in the prediction of pH-dependent aqueous solubility of drugs when the pK_a and log S_0 values of a compound are known.[12] The intrinsic solubility is a particularly important quantity as it can be used to find the pH dependent profile and estimate the pK_a; it is a quantity required by industry and hence the focus of several prediction methods.[13] The pH dependent profile of a drug is particularly important in pharmaceutics, as it has a direct effect on the absorption profile of a drug once it has entered the body. A basic drug-like molecule at a high pH (>2 pH units above the pK_a) will be almost fully unionised with solubility at a minimum (intrinsic solubility). Protonation of the base increases as pH becomes more acidic, and solubility increases. When pH and pK_a are equal, half of the solute molecules are protonated and the solubility of the drug becomes double the intrinsic solubility. According to the HH equation, this rise in solubility increases indefinitely with decreased pH, however in practice a limit is reached at the salt solubility. Two intersecting concentration curves for the base solubility and the salt solubility can be combined to give a composite curve for base solubility as a function of pH. If any one point on this curve is known (solubility and pH at which it was measured), the whole curve can be predicted providing pK_a and the acid solubility factor C_{0A}/C_{0B} (the ratio of S_0 of acid to S_0 of base) are known.[14]

Intermolecular interaction strengths play an important role in the solvation of substances from the solid state. Solutes which exhibit weak intermolecular forces (*i.e.* are weakly bound) tend to have a higher solubility, as the energy cost of breaking up the lattice is lower. Polymorphic effects can also lead to complications in solubility prediction. A classically cited example of this is the case of the anti-HIV drug Ritonavir,[15,16] in which a polymorphic shift led to a significant change in solubility, leaving the drug with a greatly reduced bio-availability. This exemplifies the consideration of solubility as a property which is dependent upon solid, solute, solvent, and solution state properties and interactions.

Two common approaches to the calculation of the Gibbs free energy of solution utilise a thermodynamic cycle approach. A first approach calculates the free energy of solution by addition of the free energy of sublimation (taking the molecule in the crystalline phase and subliming it into the gaseous phase) and free energy of solvation (taking the molecule in its gaseous phase and solvating it into aqueous solution). Examples of this approach are well cited within the literature.[13,17,18] A second approach involves calculation of the free energy of solution by addition of the free energy of fusion (taking a molecule from the crystalline state to a hypothetical supercooled liquid) and the free energy of transfer (transfer from a supercooled liquid into aqueous solution). This method is widely cited within the literature, and common GSE methods are also derived from this approach.[19]

The solid state is an important consideration for the initial crystalline phase calculated within thermodynamic cycle approaches. Lattice minimisation calculations and periodic DFT provide excellent tools for modelling these systems. Recent advances in these methods show promise for improving predictions, including updated codes and improved dispersion corrections in periodic DFT.[20,21]

Complete polymorphic screening and prediction still eludes our capabilities and hence hampers our ability to predict solubility from purely first principles. Polymorph screening refers to the practice of adjusting various experimental conditions in order to find a variety of polymorphs of the same molecular compound. Examples of these methods include: crystallisation from single or mixed solvents, seeding, and solid-state polymorphic transformations[22]. Thermodynamic stability of polymorphs is of particular interest, as the physical stability, and thus solubility, of the polymorphic form to be used in formulation is important. Thermodynamic terms can be determined through a variety of experimental methods. However, it is desirable for these terms to be computationally determined, in contrast to experimental polymorph screening. Therefore, polymorph prediction, in terms of crystal structure prediction studies, are often performed before experimental polymorph screening.

A further consideration is that of the standard states used in the different physical states. Typically, sublimation data is reported in a 1 atmosphere standard state. Solvation is typically quoted in the Ben-Naim standard state of 1 mol L^{-1} with a fixed centre of mass. The difference between the two standard states is a constant 1.89 kcal mol^{-1} (7.91 kJ mol^{-1}), calculated as $\Delta G_{atm} \rightarrow$ mol $L^{-1} = RT \ln(24.46)$, where 24.46 is the molar volume at ambient conditions.

The free energy of solution can be calculated directly by the following formula:

$$\Delta G_{solution} = -RT \ln(S_0 V_m)$$

$$\log(S_0 V_m) = \frac{\Delta G_{solution}}{2.303RT} \quad [1.3]$$

where S_0 is the intrinsic solubility V_m is the crystalline molar volume, R is the gas constant and T is the temperature in Kelvin (K).

1.2 Informatics – 'Smart' machines in solubility prediction

Informatics is the science of information processing, storage, and data mining. There are many applications and methodologies available for this type of task. Commonly used methods in chemistry are QSAR/QSPR models which are built from known data. These models correlate structural features of molecules with physical properties of interest. A major supposition of QSPR is that molecules similar in structure will have similar physical properties, and for QSAR models, perhaps chemical or biological similarities. Therefore, it is possible to train a model defining a specific relationship between structure and property/activity on a training dataset, and apply it to similar molecules to predict their properties and activities. For this reason, QSAR/QSPR models are not broadly applicable (*i.e.*, they cannot be applied to molecules differing considerably from the training set). While QSPR was once dominated by multiple linear regression, nowadays machine learning represents the state of the art. Both regression and machine learning protocols can identify these structure–property relationships by correlating structural features with experimentally determined physical data. A brief introduction to some of these methods is provided below, and for a more detailed account, see "An Introduction to Cheminformatics"[23] and references therein. Initially, one must represent a molecule in a machine-readable format to enable the calculation of molecular descriptors. Two of the most common methods for doing this are the Simplified Molecular Input Line Entry System (SMILES)[24] and the IUPAC International Chemical Identifier (InChI).[25]

1.2.1 Molecular descriptors

Descriptors represent physical, chemical, topological or energetic features of chemical structures, and can vary greatly in form and derivation. In general, a descriptor is a vector of single numerical values (features), each encoding specific information about an individual molecule.[26] This information can be a simple number, such as the molecular weight or the count of a specific atom type, or they can be a prediction of corresponding experimental quantities, such as the octanol–water partition coefficient (usually expressed as log *P*). Alternatively, they can also be derived from semi-empirical or quantum chemistry. Clearly the cost of calculating different descriptors can vary dramatically. It is often the case that descriptors offering higher levels of refinement, and therefore more useful molecular discrimination, incur a higher computational cost.[26] There are many different molecular descriptors and numerous pieces of software to calculate them.[26]

1.2.2 Methods

Regression. Regression analysis is a fundamental tool in informatics. Simple linear regression expresses a relationship between a scalar dependent variable Y and a single explanatory independent variable X. Multiple Linear Regression (MLR) extends this to allow for multiple dependent variables y_i or explanatory independent variables x_i, expressed as;

$$y = \sum_{i}^{j} \alpha_i x_i \qquad [1.4]$$

These methods have seen widespread use in many fields.[27] A disadvantage of MLR is the apparent ease of over-fitting. A useful rule of thumb is that the number of data points should be in excess of five times the number of explanatory variables (Fig. 1)[26,27]

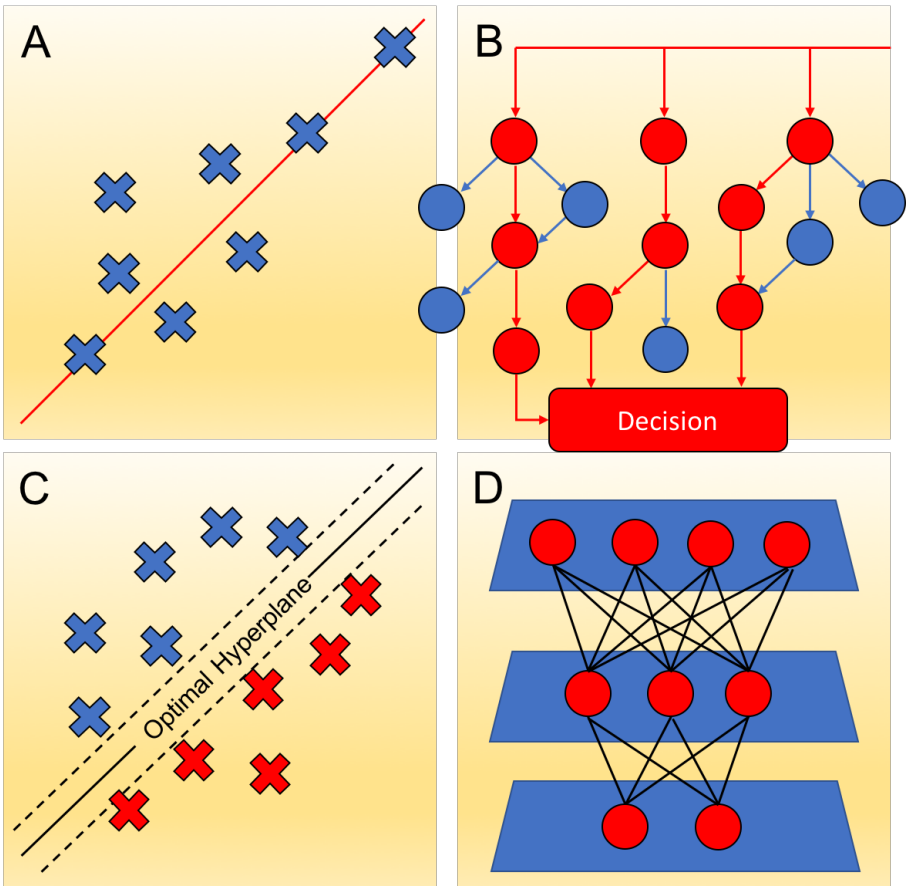


Figure 1- Machine learning methods; (a) regression analysis aims to describe how the typical value of the dependent variable changes as the independent variables are changed. The regression function (red line) characterises variation; (b) decision trees consisting of a binary separation at the nodes, leading to predictions or classifications at the leaf nodes (red circles); (c) an example of SVM separating data into distinct categories by an optimal hyperplane, which should have optimal margins either side for a clear distinction in data categorisation; (d) a typical neural network consists of layers of nodes. All nodes have connections with all other nodes in adjacent layers. The input units (top) do not count as a layer of nodes, as they do not carry out any typical arithmetic operations. A typical arithmetic operation is the generation of a net signal and transformation by a transfer function into an output signal. The input units distribute input values to all of the neurons in the layer below. The connections between nodes each have a different weight, representing different descriptors used in machine learning.

Random forest. Random Forest (RF), is a learning method based on decision or regression trees (depending on whether the predictive task requires classification or regression, respectively). These are stacked sets of binary separators following a tree like graph structure. RF uses a 'forest' of these decision trees, making use of "the wisdom of crowds"; hence, it is considered an ensemble learning method. For application to classification problems, the binary splitting is based upon the Gini index, which is a calculation of the maximal discrimination of the data points. For regression, splitting is generally based on a minimisation of the root mean squared error (RMSE). The initial node is known as the root node, with subsequent nodes being called branch nodes. The final nodes are referred to as leaf nodes and contain molecules with similar predictions of the property or activity (Fig. 1)[28].

Support vector machines. Another commonly used machine learning method is that of Support Vector Machines (SVM). SVM supports both regression and classification tasks, and is capable of handling multiple continuous and categorical variables. Methods for handling classification tasks are based on typically non-linear kernel functions. These kernel functions allow the transformation of data points into a higher dimensional feature space (Fig. 1).

SVM training algorithms are built up of binary categorised data, whereby a particular data point belongs to one of two categories. Thus, the test set data is also categorised, producing a clear separation, which should be as wide as possible, in the feature space. Alternatively, in the case of regression, the surface behaves analogously to a regression line, providing a maximal explanation of the data within the bounds of an acceptable error margin whilst attempting to remain relatively flat to avoid overfitting.[26,27]

Networks. Artificial Neural Networks (ANNs) and deep learning architectures are another common form of machine learning method in chemistry. These are models conceptually based on the brain's neuron network (although a great simplification). ANNs contain an input layer which receives the molecular information, an output layer which provides the prediction to the user, and between these at least one hidden layer which is trained using data to link the neurons of the input layer and output layer in a suitable fashion for the problem at hand. The training generally involves weighting specific paths between the neurons.[6,7,18] Deep learning algorithms attempt to abstract data on a high level through model architectures comprising multiple non-linear transformations. For example, in the case of ANNs the addition of hidden layers, which map some function of the input layer onto an output layer through a variety of unknown operations, can allow more information to be extrapolated from the input information.

1.2.3 The General Solubility Equation (GSE)

The GSE (as briefly mentioned in 1.1) is a QSPR model based on the melting point and the octanol–water partition coefficient log *P* of a chemical substance, used to predict the aqueous solubility of non-ionisable compounds,[29] and acts as a useful guide for ionisable compounds using lipophilicity (log *D*) at the pH of the aqueous buffer employed. The equation states that;

$$logS = 0.5 - 0.01(T_m - 25℃) - logP \qquad [1.5]$$

Or in terms of log *D*;

$$logS_{pH(x)} = 0.5 - 0.01(T_m - 25°C) - logD_{pH(x)} \quad [1.6]$$

The GSE is a simple QSPR model, with powerful predictive ability (coefficient of determination (r^2) = 0.96 and root mean squared error (RMSE) = 0.53 log *S* units for a data set of 1026 organic molecules[30]), and the simplicity of the model means it has found wide application in the pharmaceutical industry. However, the reliance of the GSE on experimentally determined descriptors limits its applicability, and datasets sparsely populated at their limits can lead to overestimation of the model's predictive power.[31]

Ali *et al.*[31] have revisited the GSE and have attempted to relieve the reliance of the GSE on the experimentally determined melting point by replacing it with the topological polar surface area (TPSA). They demonstrate the effects of inflated predictive power of the GSE by using a subset of an initial dataset, which reduced the overall predictive power of the GSE by approximately 6.4%. TPSA was included in a revised model to account for the fact that 88.5% of poorly performing compounds contained polarisable groups. The pure GSE model employed provided r^2 = 0.818, and the TPSA replacement of melting point model provided r^2 = 0.813, showing a comparable effectiveness. The number of compounds containing polarisable groups with log *S* predicted within ±1 log unit of experimentally determined values was also higher for the revised TPSA model (83.2% TPSA; 79.6% GSE). A final model combining melting point, log *P* and TPSA was also tested, and was found to have a better predictive power than both of the previously employed models (r^2 = 0.869) with 90.8% of compounds containing polarisable groups predicted within ±1 log unit of experimentally determined values.

The work of Ali *et al.*[31] highlights the importance of reliable descriptors in improving the overall performance of QSPR models, particularly when polar or polarisable functionality is included in test sets, and when experimentally determined values are required. As such, experimentally determined values may be best suited only for comparative analysis of predictive models to experimental data as a measure of performance in many cases.

1.3 Implicit solvation – the isotropic field as a solvent representation

Continuum solvation models consider solvent as a continuous isotropic medium. An underlying assumption of implicit solvation models is that explicit solvent molecules may be removed from the model, provided that the continuous medium replacing them sufficiently represents equivalent properties.

A simplification of continuum models can be thought of in terms of a Hamiltonian as;

$$\hat{\mathrm{H}}_{\mathrm{tot}}(r_{\mathrm{M}}) = \hat{\mathrm{H}}_{\mathrm{M}}(r_{\mathrm{M}}) + \hat{\mathrm{H}}_{\mathrm{MS}}(r_{\mathrm{M}}) \quad [1.7]$$

where *M* refers to a single solute molecule, *S* refers to the solvent, and *r* refers to position. Solvent coordinates do not appear within the Hamiltonian term, exemplifying the representation of solute

in a continuum, rather than as definite atoms, as with explicit models. $\hat{\mathrm{H}}_{\mathrm{MS}}$ is a sum of different interaction operators, which can be expressed in terms of solvent response functions, indicated by $Q_x(\vec{r}', \vec{r}')$, where $\vec{r}$ indicates a position vector, and *x* represents a contributing interaction.

In a standard continuum model, generally represented by Polarisable Continuum Models (PCM), solute–solvent interaction energies can be represented by a number of Q_x operators. The free energy of *M* is therefore described by an expression of five terms;

$$G(M) = G_{cav} + G_{el} + G_{dis} + G_{rep} + G_{tm} \quad [1.8]$$

with the order of terms corresponding to the best performing order of the 'charging processes', which are integration processes coupling a distribution function with a potential function. The terms are the free energy of cavitation, electrostatic energy, dispersion energy, repulsion energy and thermal fluctuation, respectively.

1.3.1 Continuum models for electrostatic interactions

PCM models are advantageous in that they can represent a statistically averaged (continuum) solvent so that meaningful results can be acquired within a single calculation. PCM models have been particularly useful in modelling reactivity and spectroscopy of various solvents with different polarities.[32]

In a solvent–solute system where atom *Q* (solute) has a positive charge, solvent water molecules will preferentially orientate their negative dipoles towards the solute's positive charge (Fig. 2, left). For a single water molecule, there is only a slight preference in orientation, which is smaller than that of its average thermal fluctuations. Therefore, this effect is averaged over the long range of electrostatic interactions of water in the bulk (Fig. 2, right). For an isotropic solvent with random thermal motion, the average electric field is zero at any given point. However, introduction of a solute gives a net change in orientation, introducing an overall change in electric field, known as the 'reaction field'.

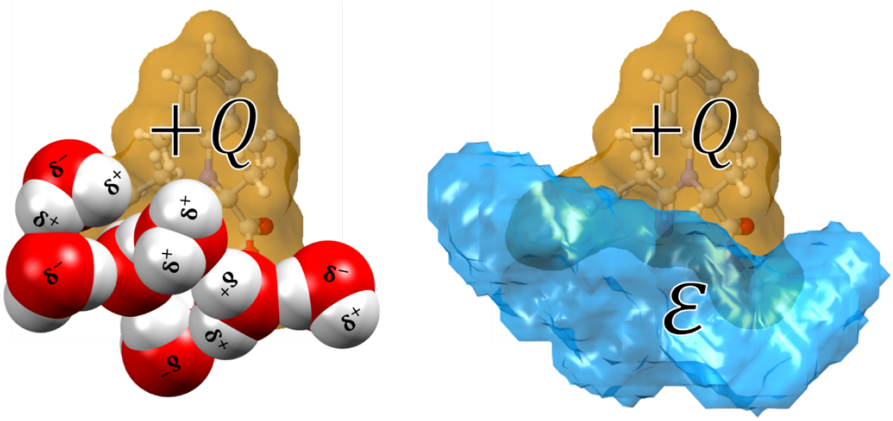


Figure 2 - (left) Water molecules reorient themselves to preferentially point the negative end of their dipole towards the positive solute charge (+*Q*). (right) The system is modelled with a continuous polarisable field. Polarisability is represented by the bulk dielectric constant, *ε*.

Accounting for the reaction field increases the solute's polarity proportionally to the solute polarisability, and the strength of the external electric field. This causes an increase in the dipole moment of *Q*, consequently polarising and increasing the change in orientation of the solvent to oppose the dipole moment of *Q*.[3]

There are energy costs associated with both the orientation and polarisation of the solvent, and the dipole moment of *Q*. As solvent molecules oppose the dipole moment of *Q*, they interact unfavourably with the reaction field. They also lose configurational freedom, with an associated free-energy cost. In a continuum model, the charge distribution of a solvent is represented as a continuous electric field, statistically averaged over all degrees of freedom at thermodynamic equilibrium. The electric field at any given point is the gradient of the electrostatic potential. The work required to create the charge distribution is determined from the interaction of solute charge density ρ with the electrostatic potential ϕ from;

$$G = \frac{1}{2}\int \rho(r)\phi(r)dr \qquad [1.9]$$

The polarisation component of *G* (G_P) is the difference between charging the system in gas and solution phases; thus only the electrostatic potentials in both gas and solution phases are needed to calculate G_P.

PCM methods are generally applied through two models; the Poisson–Boltzmann (PB) model, and the Generalised Born (GB) model. Both models are advantageous for different systems, and the accuracy of either model is mostly dependent upon the suitability of the cavity type used to surround the solute molecule within an ideal solvent system.

The Poisson–Boltzmann (PB) model. The Poisson equation combines the terms for electrostatic potential and the differential form of Gauss's law to define the electrostatic potential ϕ as a function of the dielectric constant ε and charge density ρ. When a surrounding dielectric medium responds linearly to an embedded charge, Poisson's equation states that;

$$\nabla^2\phi(r) = -\frac{4\pi\rho(r)}{\varepsilon} \qquad [1.10]$$

Continuum solvation models represent the charge distribution on the basis of two separate areas: inside (solute) and outside (solvent) of a cavity (Fig. 3). For this case, the Poisson equation states;

$$\nabla\varepsilon(r)\cdot\nabla\phi(r) = -4\pi\rho(r) \qquad [1.11]$$

The Poisson equation as expressed above is valid only for systems under non-ionic conditions. In a real solution, dissolving a solute produces mobile electrolytes. This effect is accounted for by an expansion of the Poisson equation, known as the Poisson–Boltzmann (PB) equation;

$$\nabla\varepsilon(r)\cdot\nabla\phi(r) - \varepsilon(r)\lambda(r)\frac{8\pi q^2 I}{\varepsilon k_B T}\frac{k_B T}{q}sinh\left[\frac{q\phi(r)}{k_B T}\right] = -4\pi\rho(r) \qquad [1.12]$$

where q gives the magnitude of electrolyte ionic charge, λ is a function equal to 0 in areas inaccessible to electrolyte ions and 1 for accessible areas, k_B is the Boltzmann constant, and I indicates the ionic strength of the electrolyte system.

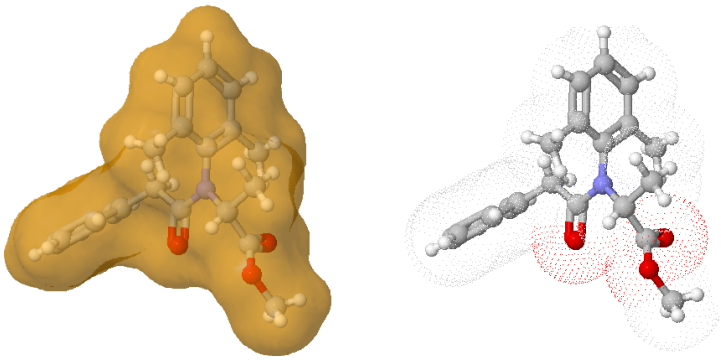


Figure 3 - An example of the solute cavity that may be calculated for a PCM calculation, represented by a solvent accessible surface area with a probe radius of 1.4Å (left) and an example of possible points for field evaluation, represented by a dot-surface (right)

PB equations are best used to calculate the electrostatic potential of systems where the cavitation of solute is near-spherical or ellipsoidal (ideal cavitation), as the convergence of the predicted electrostatic component of the solvation free energy ΔG_E is computationally expensive and often inaccurate. Thus, derivations applying approximations of the Poisson equation are often used in continuum models, the most common of which are Self-Consistent Reaction Field (SCRF) models,[32] such as the Onsager model.[33]

A further limitation of PB based models is the definition of cavitation. A number of variational SCRF models have been proposed in order to optimise cavitation parameters, most commonly using tessellation (tiling) of the cavity surface to simplify and reduce iterations of the PB equation.[32]

The Generalised Born (GB) model. For systems in which ideal cavitation is not accurate, arbitrary cavitation can be applied. Arbitrary cavitation refers to the construction of a cavity around the solute similar to the shape represented by space-filling models generated from the overlap of atomic spheres at volumes representing van der Waals (vdW) radii. An alternative method to SCRF models involves an approximation of the Poisson equation that can be analytically solved, known as the Generalised Born (GB) approach.

A conducting sphere with charge q can be considered representative of a monatomic ion. If the surface of the sphere is assumed to be entirely smooth, the charge distribution around it will be uniform, and the charge density at any point is given by;

$$\rho(s) = \frac{q}{4\pi a^2} \qquad [1.13]$$

where s is a point on the sphere's surface, and a is the spherical radius. Integrating over the entire outside surface and adding a term for the electrostatic potential, the energy term G, with $|r| = a$, becomes;

$$G = -\frac{1}{2}\int \left(\frac{q}{4\pi a^2}\right)\left(-\frac{q}{\varepsilon a}\right) ds = \frac{q^2}{2\varepsilon a}$$

[1.14]

The Born equation for the polarisation of a monatomic ion is calculated from the difference in the required work in the gas and solution phases applied to eqn. 1.14;

$$G_p = -1/2\left(1 - \frac{1}{\varepsilon}\right)\left(\frac{q^2}{a}\right)$$

[1.15]

The GB method extends the Born equation to polyatomic molecules to express polarisation energy as;

$$G_p = -1/2\left(1 - \frac{1}{\varepsilon}\right)\sum_{k,k'}^{atoms} q_k q_{k'} \gamma_{kk'}$$

[1.16]

where *k* and *k*′ run over all atoms, each with a partial charge *q*. The determination of suitable parameters for *γ* for polyatomic systems involves a radial integration of the charge *q* to determine the interaction of atom *k* with the surrounding medium. *γ* has units of reciprocal length, thus representing an inverse Coulomb integral. *γ* is given a suitable functional form in order to approximate the PB equation, and has a limiting behaviour, becoming closer to the exact reciprocal length r^{-1} at large interatomic distances.

1.3.2 Continuum models for non-electrostatic interactions

Similarly to the electrostatic components of solvation free energy, non-electrostatic contributions to the solvation free energy are not experimentally measurable. These contributions may have variable effects on the solubility of experimental systems. Various neutral model systems have been developed in accordance with this.

Specific component models. Pierotti[34] developed a model formula, based on scaled particle theory, for the calculation of cavitation free energy through the observation of the solvation energy for noble gases. Scaled particle theory is a statistical-mechanical theory of fluids derived from exact radial distribution functions, to give an expression for the work required to place a spherical particle into a fluid of spherical particles. Noble gas atoms do not exhibit permanent electrical moments, thus their transfer into solution is considered to be the most analogous example of perfect cavitation.

The experimental data from Pierotti's work has been complemented by simulation data,[35] including free energy of formation data of molecular-sized cavities in 12 common solvents obtained from free energy perturbation simulations. Pierotti's formula has since been expanded for molecular cavities by Colominas *et al.*[36]

A further, specific contributing factor to solvation free energy is dispersion. A somewhat simplistic explanation of dispersion is as follows. The average electron cloud of an atom is spherically symmetrical, but at any instantaneous time point there may be a polarisation of charge causing an instantaneous dipole moment. This dipole moment interacts with neighbouring atoms,

inducing a second instantaneous dipole, and so on, and an interaction occurs between these. The in-phase correlation of instantaneous and induced dipoles means the overall interaction energy does not average to zero over time.[3] The average interaction energy falls off (largely) proportionally to r^{-6} (where r is the distance between interacting particles). The multipole expansion of the dispersion interaction is written;

$$V(r) = \frac{C_6}{r^6} - \frac{C_8}{r^8} - \frac{C_{10}}{r^{10}} \dots \quad [1.17]$$

where C_6, C_8 and C_{10} are dispersion coefficients dependent on the atomic species. This is normally evaluated as a sum over all pairs of atoms in different interacting molecules.

Atomic surface tensions. Another approach for the evaluation of the non-electrostatic components of solvation free energy assumes the non-electrostatic component to be atom or group specific, and proportional to atomic surface area. A recent review by Wang *et al.*[37] (2009) considers four QSPR aqueous solubility models developed on the principle of weighted atom type counts and Solvent Accessible Surface Areas (SASA). They note that models considering SASA are often developed with small test-sets, and are therefore, in common with QSAR/QSPR models, poor performers for test molecules dissimilar to the original training set. The authors found that SASA descriptors did not enhance model performance any further than weighted atom type counts. This suggests the influences upon the non-electrostatic components of solvation free energy may be more complex than simple surface area considerations.

A further notable feature of continuum models based on surface tension is the neglect of any other contribution; that is, the development of these models assumes surface area as the sole determinant of solvation free energy, and that electrostatic components are implicit within the calculation parameters used.[32]

1.4 Explicit solvation models

Explicit solvation models are the primary choice of solubility models where solvent-specific effects are considered. The explicit treatment of water should, in principle, provide the most descriptive and realistic model for the investigation of solvation,[38] however it intrinsically requires a large number of degrees of freedom and thus is associated with a phase space of high dimensionality. This requires statistical averaging over the entire phase space, particularly when extracting specific underlying physical behaviour, such as thermodynamic properties.

Statistical thermodynamics relates all observable thermodynamic properties to the partition function, Q. The partition function is summarised as;

$$Q = \iint e^{-\frac{E(q,p)}{k_B T}} dq dp \quad [1.18]$$

where Q is the classical formulation integrated over all phase space of all spatial q and momentum p coordinates. Explicit models consider solvation in terms of free energy calculations, with different models for water available, as discussed below.

1.4.1 Free energy calculations – Monte Carlo (MC) and Molecular Dynamics (MD) simulations

Free energy considerations are distinctly different for intramolecular and intermolecular degrees of freedom. For intramolecular components, free energy contributions rely on vibrational and librational motions on an intramolecular energy surface.[39] For well-defined energy-minima, the free energy is easily accessible from the partition function (eqn. 1.18) from vibrational frequencies treated with the harmonic approximation. The harmonic approximation estimates the nuclear potential of a molecular system in its equilibrium geometry at a potential energy surface minimum in terms of normal vibrational modes, each governed by a 1D harmonic potential. Anharmonic effects are accounted for with MC or MD simulations for the calculation of entropy on the intramolecular energy surface.[39] Due to diffusion, the particles of a solution system do not exhibit motion definable by harmonic approximations. MC and MD simulations are restricted to only sampling the low-energy part of configuration space. Since internal energy and enthalpy are predominantly dependent on this low-energy region, they are well estimated. However, MC and MD methods do not involve the direct determination of *Q*, and exhibit an extremely slow convergence for densities of typical chemical systems, due to the exponential dependence of the Boltzmann factor on the energy, preferring the low-energy region. The high- and low-energy levels of molecular liquids are separated enough that typical MC and MD simulations will not sample the high-energy regions of configurational space necessary for an accurate calculation of the ensemble average of free energy[39].

Free Energy Perturbation (FEP) methods. Free Energy Perturbation (FEP) methods were first introduced by Zwanzig[40] in 1954, who related the thermodynamics of two different systems, in order to evaluate differences in intermolecular potentials. Zwanzig notes that at high temperatures, the forces of repulsion between molecules determine the equation of state of a gas, and that at lower temperatures the equation of state should be determinable by considering forces of attraction as perturbations on the forces of repulsion. The energy change from state A to state B is calculated by;

$$\begin{aligned}\Delta G(A \longrightarrow B) &= G_B - G_A \\ &= -k_B T \, ln \left[exp\left(-\frac{E_B - E_A}{k_B T} \right) \right]_A\end{aligned} \qquad [1.19]$$

where *T* is temperature, and the square brackets indicate an ensemble average over the simulation runs for A. A normal simulation run for A coincides with a new energy state of B on each optimisation run. The energy difference between A and B is either between the atoms in each state, or is an isomeric difference, for example A may be the *cis*-isomer of a structure, and B the *trans*-isomer, with A and B in different energy states due to different intra- and/or intermolecular interaction. For isomeric differences, the free energy map is calculated along a theoretical estimation of the reaction coordinates. The convergence of FEP calculations is only reliable for a small difference between A and B, thus traditional perturbation theory only holds true for systems which remain similar upon dissolution.

More recent derivations of Zwanzig's model allow the division of perturbations into smaller calculations, allowing parallelisation. These models involve breaking the reaction pathway down

into a series of intermediate transition state steps, allowing better convergence between the initial and final structures investigated.[41] However, FEP calculations remain one of the most computationally expensive methods for calculating free energy differences.

An example of this is shown by Lüder *et al.*[42] who have investigated the effectiveness of FEP methods for the calculation of free energy of solvation in pure melts for 46 drug molecules. Simulations were performed in two stages, scaling down the Coulomb and Lennard-Jones (LJ) interactions independently. Results were interpreted under the assumption that the free energy of the vapour to liquid process ΔG_{vl} can be calculated from the sum of the free energy term for cavitation ΔG_{cav} and the energy associated with LJ interactions and half of the Coulomb interaction term. ΔG_{cav} is obtained from hard-body theories. Interaction energies and molar volumes for each of the 64 drug molecules were compared for systems comprising 260 molecules. Deviations between systems were found to be an average of 2.9% for intermolecular interaction energy, and 1.4% for molar volume, suggesting the dataset selected would provide reliable results. Predicted and simulated ΔG_{cav} values were found to be systematically underestimated by approximately 15%. An overall average deviation of calculated ΔG_{vl} values in comparison to experiment is −1.8 kJ mol^{-1}, with reasonable errors expected in the range −1 to 1 kJ mol^{-1}. This investigation suggests that overall, FEP methods require more work at the theory level, particularly due to systematic errors that occur in phase space relationships between reference and perturbed systems.

An alternative approach to calculating the free energy difference from one state to another is to treat the change from A to B as a transformation, rather than to calculate free energies of independent structures, and calculate an energetic difference, as in traditional FEP methods.[3]

A recent application of this method, derived from FEP, has been demonstrated by Liu *et al.*[43] for the calculation of the solubility of gases in ionic liquids. The Bennett acceptance ratio (BAR) method utilises the method of transferring between states instead of treating each state as an individual structure. The Coulomb and LJ terms are calculated separately. It is found that simulated solubilities are found in good agreement with Henry's law constants. However, comparison to experimental data finds poorly soluble gases to have larger errors, with underestimated and overestimated gas solubilities found with similar calculation methods in complementary studies.

Enthalpy–entropy decomposition. A further offshoot of free energy calculations is the decomposition of the free energy term into enthalpic and entropic components.[38] As both enthalpy and entropy are experimentally measurable, the difference between theory and experiment is ascertainable, and may be applied as benchmarks for force field optimisations,[38] and give insight into the mechanism of solvation. Levy and Gallicchio have reviewed a variety of different approaches to the thermodynamic decomposition of free energies.[38]

Wyczalkowski *et al.*[44] recently proposed two new methods for the estimation of entropy and enthalpy decomposition of free energy calculations, evaluated for the solvation of *N*-methylacetamide (NMA). The methods investigated found thermodynamic contributions to be in disagreement with experimental data, highlighting the difficulty in obtaining decompositions comparable in quality to free energy estimates, with thermodynamic decomposition of computational Helmholtz free energies of solvation (ΔF at fixed volume) values yielding errors approximately two orders of magnitude larger than the initial ΔF values found. It is noted that ΔF

values are statistically reliable and can be used for quantitative comparison to experimental data. The calculation of entropic and enthalpic contributions is also extremely computationally demanding, as every temperature point of a simulation requires recalculation of the overall free energy.[3] Wyczalkowski *et al.* highlight that where calculation of free energies of solvation has advanced so that computational errors are on par with experimental ones, thermodynamic decomposition calculations suffer from statistical errors 10–100 times larger than free energy of solvation calculations.

A recent study by Ahmed and Sandler[45] uses the decomposition of free energies of hydration and self-solvation of low polarity nitrotoluenes to consider an array of thermodynamic terms and physiochemical properties. These include: solid-phase vapour pressures, solubilities, Henry's law constants, hydration and self-solvation entropies, enthalpies, heat capacities and enthalpies of vaporisation or sublimation. Their study focuses on the temperature-dependence of various terms. Decomposition of hydration free energies into enthalpic and entropic contributions is performed by a method utilising polynomial fitting of temperature-dependent self-solvation free energies (with respect to temperature). The use of fitting increases the sensitivity of derived values of hydration free energies. Self-solvation enthalpy (ΔH_{self}) values and entropy ($T\Delta S_{self}$) values are calculated within approximately 2 kcal mol^{-1} of experimentally determined values.

1.4.2 Combined Quantum Mechanical/Molecular Mechanical methodologies (QM/MM)

Explicit solvation models are often developed with respect to biological systems, due to the role of water in catalytic mechanisms, protein folding and protein–DNA recognition, to name but a few, which all require the specific detail of explicit water–substrate interactions to hold descriptive meaning. Of particular interest are combined QM/MM models, with QM describing electronic system changes (where precise system description is needed) and the rest of the system (where less precision is required) being described by a MM force field.[3] Applications of QM/MM combined models are discussed in a recent review.[46]

The foundational concepts involve the partitioning of a desired system into two subsystems: the QM subsystem, containing a small number of atoms and described by QM, with the remainder of the system described by a suitable MM force field. The Hamiltonian of the whole system is simply written;

$$H = H_{QM} + H_{MM} + H_{QM/MM} \qquad [1.20]$$

where H_{QM} is a QM Hamiltonian, H_{MM} is an empirical force field and $H_{QM/MM}$ describes interactions at the QM/MM interface. The energy of the system is also described as the sum of QM, MM and QM/MM contributions. This model is often referred to as a two-layered approach (Fig. 4, left). A derivative of this model involves adding a third “layer” as a continuum solvent representation around the MM region, and is known as a three-layered approach (Fig. 4, right).

Theoretically, any desired level of accuracy can be used within the QM region of the simulated system, within the scope of available methods. However, more accurate methods are susceptible to high computational cost. Thus, careful consideration is required by the user as to what level of accuracy is required, and at what cost. A succinct overview of different available QM methods is

provided by Friesner and Guallar[46] for QM/MM methods applied to enzymatic catalysis, with descriptions, advantages and disadvantages of respective QM methods available in textbooks such as the one by Cramer.[3]

A primary consideration when selecting a QM/MM method is the interactions at the QM/MM interface. Two aspects must be considered; (i) the presence of covalent bonds across the interface – a particular concern for large (*e.g.*, biomolecular) molecules, and (ii) the influence of the MM solvent region on the QM region – electrostatic and van der Waals interaction terms must be included.

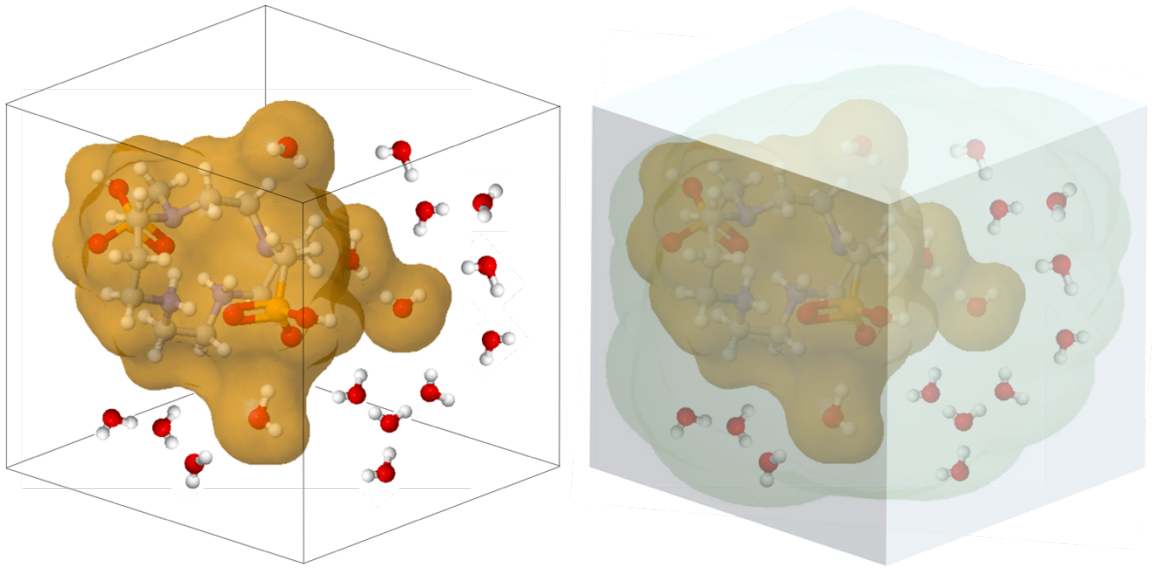


Figure 4 - (left) Two-layered approach to the QM/MM method. The solute molecule and a few water molecules are treated with QM (centre) and the rest of the solvent system is represented by MM up to a user-defined distance. (right) Three-layered approach – an additional layer surrounds the MM region and uses a continuum approach to describe the long-range solvent in the bulk.

In order to treat covalent bonds at the interface, it is possible to introduce "link atoms". Link atoms are QM hydrogen atoms that fill free valencies of QM atoms connected to MM atoms. A disadvantage of this method is the debate about inclusion of Coulombic interaction terms for the link atoms. Other methods developed in order to avoid the use of link atoms include the Local Self-Consistent Field (LSCF) method, which applies a mixture of hybrid and atomic orbitals to represent the QM system, and the "connection atom" method, where MM and QM interface atoms are described as QM methyl groups with a free sp^3 valence.

A recent three-layered approach aiming to tackle the issues associated with the QM/MM interface and the interaction terms for MM solvent effects has been proposed by Steindal *et al.*[47] This approach is described as the fully polarisable QM/MM/PCM method (see Section 3 for a description of PCM), and is designed for the effective inclusion of a medium in a QM calculation. Short range solvent electrostatic potentials are described by an atomistic model (QM/MM) whilst the long-range potentials are described by a continuum. The method is implemented in combination with linear response techniques with a non-equilibrium formulation of environmental response. The authors find a faster convergence with respect to system size for QM/MM/PCM than for QM/MM methods. This approach allows for reduction of the MM part of

the calculation with PCM, allowing less demanding calculations, and reduced sampling. However, three-layered approaches such as this often require much more user input and method manipulation, for example, considerations for MM/PCM interactions have to be considered in addition to QM/MM interactions, and so such methods are suited only to experts.

1.4.3 Explicit representations of water atoms

When solvent is represented explicitly, solvent molecules usually greatly outnumber solute molecules. Thus, in order for a model to be efficient, it is advantageous to use the simplest possible solvent representation.[48] Water is often considered the most useful solvent system, and thus is the solvent most widely used in explicit solvent models. The macroscopic properties are well established, yet the microscopic forces that determine water structure are not fully understood.

The treatment of water can be rigid or flexible. Rigid models often include a fictitious H–H bond to constrain bond angles in the water monomer.[3] Three of the most common rigid models for water are the TIP3P (transferable intermolecular potential 3P), SPC (simple point charge) and SPC/E (simple point charge extended) models, and their modified counterparts. These three models are effectively rigid pair potentials comprising LJ and Coulombic terms. However, the terms used differ in each model, and give rise to different calculated bulk properties for water.[48] Values for various properties of water obtained with different rigid models of water are shown below, in Table 1.

Table 1 - Model *vs.* experimental (exp.) values for bulk properties of water under standard conditions (298 K; 1 bar), including dipole μ, density ρ, static dielectric constant ε_0 and heat capacity C_p

Property	*TIP3P*[49,50]	*TIP4PEw*[51]	*SPC/E*[50,52]	*Exp.*[50]
μ *(D)*	2.348	2.32	2.352	2.5–3.0
ρ *(g cm^{-3})*	0.980	0.995	0.994	0.997
ε_0	94	63.90	68	78.4
C_P *(cal K^{-1} mol^{-1})*	18.74	19.2	20.7	18

MD calculations require the integration of Newton's equations of motion for all atoms, which is achieved through the evaluation of all atomic forces at each time step. Non-bonded interactions, especially long-range electrostatic interactions, dominate computationally, requiring extensive CPU time. In order to minimise this to an acceptable level, approximations are necessary. Boundaries are introduced into water models to restrain the system to a finite size, which almost always leads to artefacts in the obtainable data.[48] The most commonly utilised method for cost-effective solute computations is the application of a spherical cut-off, limiting the number of pairwise interactions to those within a specified radius.[48] The use of cut-offs for non-bonded interactions can have undesirable effects. LJ interactions are susceptible to small energetic effects, and large pressure effects induced by cut-offs. Pressure scaling can be used to correct for pressure related cut-off effects, usually to the order of several hundred bar. Cut-off effects for systems with dipolar electrostatic interactions are more prominent, with cut-offs selected within the parameters of experimental radial distribution functions up to ~10 Å. However, computer simulations have shown ordering within water up to ~14 Å, so the full structure of water is not typically accounted for, resulting in a poor description of dielectric properties. A further, and the most prominent, effect of cut-offs occurs in systems with full charges, where accumulation of the charge occurs at the cut-off boundary.[53]

Spoel *et al.*[53] (1998) investigated the effectiveness of TIP3P, TIP4P, SPC, and SPC/E models in describing the density and energy, dynamic, dielectric and structural properties of water. All simulations and analyses were identical for each model investigated, allowing the evaluation of simulation methodology independent of the model. It was found that system size, cut-off length and reaction fields had comparable effects on the overall calculated structural properties of water.

System size effects are considered through the comparison of systems comprising a small (216) and a large (820) number of molecules. The average thermodynamic properties (ρ, E_{pot}, T, P) are the same regardless of system size. Fluctuations in thermodynamic properties are known to be proportional to the square root of the system size, which is confirmed within the study. However, differences between large and small systems are observed, particularly for the dielectric constant, which is higher for all systems with a large number of molecules. The diffusion constant for large systems is also higher, attributed to periodic boundary conditions (PBC).

Cut-off effects are considered by the use of two different cut-off lengths (9 Å and 12 Å) for the large systems. It is found that density increases with an increased cut-off length, and energy decreases. There is no effect on dielectric behaviour.

In all simulations density is reduced, and the energy is decreased by approximately 1 kJ mol^{-1} on application of a reaction field. The self-diffusion constant D, and rotational correlation times were found to increase, indicating that the reaction field affects both the translational and rotational mobility of molecules.

Quantum chemical MD simulations of water are often developed with Density Functional Theory (DFT) methods, using either plane wave or atom-centred basis sets, to determine the electronic structure and forces. These methods offer reasonable estimates of the structural and dynamic properties of water when compared to experimental measurements. However, problems exist in the description of electronic gradient corrections, and equilibrium pressure. The interatomic forces of early quantum simulations, including DFT based methods, were originally parameterised with classical mechanics, leading to an unsatisfactory agreement between quantum and experimental results. DFT models also tend to calculate liquid structure with too much order, and underestimate equilibrium density. This is often attributed to the inability of local functionals to describe dispersion effects.

A recent approach to water simulation has claimed to provide a model, called the electronically coarse-grained model, capable of accounting for the shortcomings of both existing classical and quantum models.[54] Jones *et al.*[54] (2013) base their method on the replacement of valence electrons of an atom with an embedded Quantum Drude oscillator (QDO). QDO treatment of water is based upon the TIP4P classical rigid model of water, with the three water atoms supplemented by a dummy atom with a negative charge, added along the $\angle$HOH bisector to create an additional interaction point. The QDO parameters aim to reproduce the dipole and quadrupole polarisabilities, and the dispersion coefficient. The dispersion interaction is then adjusted by scaling, whilst preserving polarisability. The baseline unadjusted model produces a realistic, but over-structured liquid with a density that is too low by up to 20%, attributed to its underestimation of dispersion. Note also that the value of the enthalpy of vaporisation (at ambient pressure) ΔH_{vap} was found at 40 ± 2 kJ mol^{-1}, close to the experimental value of 43.91 kJ

mol^{-1}. Scaling the dispersion term results in an increased equilibrium density for increased dispersion. This induces a weakening effect on the H-bonding network of water, bringing the overall structure closer to agreement with benchmark data. However, the calculated ΔH_{vap} increases to 46 ± 2 kJ mol^{-1}, which is 4% higher than the experimental value. It is also found that the H-bond network is sensitive to changing polarisation at fixed dispersion, affirming the independent importance of both polarisation and dispersion effects on an overall explicit model.

1.5 Hybrid models

Within an aqueous solution phase, single snapshot images of structure are of limited use. Water is one of the few single component liquids for which there are highly competitive interactions at short range (hydrogen bonding), capable of damping the effects of repulsion. For this reason, ensemble averaging is required to identify the most probable geometric configurations which most heavily contribute to the system's interactions. This idea has already been introduced within explicit models of solvation, using ensembles taking snapshots at specific time periods. However, the cost of calculating the many configurations accessible in a solution is enormous. A number of methods, based on statistical mechanics, enable a more efficient calculation process.

1.5.1 Correlation functions

From a chemical point of view, a solution is a highly mobile system in which the dynamics are a vital contribution to the system's properties and behaviour. Therefore, mathematically we wish to capture this. Attempting to quantify dynamics with static properties is not sufficient; we must therefore provide averages or probabilities of interactions occurring at given distances. For this reason, a natural choice is to represent the solvent using Pair Correlation Functions (PCF), or equivalently Radial Distribution Functions (RDF). These functions allow us to determine a probabilistic structure of the solvent.

PCF can be interpreted as showing the probability against distance of there being an atom of interest at that distance from the atom under study. For example, the first large blue peak in Fig. 5 would correspond to either a water H at a distance from an O atom under study or *vice versa*. These functions are experimentally determinable from scattering experiments. We would expect that the PCF/RDF would go to a constant value of 1 at large values of *r* (*i.e.* it would become isotropic, like a continuum model, as there are no solute interactions to perturb the system). However, at small values of *r* we would not expect this. At very small values (less than the van der Waals radii of the solute atoms) we expect zero as only one particle can occupy the space at a time. Just outside this distance we see sharp non-uniform behaviour as solvent in the space interacts favourably with the solute holding a more rigid form. This leads to troughs in the PCF/RDF just behind the peaks, thus deviating from the value of 1 for a uniform solvent (Fig. 5).

Computational use and determination of correlation functions. The starting point for the use and determination of these functions for solvation modelling in statistical mechanics is integral equation theory (IET). In this theory, a molecule is fully described by a six-dimensional vector (three degrees of freedom relate to position x, y, z and three degrees of freedom determine the orientation ψ, θ, ϕ). To refer to these two sets of variables collectively, we will use the following symbols r = {x, y, z} and Θ = {ψ, θ, ϕ}. These variables are incorporated into the

fundamental 6D integral equation, the Molecular Ornstein–Zernike equation (MOZ). This equation utilises PCF/RDF between the various constituents of the liquid, $g(r_1, r_2, \Theta_1, \Theta_2)$. This simplifies for homogeneous solution to relative positions and orientation of the constituents, $g(r_1 - r_2, \Theta_1 - \Theta_2)$. This can most conveniently be written with reference to the total correlation function $h(r,\Theta)$.[55]

$$h_{ij}(r_1 - r_2, \Theta_1 - \Theta_2) = g_{ij}(r_1 - r_2, \Theta_1 - \Theta_2) - 1 \quad [1.21]$$

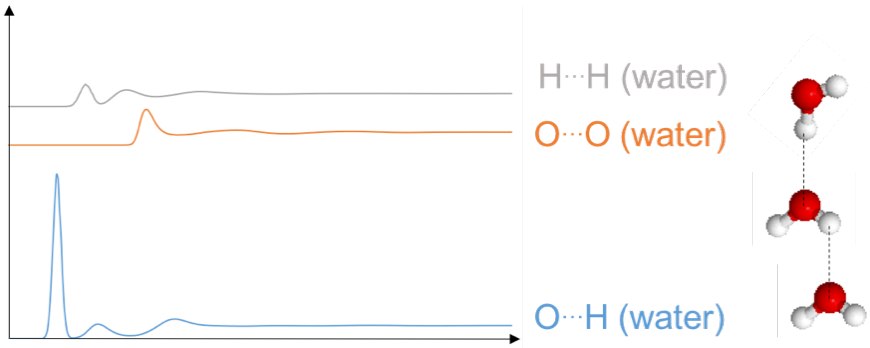


Figure 5 - A schematic representation of PCF for liquid water; water oxygen – water hydrogen (blue), water oxygen – water oxygen (orange) and water hydrogen – water hydrogen (grey).

We can simplify this equation by assuming spherical symmetry of molecules, hence removing consideration of orientational degrees of freedom by treating each water molecule as a hard sphere. This simplification leads to a 1 dimensional treatment of the integral, known as 1D-RISM (it is more accurate to treat the integral in 3D). We can now further separate the contributions to the total correlation function into direct and indirect components. To do this we must introduce the direct correlation function $c(r)$. We can now re-write the equation 1.21 assuming spherical symmetry as follows:

$$h(r_{1,2}) = c(r_{1,2}) + \int dr_3 c(r_{1,3})\rho(r_3)h(r_{2,3}) \quad [1.22]$$

Two effects contribute to the total correlation function (eqn. 1.22); (i) the direct correlation between r_1 and r_2, and (ii) an indirect correlation *via* a third body, r_3. The indirect correlation *via* r_3 is weighted by the density at r_3, and thus allows the consideration of all possible positions of the third body (Fig. 6).[56]

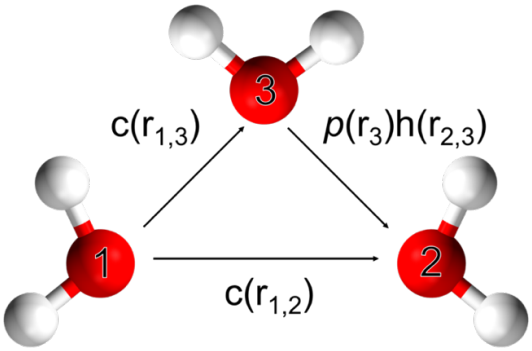


Figure 6 - Illustration of the contributions, both direct and indirect, to the total correlation function.

To solve this equation, $h(r)$ and $c(r)$ need to be found. As we have only a single equation and two unknown functions, $h(r)$ and $c(r)$, another equation is required; a closure relation must be introduced. There are several such equations available from statistical mechanics. The exact closure relation is as follows:

$$g(r) = e^{-\beta U(r)+h(r)-c(r)+B(r)} \Longrightarrow e^{-\beta U(r)+T(r)+B(r)} \quad [1.23]$$

where β is equal to $1/k_B T$ and $U(r)$ is the interaction potential which is often of the following form:

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma_{ab}}{r}\right)^{12} - \left(\frac{\sigma_{ab}}{r}\right)^{6}\right] + \frac{q_a q_b}{r} \quad [1.24]$$

where ε is the depth of the potential well, and σ is the finite distance for which the inter-particle potential is zero. $T(r)$ is known as the indirect correlation function as it is the difference between the total and direct correlation functions, and quantifies the indirect contribution. $B(r)$ is the bridge function, which comes from graph theory – its exact form is not known. Several approximate closure relations exist; some will be discussed here, although others are available. Originally the HyperNetted-Chain (HNC) approximate closure was used:

$$h(r) = e^{(-\beta U(r)+T(r))} - 1 \quad [1.25]$$

This closure works in principle for charged systems but neglects the bridge function term completely, assuming it to be zero. This can lead to poor convergence due to uncontrolled growth in the argument of the exponent. An alternative is the Partially Linearised Hyper-Netted Chain (PLHNC). This closure linearises the HNC once a cut off value (C) is exceeded;[17]

$$\Lambda = -\beta U(r) + T(r)$$

$$h(r) = \begin{cases} e^{(-\beta U(r)+T(r))} - 1 & when\ \Lambda \leq C \\ -\beta U(r) + T(r) + e^{C} - C - 1 & when\ \Lambda > C \end{cases} \quad [1.26]$$

This improves the convergence of the equations and is now regularly used in many applications for a variety of systems.

Due to the spherical symmetry approximation, the MOZ can only be applied to simple solutions. Additionally, due to the high dimensionality of the full equation, before the spherical symmetry approximation was invoked, it was practically incomputable. For this reason, a number of approximations have been developed which are collectively referred to as Reference Interaction Site Models (RISM).[57,58]

The simplification of the MOZ equation discussed above is a simplification used to alleviate some of the problems associated with the high dimensionality of the equation. These types of simplification originate from the work of Chandler and Anderson.[59] 1D RISM is an approach reducing the 6D MOZ equations to an approximation involving a set of 1D integral equations, treating the solvent as sets of sites with spherical symmetry. The main advantage of this is fast computational solution of the resulting integrals.[60]

Chapter 2

Theory & Methods

This thesis investigates the improvement of solubility prediction. The methods involved within the projects discussed herein are primarily focused on data-mining and informatics, utilising empirical data from a number of sources. The primary data for all of the methods used comes from the Cambridge Crystallographic Data Centre's (CCDC) Cambridge Structural Database (CSD). Other data comes from a variety of sources, which will be detailed with respect to each method in their own chapters. This chapter discusses the fundamental theories and methods explored and applied in this thesis.

2.1 Crystallography

The unit cell of a crystal structure contains a group of atoms with a fixed geometry relative to one another. The intrinsic highly ordered symmetry of crystal structures gives rise to geometrical and symmetrical relationships known as symmetry elements and operators. These operators are determined from the original diffraction pattern of a crystalline material during crystal structure solution and refinement. The symmetry properties of crystal structures are described by spacegroup notation. The spacegroup of a crystal structure is specific to the translation of atom positions of the asymmetric unit, within the unit cell, to positions with a symmetrical equivalence *(i.e the symmetry of every crystal structure with the same spacegroup can be determined through the same translation matrix operations to fill the unit cell with symmetry equivalent atom points)*. These translations are specified in three dimensions, in terms of the unit cell parameters. The unit cell parameters define the length of the three unit cell edges in the x, y, and z direction, and are notated a, b and c. The angles between the unit cell axis are notated α, β and γ.

Spacegroups in 3 dimensions are constructed through the combination of the 32 crystallographic point groups with 14 Bravais lattices, with each Bravais lattice belonging to one of 7 lattice systems. The resultant spacegroup is therefore a representation of the translational symmetry of the unit cell, combining lattice centring and the symmetry operations of reflection, rotation, rotoinversion, screw axis and glide planes. There are 230 known possible spacegroups derived from the combination of point groups and Bravais lattices.

The translational symmetry operations of the asymmetric unit to give atom positions of the unit-cell are often computed through black box operations within a computer program, however, this can be done manually by application of the symmetry operations as described within the "International Tables of Crystallography: Volume A."[61]

Within the "Tables for Crystallography", space groups are denoted by International short Hermann-Mauguin symbols, which represent space groups in two parts. The first part is a letter describing the centring of the space group *(e.g. P for primitive or F for face-centred),* and the second part is a set of characters representing the symmetry elements of the space group. The space groups are also represented in terms of space-group diagrams, which show the relative locations and orientations of symmetry elements, and the arrangement of symmetry equivalent points.

2.1.1 Transformations of the coordinate system

Symmetry operations are transformations by which the coordinate system and the origin are considered to be at rest, whilst the 'object' or molecule(s) is mapped onto itself. The coordinate system can be considered as the basis vectors *a, b and c*, and the origin 0. A symmetry operation *W* transforms every point *X* with the coordinates *x, y, z* to the point $\tilde{X}$ with coordinates $\tilde{x}, \tilde{y}, \tilde{z}$. In matrix notation, this transformation is equivalent to[62];

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\end{pmatrix} = \begin{pmatrix}w_{11} & w_{12} & w_{13}\\ w_{21} & w_{22} & w_{23}\\ w_{31} & w_{32} & w_{33}\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} + \begin{pmatrix}w_1\\ w_2\\ w_3\end{pmatrix}$$

$$= \begin{pmatrix}w_{11}x & + & w_{12}y & + & w_{13}z & + & w_1\\ w_{21}x & + & w_{22}y & + & w_{23}z & + & w_2\\ w_{31}x & + & w_{32}y & + & w_{33}z & + & w_3\end{pmatrix} \qquad [2.1]$$

The 3x3 matrix *(W)* represents the rotation part of the symmetry operator, and the column matrix *(w)* the translational part of the symmetry operation. W, w characterises the symmetry operation uniquely. This can be simplified by the use of an augmented 4x4 matrix;[63]

$$W = \begin{pmatrix}W & w\\ 0 & 1\end{pmatrix} = \begin{pmatrix}w_{11} & w_{12} & w_{13} & w_1\\ w_{21} & w_{22} & w_{23} & w_2\\ w_{31} & w_{32} & w_{33} & w_3\\ 0 & 0 & 0 & 1\end{pmatrix} \qquad [2.2]$$

This augmented matrix allows the calculation of the points $\tilde{x}, \tilde{y}, \tilde{z}$ by;

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\\ 1\end{pmatrix} = \begin{pmatrix}w_{11} & w_{12} & w_{13} & w_1\\ w_{21} & w_{22} & w_{23} & w_2\\ w_{31} & w_{32} & w_{33} & w_3\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ z\\ 1\end{pmatrix} = \begin{pmatrix}w_{11}x & + & w_{12}y & + & w_{13}z & + & w_1\\ w_{21}x & + & w_{22}y & + & w_{23}z & + & w_2\\ w_{31}x & + & w_{32}y & + & w_{33}z & + & w_3\\ & & & 1 & & & \end{pmatrix} = \tilde{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{x} \qquad [2.3]$$

Point group (any of the 32 symmetry operations which characterise 3D lattices) matrices (W) for operations are given in the "International Tables for Crystallography" (Tables 11.2.2.1 and 11.2.2.2).[64]

2.1.2 Calculating interatomic distances

The distance, *d*, between two Cartesian coordinates in 3D space can be deduced from Pythagoras' rule as;

$$d^2 = (X_p - X_q) + (Y_p - Y_q) + (Z_p - Z_q) \quad [2.4]$$

Using vector notation, the points *p* and *q* can be represented by vectors;

$$p = X_p i + Y_p j + Z_p k$$
$$q = X_q i + Y_q j + Z_q k \quad [2.5]$$

The distance between two atoms, *p* and *q* is equal to the magnitude of the vector;

$$d = p - q$$
$$= (X_p - X_q)i + (Y_p - Y_q)j + (Z_p - Z_q)k$$
$$= \Delta X i + \Delta Y j + \Delta Z k \quad [2.6]$$

The length of a vector is calculated from its scalar product with itself, so the distance, d, is defined by d^2=d·d. From this, the non-vector equation of d can be determined. The equations derived above show a relatively simple method for calculating the distance between two atoms in a Cartesian system. However, the high order symmetry of crystal structures, and the resultant symmetry operators, mean that a Cartesian system cannot be used for crystal structures, as it would not be possible to calculate symmetry-equivalent positions. Consequently, the standard method used to define atom positions within a unit cell is to use fractional coordinates based on the unit cell vectors a, b and c. Thus, the calculation for *d* becomes more complicated, and the treatment of fractional coordinates can be approached in two ways;

Conversion to Cartesian co-ordinates. The conversion of fractional coordinates to Cartesian coordinates can be performed using matrix multiplication. The disadvantage of this method is that the problem of calculating symmetry-equivalent positions still remains, and so the determination of inter-atomic distances would only be applicable for the asymmetric unit. The conversion of fractional coordinates to Cartesian coordinates is facilitated though a matrix operation, which is defined as follows;

$$\begin{bmatrix} a & b\cos\gamma & c\cos\beta \\ 0 & b\sin\gamma & c\left(\dfrac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}\right) \\ 0 & 0 & \dfrac{1}{c^*} \end{bmatrix} \quad [2.7]$$

The use of vector mathematics. Two points p and q with fractional coordinates in a unit cell are represented in vector notation as follows;

$$P = x_p a + y_p b + z_p c$$

$$Q = x_q a + y_q b + z_q c \qquad [2.8]$$

As with Cartesian systems, the distance *d* between *P* and *Q* is equal to the modulus of the vector joining them; so the overall calculation for the distance between two atoms with fractional coordinates would be derived as follows;

$$\begin{aligned} d^2 &= d \cdot d \\ &= (\Delta x a + \Delta y b + \Delta z c) \cdot (\Delta x a + \Delta y b + \Delta z c) \\ &= a^2(\Delta x)^2 + b^2(\Delta y)^2 + c^2(\Delta z)^2 + 2bc\cos\alpha(\Delta y)(\Delta z) + 2ac\cos\beta(\Delta x)(\Delta z) + 2ab\cos\gamma(\Delta x)(\Delta y) \end{aligned} \qquad [2.9]$$

When considering crystal structures, it is more convenient to deal with atomic coordinates and resultant calculations in terms of fractional coordinates, particularly when dealing with symmetry, as symmetry operators are traditionally expressed in fractional transformations in terms of the unit cell parameters.

The application of symmetry operators to obtain atom positions necessary for calculation of bond lengths and angles is both complicated, and computationally expensive. The calculation of bond angles can also be computationally expensive. Thus, treatment of the relevant equations by a computer program should be systematic and logical. Real-space metric tensor terms and Δ terms should be calculated previously for bond length calculations and should all be tabulated. Next, the value of the dot products should be calculated, followed by a final division of the result by each of the two bond lengths and derivation of the inverse cosine function to give the desired bond angle.

2.2 Calculating solvation free energy

In chapter one, a number of different solvation models were discussed in order to cover the current state of solubility prediction. Here, we focus on the specific methods used in this work, which are the RISM models briefly discussed at the end of chapter one.

2.2.1 Thermodynamics of solutions

The chemical potential of a component *i* in a mixture is given by;

$$\mu_i \equiv \mu_i^{ref} + RT\ln(\gamma_i x_i) \qquad [2.10]$$

where x_i is mole fraction, and γ_i is the activity coefficient, defined in terms of the chemical potential in the reference state μ_i^{ref}. Both terms can be combined in different ways, according to convenience. Two different approaches are typically used for the determination of μ_i; the symmetric approach, which is typically used for a binary system where both compounds are liquids, and the asymmetric approach, which is typically used for a binary system of two phases.

In the symmetric convention, each component γ_i approaches unity as its x_i approaches unity, thus the mixture approaches ideal behaviour in agreement with Raoult's law. The reference chemical potential in this convention μ_i^{*R} is the molar Gibbs free energy of *i* under the same conditions as the mixture, and the choice of γ_i reflects this;

$$\mu_i = \mu_i^{*R} + RT\ln(\gamma_i^R x_i) \qquad x_i \to 1 \Longrightarrow \gamma_i^R \to 1 \tag{2.11}$$

The activity coefficient accounts for non-ideal behaviour occurring due to interactions differing from those found in the pure substances of the mixture.

The asymmetric convention is preferred for solutions, as these typically contain substances that are not in the same state prior to solvation (under the same conditions). In this convention, γ_i for the solvent still approximately approaches unity as x_i approaches unity. However, for the solute, γ_i approaches unity in the limit of infinite dilution; the whole system approaches ideal behaviour in accordance with Henry's law;

$$mu_i = \mu_i^{*H} + RT\ln(\gamma_i^H x_i) \qquad x_i \to 0 \Longrightarrow \gamma_i^H \to 1 \tag{2.12}$$

where the superscript H refers to the system in a hypothetical reference state. This state is obtained from the extrapolation of the infinite-dilution limit, where no solute-solute interactions are present, but the solute mole fraction remains at unity. We will refer to the chemical potential derived from the asymmetric convention henceforth.

The Gibbs free energy of solution $\Delta_{sol}G_i$ is the difference between μ when transferring the solute from its pure state into an infinitely dilute solution, under constant temperature and pressure. The solute mole fraction retains unity;

$$\Delta_{sol}G_i \equiv \mu_i^{*H} - \mu_i^* \tag{2.13}$$

Relating $\Delta_{sol}G_i$ to an experimental solubility becomes more complicated when a solute's phase is not the same in its stable physical state and in solution. The free-energy difference between the pure solute and the solute phase in solution has to be computed. If the solute remains pure at equilibrium with the solution, and $\gamma_i^H \approx 1$, using eqn 2.12 and 2.13, the following approximation can be made;

$$\Delta_{sol}G_i \approx -RT\ln x_i^{sol} \tag{2.14}$$

where x_i^{sol} is the solute's measured solubility measured in mol/L. The Gibbs free energy of solution expresses the difference between solute-solute interactions in its stable physical state, and solute-solvent interactions in solution. In order to isolate the role of solute-solvent interactions, the free energy of solvation is used. This free-energy term gives the difference in chemical potential when the solute is transferred from an ideal gas at standard pressure into the reference state at infinite dilution;

$$\Delta_{solv}G_i \equiv \mu_i^{*H}\mu_i^{ig,0}$$

[2.15]

where $\mu_i^{ig,0}$ is the chemical potential of the solute as an ideal gas under standard conditions. If the chemical potential is expressed in terms of its fugacity f_i;

$$\mu_i = \mu_i^{ig,0} + RT\ln\left(\frac{f_i}{\rho^0}\right)$$

[2.16]

where ρ^0 refers to standard pressure. Equations 2.13 and 2.16 give;

$$\Delta_{solv}G_i = RT\ln\left(\frac{K_{H,i}}{p^0}\right)$$

[2.17]

where $K_{H,i} \equiv \lim_{x\to 0}(f_i/x_i)$ defines the Henry's law constant.

Equations 2.13 and 2.15 can be used to relate $\Delta_{sol}G_i$ and $\Delta_{solv}G_i$, as;

$$\Delta_{solv}G_i = \Delta_{sol}G_i + (\mu_i^{*R} - \mu_i^{ig,0})$$

[2.18]

The difference $(\mu_i^{*R} - \mu_i^{ig,0})$ in eqn. 2.18 is the residual chemical potential, and approaches zero for gaseous solutes at low pressure. These conditions reflect thermodynamic behaviour whereby the properties of solution and solvation are approximately equal.

Knowing the Gibbs free energy of solvation under a given temperature or pressure, other thermodynamic properties can be calculated;

$$\Delta_{solv}H_i = -T^2\frac{\partial}{\partial T}\left(\frac{\Delta_{solv}G_i}{T}\right)_p$$
$$\Delta_{solv}S_i = -\left(\frac{\Delta_{solv}G_i}{\partial T}\right)_p$$
$$\Delta_{solv}V_i = \left(\frac{\Delta_{solv}G_i}{\partial p}\right)_T$$

[2.19]

The enthalpic contribution to solvation gives the energy of solute-solvent interactions, and the entropic term gives insight into structural reorganisation in solution.

The Gibbs free energy of solution and solvation are related to experimentally measured values through solubility measurements, limiting activity coefficient and Henry's law constant. These transformations are often explored with molecular modelling based in statistical thermodynamics[65].

2.2.2 RISM

The concepts of the partition function, correlation functions (radial, total and direct), the MOZ, and closure relations have already been introduced in section 1.5.1. The total correlation function *h(r)* and the direct correlation function *c(r)*, expressed in eqn. 1.22 in terms of the MOZ equation,

are dependent on each other. Therefore, the unknown function *h(r)* must be found self-consistently, which is a characteristic of all many-body problems. This is achieved through a closure relation, as described exactly in eqn. 1.23. An approximate closure, HNC, is also presented in eqn. 1.25. The theory of RISM and closure relations are discussed in more detail below.

RISM, briefly mentioned in section 1.5.1, is a method designed to calculate the site-site correlation functions $g_{\alpha\beta}(r_{\alpha\beta})$. RISM typically treats molecules as rigid, comprising atoms represented by hard- spheres. RISM provides an approximate statistical mechanical theory. The theory consists of an Ornstein Zernike (OZ)-like relation between the total and direct correlation functions, and a closure. The OZ-like relation is a matrix, and in reciprocal **k**-space is;

$$h_{\alpha\beta}(k) = \sum_{\alpha'\beta'} \omega_{\alpha\alpha'}(k) c_{\alpha'\beta'}(k) \omega_{\beta'\beta}(k) + \rho \sum_{\alpha'\beta'} \omega_{\alpha\alpha'}(k) c_{\alpha'\beta'}(k) h_{\beta'\beta}(k) \quad [2.20]$$

where $\omega_{\alpha\alpha'}(k)$ is an intramolecular correlation function, which describes the structure of the molecules, and is given by;

$$\omega_{\alpha\alpha'}(k) = \frac{\sin k l_{\alpha\alpha'}}{k l_{\alpha\alpha'}} \quad [2.21]$$

where $l_{\alpha\alpha'}$ is the intramolecular distance between α and α'. In real space, this is written;

$$\omega_{\alpha\alpha'}(\boldsymbol{r}) = \frac{\delta(r - l_{\alpha\alpha'})}{4\pi l_{\alpha\alpha'}^2} \quad [2.22]$$

and is proportional to the probability density of finding α' at the position r from α. Equation 2.20 is therefore exact, and can be interpreted physically by iteration. A typical term for ρ^i, as represented diagrammatically in Fig. 7, is;

$$\rho^2 \omega_{\alpha\alpha'} c_{\alpha'\gamma'} c_{\gamma'\delta'} \omega_{\delta'\delta''} c_{\delta''\beta'} \quad [2.23]$$

Figure 7 represents a diatomic fluid. The correlation between the α and β sites of molecules 1 and 2 respectively occurs via an intramolecular interaction between α and α', followed by an intermolecular interaction α' to γ', and so on. Equation 2.20 is a sum over all such interaction chains for a system. This can only be solved exactly for a homonuclear (including diatomic) solution.

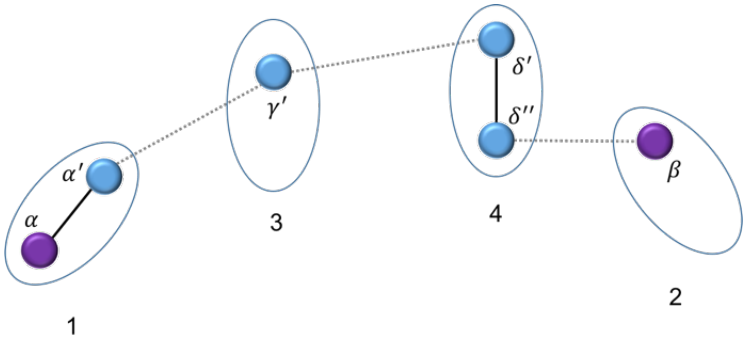


Figure 7 - Correlation of site α of molecule 1 with site β of molecule 2, via other sites of molecule 3 and 4. Solid lines represent intramolecular correlation functions (ω) and dashed lines represent intermolecular correlation functions (c). Adapted from[66]

If the chain-like model described above is used to describe the effects of bonding on the local intermolecular structure, then $c_{\alpha\gamma}(r)$ is effectively an intermolecular site-site potential. Assuming this as true, it is sensible to assume that $c_{\alpha\gamma}(r)$ has the same range as the actual potential between molecular sites. Using molecules with a hard-core, this corresponds to the distance of closest approach $d_{\alpha\gamma}$ for sites α and γ. For small distances of r, $c_{\alpha\gamma}(r)$ is not zero, but $g_{\alpha\gamma}(r)$ is. The exact expansion for $c_{\alpha\gamma}(r)$ is known, but is an infinite series expression, and some of its terms become difficult to integrate. Hence, a closure relation is required to solve eqn. 2.20.

The most commonly used closure relations are those which are most simply solved, or those with the best results for a chosen model. A good choice of the closure relation is given in terms of the RDF by[67];

$$g(r) = \exp(-\beta u(r) + t(r) + B(r)) \qquad [2.24]$$

where $t(r) = h(r) - c(r)$, the indirect correlation function, and $u(r)$ is the interaction pair potential. $B(r)$ is a bridge function, comprising all contributions to $g(r)$ not accounted for in the indirect correlation function or its products, and can be written as an exact functional of $h(r)$. It is not possible to calculate $g(r)$ exactly from eqn. 2.24, as this involves an infinite sum of integrals. Approximations to solve this problem usually involve setting the bridge functional to 0. Two closures that adopt this method, as are used most commonly, are the Percus-Yevick[68–70] (PY) and HNC[71–74] (as described in eqn. 1.25). The PY closure is given by;

$$h(r) = \exp\big(-\beta u(r)\big)\big(1 + t(r)\big) - 1 \qquad [2.25]$$

The PY closure has proved most successful when used with models represented by hard-spheres, or for systems without electrostatic interaction. The HNC closure is more mathematically rigorous, but is not necessarily always more accurate. The HNC closure is generally more successful for liquids with substantial attractive potentials than the PY closure.

2.2.3 Solvation free energy from RISM

For an infinitely dilute solution, the solvation free energy is the excess chemical potential $\Delta\mu$, as defined in section 2.3.1. Morita and Hiroike[75] have used the Kirkwood charging formula[76] to calculate this. The Kirkwood charging formula uses a coupling parameter, λ, which is varied

between 0 and 1, describing whether there is an interaction between solute and solvent (λ = 1) or not. For a solute site α, the formula states;

$$\Delta G_\alpha = \Delta\mu_\alpha = \sum_\gamma \rho_\gamma \int_0^1 d\lambda \int d\boldsymbol{r} \frac{\partial u_\gamma(\boldsymbol{r};\lambda)}{\partial\lambda} g_\gamma(\boldsymbol{r};\lambda)$$

[2.26]

As clear from the $d\lambda$ term, integration of this equation requires evaluation over several values of λ, which correspond to intermediate states between λ = 0 and λ = 1. However, we are primarily interested in the free energy at λ = 1. This evaluation is, however, dependent on closure relations, and may not actually exist for a given closure. If there is no direct differential for eqn. 2.71, then $\Delta\mu$ becomes path, rather than state, dependent.

The RISM-HNC expression for solvation free energy, as defined by Singer and Chandler[67] is;

$$\Delta\mu^{HNC} = -\frac{\rho}{2\beta}\sum \int 4\pi r^2 [2c_{\alpha\gamma}(r) + h_{\alpha\gamma}(r)c_{\alpha\gamma}(r) - h_{\alpha\gamma}^2(r)]\, d\boldsymbol{r}$$

[2.27]

2.3 Machine learning & cheminformatics

A brief introduction to machine learning, particularly applied to solvation methods is given in section 1.2. Here, a more in depth discussion of machine learning methods is given, in relation to the specific methods used within the work presented within the thesis.

2.3.1 Molecular representation

Molecular graphs are often used to describe chemical structure in a way understandable to computers. These graphs are based in graph theory, and consist of nodes, that are connected by edges. Applying this to a molecule, nodes and edges have properties associated with them. This is typically an atom type or atomic number for a node, and a bond order for an edge. The molecular graph only describes the topology of the molecule. This is demonstrated for paracetamol below.

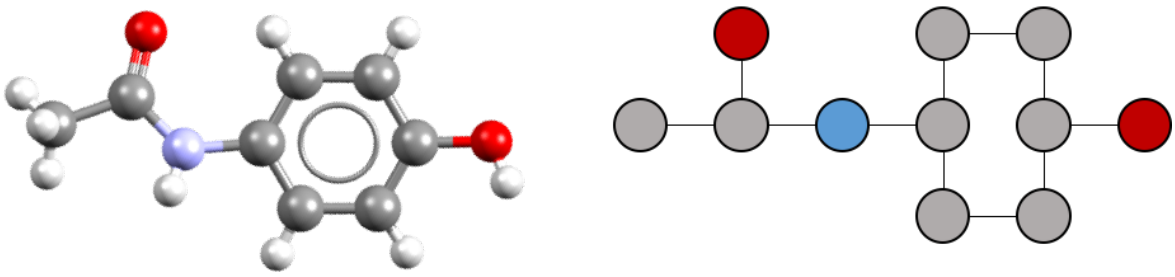


Figure 8 - A possible representation of paracetamol as a molecular graph (right), where oxygen atoms are represented by red nodes, and nitrogen as a blue node. The 3D structure (CSD refcode: HXACAN) is shown for comparison.

Subgraphs are subsets of nodes and edges in a graph. An example of a molecule containing a subgraph would be aspirin, where the ortho- substitution of groups on benzene make benzene a

subgraph of the overall molecular graph. Acyclic structures are represented by trees, which are graphs with no cycles or rings[23].

Two primary methods exist for communicating a molecular graph to a computer: connection tables and linear notations. Connection tables typically describe a molecule by its atomic coordinates (xy or xyz) and a list of the connections between atoms. An example of a connection table for the 3D structure of paracetamol is shown in Fig 9.

Linear representations of molecular structures use alphanumeric codes. Such notations are much compact, and are therefore useful for storing and transferring large numbers of structures. The Simplified Molecular Input Line Entry Specification (SMILES)[24,77–79] representation of molecules is a popular choice of linear representation, due to its simplicity and interpretability.

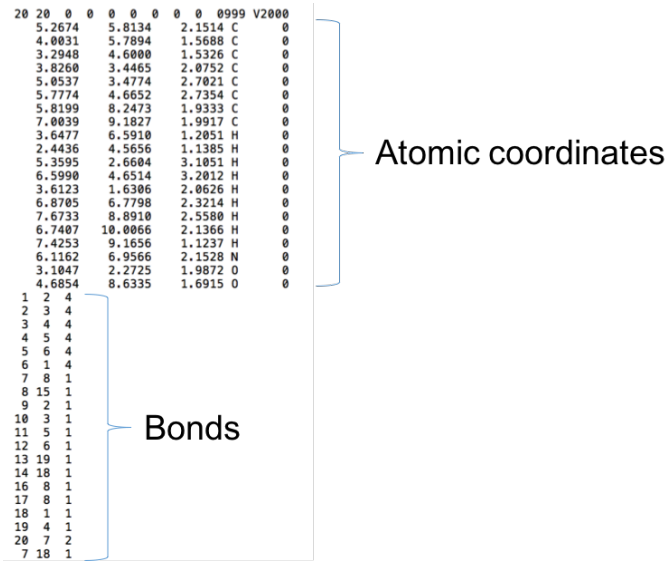


Figure 9 - The connection table for the 3D structure of paracetamol, as depicted in Fig. 8. The top line gives the number of atoms and the number of bonds. Below this, the atomic coordinates are given in Cartesian x,y,z representation. The final block gives the number of the atoms connected in the first and second columns, and the bond order in the third.

Chirality and isomerism can also be described by SMILES strings, where the absolute stereochemistry at chiral atoms is described with "@" or "@@", and geometric isomerism is described with "/' and "\" (two slashes in the same direction describe cis/Z conformation).

There are many ways to construct connection tables, and linear representations of the same molecules. For chemical databases and datasets, it is important to be able to determine repeat structures so that they are not given multiple entries. In order to achieve this, canonical representations are used. Canonical representations are constructed through a unique ordering of atoms for a given molecular graph, following a precise set of rules[79,80].

2.3.2 Molecular descriptors

Once a suitable representation of molecular structure has been implemented, molecular descriptors can be calculated and analysed. Molecular descriptors are numerical values associated to molecular properties.

The simplest of descriptors are counts of a particular feature. This could be a number of hydrogen bond donors or acceptors, for example. These descriptors are readily calculated from the molecular graph.

Another class of molecular descriptors are related to physiochemical properties, or their estimates[23]. A particularly important property is hydrophobicity, especially for the calculation of drug activity and transport, and as a significant contributor to solubility. It is most commonly modelled through the logarithm of the octanol-water partition coefficient, log P. The experimental determination of log P is particularly difficult, and although databases exist[81], its prediction for unknown compounds is clearly desirable.

The first method[82] to estimate log P was based upon an additive scheme;

$$\pi X = \log P_X - \log P_H \qquad [2.28]$$

where X represents a substituent, and H represents a parent compound. These values are evaluated from experiment, and the substituent constants πX are used to estimate the log P of unknown compounds. This method was not very successful, as substituent constants are not additive across species.

Fragment based schemes for estimating log P also exist, where log P is given by the sum of experimental log P values for the fragments and addition of a number of correction factors. This method[83,84] is represented mathematically by;

$$\log P = \sum_{i=1}^{n} a_i f_i + \sum_{j=1}^{m} b_j F_j \qquad [2.29]$$

where there are a_i fragments of type *i* with a contribution f_i, and b_j occurrences of the correction factor F_j.

This type of method is employed by the Clog P method, developed by Leo and Hansch[85]. The program contains a small number of experimentally determined log P values for simple molecules. The method breaks a molecule into fragment by isolating carbons with no double or triple bonds to heteroatoms, and treating them as hydrophobic fragments. This approach can be inaccurate for a large number of molecules.

Fragment methods have the advantage of accounting for significant electronic interactions, but only work well for structures where all fragments are characterised by an experimental log P value. Recent versions of the Clog P code include methods to estimate log P for missing fragments. An alternative approach to fragment based approaches is an atom based approach[86–89], which is given simply by;

$$\log P = \sum n_i a_i \qquad [2.30]$$

where n_i is the number of atoms of type *i* and a_i is its atomic contribution. The atomic contributions are determined from regression analysis, with a training set of compounds with experimentally determined log P. Atom methods do not have the problem of missing fragments, but a large number of atom types are needed to describe the molecule sufficiently. Long-range interactions are also neglected.

Another class of molecular descriptor are topological indices, which are calculated from a 2D molecular graph, and characterise structures by their size, degree of branching, and overall shape.

An example of a topological index is the Wiener index[90], which counts the number of bonds between atom pairs, and sums the distances between the pairs;

$$W = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij} \qquad [2.31]$$

where D_{ij} are distances.

Another index, the branching index, developed by Randić (1975)[91] calculates the number of adjacent non-hydrogen atoms for each atom, with its value known as the degree δ_i. The reciprocal of the square root of the product of δ_i for two atoms in the bond gives a bond connectivity value. The sum of the bond connectivity values for all non-hydrogen bonds in the molecule gives the branching index;

$$Branching\ index = \sum_{bonds} \frac{1}{\sqrt{\delta_i \delta_j}} \qquad [2.32]$$

This approach was extended by Kier and Hall[92] to produce the chi molecular connectivity indices. This index includes electronic information. The value of δ_i from eqn. 2.32 is defined by the number of sigma electrons of atom *i* minus the number of hydrogen atoms bonded to it. This is known as the simple delta. An additional valence delta is calculated by the same method, but using valence instead of sigma electrons. The chi molecular indices are sequential, and sum these delta values over different numbers of bonds.

A method of numerical description of molecular shape is given by kappa shape indices[93]. Molecular shape is compared with possible shapes produced by the same number of nodes. These shapes are of different order, with a first order shape being a count over single bonds. In this case, the shapes are a linear molecule, and a completely connected graph of each atom connected to every other atom. The resultant kappa shape index is calculated as;

$${}^1\kappa = \frac{2\ {}^1P_{max}\ {}^1P_{min}}{({}^1P)^2} \qquad [2.33]$$

where ${}^1P_{max}$ is the number of edges in the maximally connected graph, ${}^1P_{min}$ is the number of edges in the minimally connected graph, equivalent to a linear molecule, and ${}^1P$ is the number of bonds in the molecule for which the graph is being calculated.

2.3.3 Descriptor selection & linear regression

Linear regression methods find the response value of an input in terms of a linear combination of its predictors. In mathematical notation;

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p \quad [2.34]$$

where $\hat{y}(w, x)$ is the response value in terms of coefficients *w* of associated predictors *x*, and w_0 is the intercept. The aim of linear regression is to minimise the sum of differences between actual, and predicted values of y. This is known as a least-squares estimation.

l_1 norm and l_2 norm loss functions and the l_1 and l_2 regularisers. In machine learning, the l_1 norm and l_2 norm loss functions and the l_1 and l_2 regularisation are often referred to when discussing regression models. This terminology can be quite confusing, and so is discussed briefly here.

The l_1 norm and l_2 norm loss functions refer to error functions of the regression models selected. The l_1 norm loss function is also known as the least absolute errors (LAE), and minimises the mean absolute error (MAE). The l_2 norm loss function is also known as the least squares error (LSE), and minimises the mean squared error (MSE). As the l_2 norm loss function minimises square error, it is extremely sensitive to outliers, since outliers will have much larger squared errors, and the loss function will focus on reduction of this error, rather than the common example of error within the regression problem. This also means that the l_2 norm loss function has one solution – the one with the lowest MSE. The l_1 norm loss function is therefore more robust, and can also give multiple possible solutions in some cases. However, it is also less stable; movement of a single data point by a small distance can affect the regression line drastically, meaning a possible solution can be missed.

Regularisation in machine learning refers to the addition of a term to a method to prevent overfitting to the training data. The difference between the l_1 and l_2 regularisers are that the l_1 regulariser is a term for the sum of the weights in the model, whereas the l_2 regulariser is the sum of the squares of the weights.

Two factors can affect the overall quality of least-squares estimates from linear regression. The first is prediction accuracy. Least-squares estimates often have a low bias but large variance, and this can often be improved by either shrinking coefficients, or setting them to zero. The second is interpretation – with large numbers of predictors, it can be difficult to interpret the most important contributors to the model, and determining a smaller subset can allow us to look at the bigger picture by only considering the strongest effects of the predictors on the final prediction[94].

Two of the most commonly used methods for improving prediction accuracy and reducing the number of predictors in the final regression model are subset selection and ridge regression. However, both models have drawbacks[95]. Subset selection can be highly interpretable, but can also be extremely variable because it is a discrete method, meaning predictors are either retained or dropped from the model. Ridge regression attempts to resolve this by means of a continuous

process that shrinks coefficients rather than removing them. However, because it does not set any coefficients to zero, reduces the interpretability of descriptors in the final model.

Ridge. Given a regression with predictors x_{ij} and response values y_i for *i = 1, 2, …, N* and *j = 1, 2, …, p*, the ridge regression solves the l_2 regression problem of finding $\beta = \{\beta_j\}$ to minimize;

$$\sum_{i=1}^{N}\left(y_i - \sum_j x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

[2.35]

where $\lambda \geq 0$ is a tuning parameter, controlling the strength of a penalty term, which is related to β by;

$$\hat{\beta}^{ridge} = argmin_\beta ||y - X\beta||_2^2 + \lambda||\beta||_2^2$$

[2.36]

where the second term is the penalty term. When $\lambda = 0$, the usual linear regression estimate of the coefficients is found. When $\lambda = \infty$, $\hat{\beta}^{ridge} = 0$. The ridge regression works for the case between these two expressions, fitting a linear model of y on x, and shrinking the coefficients to find the optimal solution.

Lasso. The Least Absolute Shrinkage and Selection Operator (lasso) model aims to rectify the issues typical of subset selection and ridge regression by estimating sparse coefficients via reduction of the residual sum of squares (RSS) to the sum of the absolute value of the coefficients, being less than a constant. This constraint reduces some coefficients to zero, thus attempting to increase interpretability of the final regression model. Given a regression with predictors x_{ij} and response values y_i for *i = 1, 2, …, N* and *j = 1, 2, …, p*, the lasso solves the l_1 regression problem of finding $\beta = \{\beta_j\}$ to minimize;

$$\sum_{i=1}^{N}\left(y_i - \sum_j x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

[2.37]

where $\lambda\sum|\beta_j|$ is a penalty function assigned for each β_j coefficient. For some choice of the tuning parameter λ, this is equivalent to setting $\hat{\beta}_j = \hat{\beta}_j^0$ if $|\hat{\beta}_j^0| > \lambda$, and to zero otherwise, where $\hat{\beta}_j$ are usual least-squares estimates[95,96].

A method applied to the estimation of sparse coefficients in the lasso model is coordinate decent, which aims to find the local minimum of a function (in this case $\hat{\beta}_j$) by applying a linear search along one coordinate direction at the current point in each iteration, solving univariate optimization problems in a loop.[97] Applied to the lasso method it fixes λ in the Lagrangian form of the lasso problem;

$$\hat{\beta}^{lasso} = argmin_{\beta}\left\{\frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\} \quad [2.38]$$

denoting the current estimate for β_k at λ as $\tilde{\beta}_k(\lambda)$ and isolating β_j;

$$R(\tilde{\beta}(\lambda), \beta_j) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k(\lambda) - x_{ij}\beta_j\right)^2 + \lambda\sum_{k\neq j}|\tilde{\beta}_k(\lambda)| \quad [2.39]$$

which can be viewed as a univariate lasso problem where the response variable is the partial residual $y_i - \tilde{y}_i^{(j)} = y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k(\lambda)$. This has an explicit solution, and updates $\tilde{\beta}_k(\lambda)$ by;

$$\tilde{\beta}_j(\lambda) \leftarrow S\left(\sum_{i=1}^{N} x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right), \lambda\right) \quad [2.40]$$

where $S(t, \lambda) = sign(t)(|t| - \lambda)_+$ is a soft-thresholding operator. The first argument to $S(\cdot)$ is the simple least-squares coefficient of the partial residual on the standardized variable x_{ij}. Cycling through each variable independently, until convergence, yields the lasso estimate $\tilde{\beta}_k(\lambda)$.

Coordinate descent can be viewed as a version of forward stepwise regression, whereby the model is built in sequence, adding one variable at a time. For forward stepwise regression, at each step the best variable is identified and included in an active set. The least-squares fit is then updated to include all of the active variables. Least Angle Regression (LAR)[98] is similar, but also includes an estimate of how much of the predictor should be included in the final model. The first step identifies the variable with the best correlation to the response vector. The coefficient of this predictor is then moved continuously toward its least-squares value. When other variables become similarly correlated the process is paused and the second variable added to the second set. The variables in the active set are moved in a direction defined by their joint least-squares coefficient until all of the predictors have been entered. The LAR algorithm, unlike the coordinate decent algorithm, yields a full least-squares solution for the regression problem when implemented with the lasso method[94].

Elastic Net. The elastic net method[99] is a hybrid regression method, which includes penalty terms for both the l_1 and l_2 regularisation problems;

$$\hat{\beta}^{elastic\ net} = argmin_{\beta}\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1 \quad [2.41]$$

where the l_1 part of the penalty term creates a sparse model, and the quadratic part removes the limitation on the number of terms selected, encourages the grouping effect, and stabilises the l_1 regularisation path. This combination of penalty terms can be thought of as an effective mix between the lasso and ridge models. This is demonstrated below, in figure 10, which shows the geometry of the regularisation path for the ridge (black), elastic net (red) and lasso (blue) models. Singularities at the vertices correspond to sparsity, and the degree of convexity on the path varies

with the grouping effect. The grouping effect refers to the grouping together of highly correlated descriptors, which appear together either inside or outside of the model (for lasso, where coefficients can be zero).

In the lasso model, the grouping of descriptors is not revealed. It also exhibits limitations when there are more predictors than descriptors (p > n), and in this case, a maximum of n descriptors are included in the model, due to the convex optimisation problem. In addition to this, if a group of highly correlated descriptors exists, the lasso will select one from the group at random, and does not care which. In the case where n > p, it has been shown that lasso is outperformed by ridge[95]. The inclusion of the l_2 penalty in elastic net aims to improve the prediction performance issue of the lasso method by mimicking that of the ridge regression, and the inclusion of the l_1 term aims to mimic the variable selection of the lasso method[99].

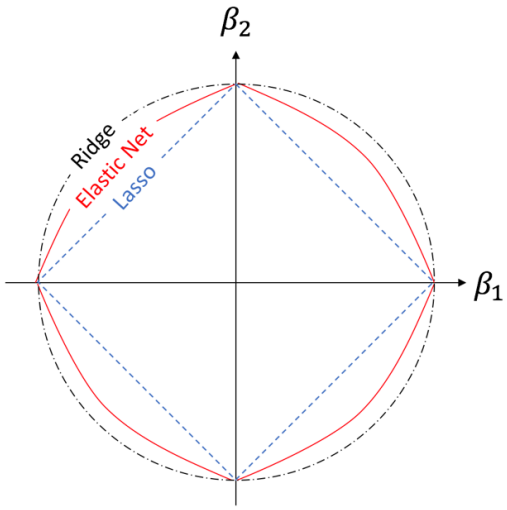


Figure 10 – comparison of the geometry of the regularisation paths for the lasso, elastic net and ridge regression problem models.

2.3.4 Statistical measures

Residual Sum of Squares (RSS). The sum of squares of deviations of predicted from actual empirical values of data (calculated for the test data).

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad [2.42]$$

where y_i are the actual values, and $\hat{y}_i$ are predicted values.

Explained variance. Measures the extent to which each model accounts for the dispersion of the given data.

$$explained\ varience(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}} \qquad [2.43]$$

where Var is variance (the expected value of the squared deviation from the mean) - $Var(x) = \sigma^2 = \int x^2 f(x)\, dx - \mu^2$ where $\mu = \int x\, f(x) dx$.

Mean Absolute Error (MAE). A risk metric corresponding to the expected error loss or l_1-norm loss.

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|$$

[2.44]

Mean Squared Error (MSE). Risk metric corresponding to the expected value of the quadratic error loss or l_2-norm loss.

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2$$

[2.45]

Spearman's rank correlation coefficient (r^2). A measure of how well future values are likely to be predicted from the model. Unlike most other scores, r^2 score may be negative (it need not actually be the square of a quantity r). The best possible score is 1.0 with smaller values being worse.

$$r^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{samples}-1}(y_i - \bar{y}_i)^2}$$

$$\bar{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} y_i$$

[2.46]

Coefficient of determination (R^2). R^2 is an interpretation of the proportion of the variance in the dependent variable that is predictable from the independent variable, ranging from 0 to 1; where 1 is a perfect prediction of the dependent variable from the independent variable, and 0 means that the dependent variable is not predictable from the independent variable.

$$R^2 = \left[\frac{1}{n_{samples}} \sum \frac{[(x_i - \bar{x})(y_i - \bar{y})]}{\sigma_x \sigma_y}\right]^2$$

[2.47]

where σ is the standard deviation.

AIC (Akaike Information Criteria). A measure that aims to select the best approximating model from a group of non-linear models[100]. Given a collection of models for the data, the AIC estimates the quality of each model, relative to all of the models being tested. It offers a relative estimate of the information lost when a model is used to mimic the process that generates the data. AIC is calculated by;

$$AIC = 2p - \ln(L)$$

[2.48]

where p is the number of parameters and ln(L) is the maximum log-likelihood of the estimated model;

$$\ln(L) = 0.5\left[-N\left[\ln(2\pi) + 1 - \ln(N) + \ln\sum_{i=1}^{n} {x_i}^2\right]\right] \quad [2.49]$$

where $x_1 \ldots x_n$ are the residuals from the nonlinear least-squares fit and n is the number of data points.

BIC (Bayesian Information Criteria). Has the same aim as the AIC, but gives the number of parameters in the model a higher penalty;

$$BIC = p(ln(n)) - 2\ln(L) \quad [2.50]$$

where n is the sample size.

Chapter 3

Machine Learning and Regression Models: Predicting log S

The programs developed and described in this chapter are available in Electronic Appendix I

3.1 Introduction

The general solubility equation (GSE)[29] is a Quantitative Structure-Property Relationship (QSPR) model used to predict the log S (log of the aqueous solubility) of a non-ionisable compound from its melting point (T_m) and its log P (log of the octanol-water partition coefficient) and is stated as;

$$\log S = 0.5 - 0.01(T_m - 25^o) - \log P \qquad [3.1]$$

with the melting point term set to zero (total term = 25°C) for solutes that are liquid at room temperature. The GSE has been found to be a good prediction model, performing very well[101] for a data set of 1026 organic compounds, with a coefficient of determination (r^2) = 0.96 and a root-mean-square error (RMSE) = 0.53. The equation assumes that the solubility of a solid (in water) is determined by its crystallinity and its interaction with water. The log P component of the GSE accounts for solute-solvent interactions, in terms of a difference between ideal and aqueous solution, and the T_m for the crystallinity of the compound. A simple view of solubility is represented by the GSE, and an example of properties relating to log P and T_m are shown in Fig. 11.

The logP term of the equation has more of an effect on the overall log S value predicted than the T_m term, in that the difference between lipophilicity of molecules is usually the largest contributor to varying solubility. No coefficient is applied to log P to scale it, whereas the T_m is multiplied by a coefficient of 0.01 and additionally has 25°C subtracted from it, meaning less than 1% of the solute's T_m is inherently accounted for in the GSE. This suggests that the measurement of solid-state properties and interactions is less useful than the solvation properties of the compound, as reflected in its log P contribution.

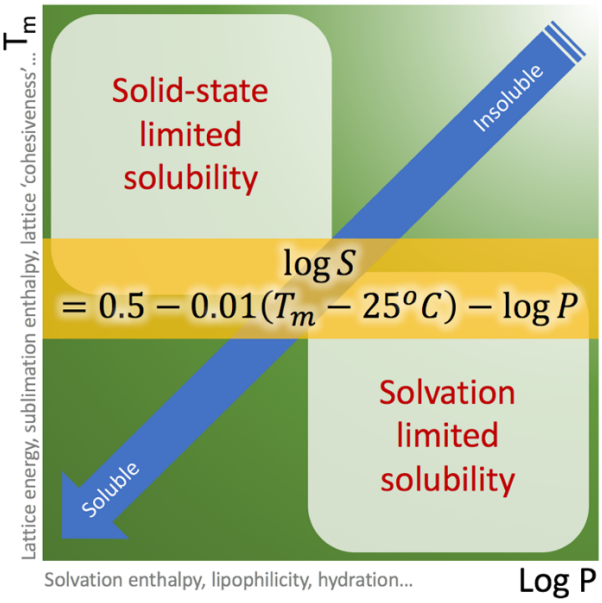


Figure 11 - a simple view of solubility. Improving solubility may be attainable by lowering the contributing properties relating to T_m and log P in the general solubility equation.

Wassvik *et al*[102] (2008) have investigated solid-state effects on the solubility of drug-like molecules. Their previous work found lipophilicity (represented by Clog P) for poorly soluble compounds to be in the range of 3.5-6.8, with a mean value of 5.3, indicating solubility limited by poor solvation (log P) – theoretically, in terms of the thermodynamic cycle discussed in chapter 1, this corresponds to hydration.

Using a set of hypothetical compounds with T_m in ranges considered low, intermediate and high, and lipophilicities considered within the same ranges, Wassvik *et al* were able to predict that only compounds with a ClogP (lipophilicity indicator) of ≤ 2, and a high melting point led to predictions where the solubility was predominantly driven by solid state effects. This is depicted in Fig. 12, where the blue segments represent solid-state effects (indicated by T_m), and the pie chart outlined in the purple circle represents this finding.

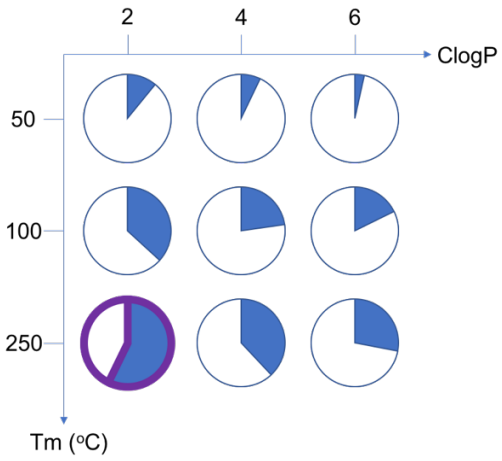


Figure 12 - The percentage of solubility determined by solid-state effects (blue) and lipophilicity (white) as determined by the GSE. Adapted from Wassvik *et al*[102].

From this finding, Wassvik *et al* aimed to evaluate the structural features of molecules that can lead to solid-state limited solubility. A set of 20 molecules, predicted to have solubility limited by solid-state effects, was investigated and showed a poor correlation between log S_0 and Clog P (R^2 = 0.04), confirming that factors other than lipophilicity were the driving force for solubility. Regression analysis of log S_0 with T_m, enthalpy of melting ΔH_m, and entropy of melting ΔS_m was conducted in order to investigate the driving-forces of solubility for the dataset compounds. Both T_m and ΔH_m correlated well with log S_0, but not ΔS_m. These correlations are indicative of solid-state driven solubility for the 20-molecule subset, and suggest that the rearrangement of molecules on melting into a liquid is not similar to the rearrangement of molecules on solvation. However, the energy required to break intermolecular interactions of the lattice during melting may be similar to the energy required during solvation.

In order to further probe these contributions, solubility was investigated with multivariate analysis using a number of 2D molecular descriptors, selected to capture structural features of the dataset molecules. These included descriptors for the rigidity of bonds, aromaticity, and the number of rigid fragments in the molecule. It was postulated that more rigid molecules would form more stable lattices, thus increasing the lattice energy, and decreasing solubility. In accordance with this, it was found that rigid molecules with high values for aromaticity descriptors were, as expected, consistently poorly soluble, suggesting that a thorough investigation of the structural features present within a molecule may be conducive to a good prediction of solubility, at least where solid-state features may be determined as the dominant contributor to solubility. Interestingly, the analysis found that the number of hydrogen bond donors and acceptors for the molecule did not correlate well with log S_0. This could indicate that the hydrogen bonding interactions of molecules in the crystalline phase are not easily deducible from 2D molecular descriptors, and that lattice contributions to solubility require more sophisticated description. However, it is important to note that the dataset investigated in this study is specifically selected, and is acknowledged by Wassvik *et al.* to not be unrepresentative of larger, drug-like datasets.

A study by Wolk *et al*[103] aiming to investigate the *in silico* prediction of physiochemical properties gives an indication of what percentage of compounds on the World Health Organisation's (WHO) Model List of Essential Medicines have compounds which are similar to those indicated within the purple circle in Fig. 12. They found 6.27 % ($\pm$ 4.39) of a dataset of 185 compound selected from the list (selected if for immediate release of the solid oral dosage form and with the highest dose strength) had a log P between 0.03 and 1.56, and a T_m between 164 ^{o}C and 289 ^{o}C.

Another study which has considered the use of specific structural information for the prediction of a property was conducted by Ouvrard and Mitchell;[104] investigating the prediction of enthalpies of sublimation from atom types defined by atomic number, hybridization state, and bonded environment. This approach effectively assumes that the energetic contributions of interactions are dependent on the functional group environments and atomic number (or size) of the atoms. Such atom-type descriptors are much easier to interpret and understand than the most widely used and conventional (such as electronic, topological, and graph-based) descriptors. Only heavy (non-H) atom descriptors were initially included, and models specific to certain classes (e.g. hydrocarbons), were initially investigated. The omission of information from hydrogen atoms aimed to ensure that no prior assumptions were made about packing effects such as hydrogen

bonded networks. No other information specific to crystal systems was included either (e.g. spacegroup). After investigating specific classes of compounds, it was found that for hydrogen-bonded systems, it was necessary to include some information about the hydrogen atoms contained in the structure, as enthalpy was partially dependent upon these contributions. The final model gave a regression equation, as the sum of a number of different atom type counts, and gave an $R^2 = 0.925$ for a training set of 226 compounds, and an $R^2 = 0.937$ for an independent (but similar to the training set) test set. This result shows that it is possible to predict the sublimation enthalpy of a compound with no prior knowledge of its crystal packing behaviour. This is in contrast with the suggestion by Wassvik *et al.* that these sorts of information should be included to improve predictions, if it is assumed that the solid-state contributions to solubility represented by T_m in the general solubility equation are similar or equivalent to the enthalpy of sublimation used in solubility prediction in terms of a thermodynamic cycle.

Another study investigating structural effects on solubility was conducted by Lovering *et al.*[105] (2009) who identify a lack of investigation into the relationship between the complexity of molecules and solubility. They identify this as an important characteristic, especially as drug-design and synthesis has tended toward representing complex molecules found in nature. The authors suggest a good correlation between bond saturation and changing solubility as an indicator toward synthetic strategies to improve solubility in the future, without affecting other important physiochemical properties. The authors investigate the applicability of bond saturation descriptors to solubility by regression analysis. Two descriptors are suggested as representative of the complexity of the investigated molecules. The first, Fsp^3, is a measure of carbon bond saturation, where Fsp^3 is the ratio of sp^3 carbons to the total number of carbons in the molecule. The second is a binary indicator representing the presence of a chiral carbon in the molecule. Lovering *et al.* aim to gauge whether there is an historical link between the saturation of a molecule with the stage of development to which it proceeded in the drug discovery pipeline. The GVK BIO database was used to identify the development stage of drug candidates. A trend was found between the complexity measurements employed and the development stage, with higher values of Fsp^3 (i.e. more saturated) found at the later stages of development; this corroborates the findings of Wassvik *et al*[102]. It was also found that more drugs with one or more stereo centres present made it through to the later stages of development. Increased Fsp^3 led to an increase in log S, suggesting more saturated molecules are likely to be more soluble. This finding is probably linked to the increased flexibility of a molecule with increased stereocentres leading to an increased solubility. If the molecule is more flexible, it is more easily able to reorganise itself in solution to form favourable interactions with the solvent.

This idea is also reflected in the results of a study by Salahinejad *et al.*[106] (2013), who have questioned the importance of crystal lattice interactions in the prediction of solubility for drug-like compounds. Salahinejad *et al.* initially aim to include descriptors for crystal-packing effects and intermolecular forces in a model for solubility prediction. In order to assess whether such descriptors could improve the performance of QSPR prediction models, calculated lattice energies and sublimation enthalpies were used as descriptors in a number of models. 86 descriptors including VolSurf (volume and surface based descriptors) and charged partial surface area (CPSA) descriptors were also included in the model. It was assumed that VolSurf descriptors would provide useful information about molecular interactions and that CPSA descriptors would explain polar intermolecular interactions. All of the lattice interaction descriptors, VolSurf, and CPSA

descriptors were used to generate a model of log S for 8421 small drug-like molecules. The most important descriptors were selected with multiple linear regression (MLR), MLR with expectation maximisation (MLREM), and a Bayesian regularised artificial neural network with a Laplacian prior (BRANNLP). The employed algorithms effectively remove non-important descriptors by setting them to 0, rather than creating new descriptor values based on linear combinations of descriptors which may be irrelevant on their own. The BRANNLP model provided the best overall performance for the entire dataset, with r^2 values of 0.83 and 0.82 for the training and test sets, respectively. In order to improve the model further, compounds that were largely under-represented in the dataset were removed, improving the RMSE of the model by between 0.15 and 0.25 log S units. Interestingly, an examination of the descriptors included in the model showed that the models performed similarly whether lattice interactions were included or removed, with only a slight preference for their inclusion. This falls in line with the findings of Mitchell and Ouvrard.

In this chapter, an evaluation of the GSE is conducted on a custom dataset. This dataset has not been manipulated to remove structures with poor solubilities, or to discriminate based upon solid-state effects or lipophilicity. This analysis aims to give a general idea as to the applicability of the GSE to a diverse set of drug-like compounds.

Following this, an attempt is made to determine the best possible regression model for logS from ordinary molecular descriptors. This is facilitated through a workflow developed in order to investigate logS prediction with a brute-force evaluation of a number of different estimators, with differing numbers of feature inputs, and a grid search of estimator hyper-parameters; using standard molecular descriptors. This is done through a set of routines, wrapped around existing machine-learning algorithms, which have been written into two new programs – BruteReg (Brute-force Regression) and BruteSis (analysis routines) – as described below. The programs developed can also be applied to any regression problem in the future.

3.2 Programs developed

3.2.1 BruteReg: A brute force workflow to find the 'best' regression methods

The problems and potential solutions associated with descriptor selection and regression model selection are discussed in section 2.4.3. Here, we develop a framework, employed in the form of the BruteReg program, for assessing multiple model estimators with different feature selection algorithms.

A workflow (as depicted in Fig. 14) is implemented through a programmatic process, developed in python, utilising a number of modules from the sci-kit learn[107] (sklearn) python package. Only brief descriptions of the sklearn modules used are included here, but more in-depth explanations can be found in the sklearn documentation and source code[107].

Workflow. The workflow employed by BruteReg for method grid searching is shown in Fig. 14. First, the user inputs a dataset of structures and response values to work with, represented for

each structure as a SMILES string, a label, and the true response value (i.e. the value to be predicted). Optionally, the program calculates descriptors for the input set (currently implemented with rdkit) – alternatively the user inputs a custom set of descriptors. This data is combined with the input data to create the full working dataset. The data is split into a user defined percentage split to generate a 'development' set and an 'evaluation' set, using a random number generator to select molecular structure indices (to avoid dataset bias). Each set comprises a data structure (or object in python) with its X values (descriptors – as a list of arrays with an array for each structure, and a complete matrix (data frame) including descriptor labels), Y values (response values) and the structural information (SMILES strings and labels).

The development set is then passed on to a further set of algorithms which perform feature selection, reducing the number of descriptors (features) to be evaluated in model development. Either a single value, k, can be provided as the k-best number of features to select, or a range of values and their separation can be provided (resulting in multiple sets of selected features). The output of this step is a set of arrays containing the column indices of the descriptor matrix (generated in the previous step), which can be accessed by further algorithms to build the feature-reduced descriptor sets for any structures with the same calculated descriptors. There are three algorithms implemented for feature selection:

- **F-regression** – This method calculates the cross correlation of each feature with the target (y) values, and calculates the corresponding f-values which are used as the selection values for the k-best features.
- **Mutual Information Regression** – This method looks for non-linear relationships between each feature and the target values. Mutual information (MI) is represented by a non-negative value, measuring the dependency between two variables, with a zero-value corresponding to complete independence, and higher values corresponding to higher dependency. The k-best MI are selected.
- **ExtraTrees Regressor** – This method uses a random forest of trees to calculate feature importance, and the k-best trees are selected.

The user selects which estimators should be evaluated, and defines a set of dictionaries containing any hyper-parameters that should be evaluated for each estimator. There are a number of estimators implemented into BruteReg as default:

- **Linear models:**
    - **Linear regression** – an ordinary least squares regression estimator
    - **Ridge regression** – solves the regression problem with a linear least squares function for loss, and regularisation with an L_2 prior.
    - **Ridge regression with cross validation (CV)** – ridge regression with built in CV
    - **Lasso** – performs variable selection and regularisation with an L_1 prior
    - **Lasso with CV** – lasso with built in CV
    - **LassoLars with CV** – lasso with least angle regression (instead of coordinate decent) and built in CV
    - **LassoLars with Information Criterion (IC)** – lasso lars with IC as the built-in validation method (either AIC or BIC)
    - **Elastic net** – a regularised regression method implementing the linear combination of the L_1 and L_2 penalties of the lasso and ridge methods
    - **Elastic net with CV** – elastic net with built in CV

- **Ensemble models:**
  - **Random forest regression**
  - **Extra random forest regression** – random forest with extremely randomised decision tree fitting (rather than deterministic) on sub-sets of the data, to improve predictive accuracy and control over-fitting
- **Linear support vector regression (SVR)**

For each set of selected features, the models above, and their hyper-parameters, a CV grid search is run and the results compiled. The grid search is performed by a module implementation from sklearn. The results from the grid search are output in a dictionary object, which are saved out from the program so that they can be re-loaded later, as this approach takes a long time. The methods tested in this step are referred to as the 'development methods'. The development methods are optionally saved out to a 'project' file, which contains important variable data and results (this is done by serialization with the python 'pickle' module).

Optionally, the development methods are then filtered to remove those that perform poorly, based upon user specified criteria of an average R^2 value (overall performance) used in CV, and an R^2 difference between the training and test sets (model generalisability). The methods that remain are now referred to as the 'evaluation methods'.

The evaluation methods are now retrained with the full development set, and then tested by prediction of Y for the evaluation set, giving a new set of method results, referred to as the 'analysis set'. These are the final methods for consideration as good models by the user, and as such, more comprehensive analysis metrics are calculated for each model evaluation; an explained variance score, MAE, MSE, the median absolute error, and R^2 for both the development and evaluation set. The final output of this step (and of the automated part of BruteReg) is a dictionary object containing the analysis set. This object is saved into the same file as the evaluation set, and can be loaded by external routines or scripts for further manipulation. An example of a method, as stored in a dictionary object, is shown in Fig. 13.

Default Parameters. The default parameters, defined in BruteReg include the values for which k-best features should be calculated, the estimators to use (as described above) and default parameter grids. These parameters are defined in the './modules/pipemodules.py' script in Electronic Appendix I.

```
In [59]: evaluation_results.iloc[0]

Out[59]: dev_evs                                                    0.739102
         dev_mae                                                    0.623838
         dev_median_ae                                              0.501034
         dev_mse                                                    0.639926
         dev_set_score                                              0.819417
         eval_evs                                                   0.694692
         eval_mae                                                   0.616625
         eval_median_ae                                             0.441952
         eval_mse                                                   0.720915
         eval_set_score                                             0.787241
         method_ids                                               [7, 0, 11]
         parameters          {u'normalize': True, u'l1_ratio': 0.8, u'n_alp...
         rank_test_score                                                    0
         Name: 0, dtype: object
```

Figure 13 - An example entry in the evaluation results (analysis set) dictionary, showing the keys (left) and corresponding data/values (right). Keys preceded by dev_ or eval_ are metrics, method_ids is an array of three numbers which are used to identify the specific methodology used to build the estimator, parameters is an additional dictionary string used to specify the optimised hyper-parameters of the estimator in the model, and rank_test_score gives the rank of the R^2 value of the test set prediction vs. true value, used in preliminary filtering routines.

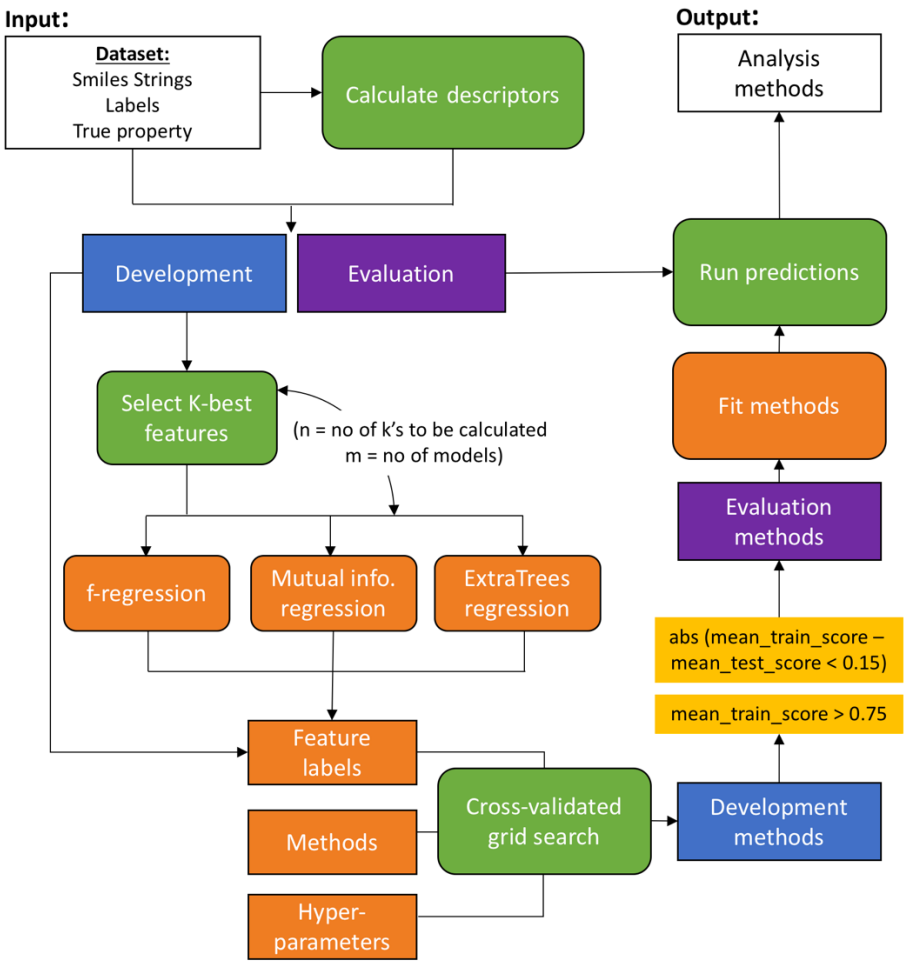
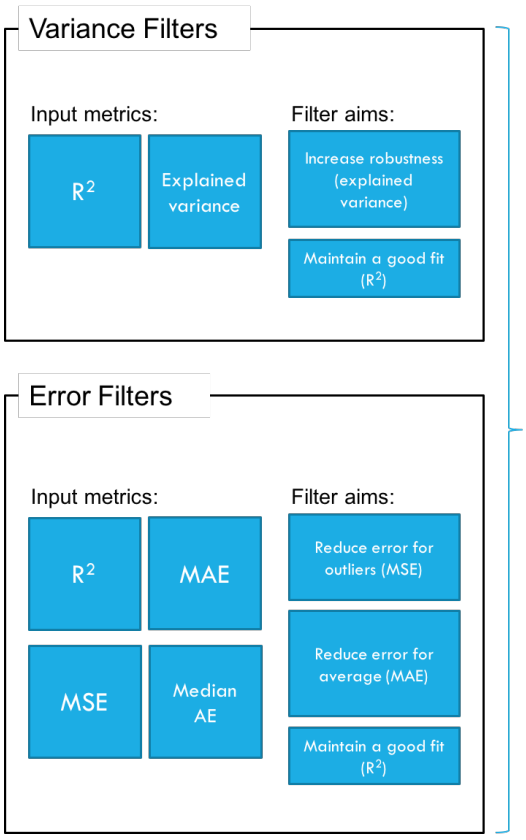


Figure 14 – The workflow implemented in BruteReg to develop regression models.

3.2.2 BruteSis: A GUI enabling filters, analysis, and visualisation

BruteSis is a GUI for the manipulation of the data output from the BruteReg algorithms. Upon the first instance of a 'project' – the opening of the results output from BruteReg - if the analysis set has not been automatically generated in BruteReg, the user is prompted to apply a standard filter to narrow down the number of models to initially be considered (at this point non-default parameters can be applied).

Filters. Other filters employed by BruteSis are not standard – that is, they should be used in accordance with the aim of the user. For example, the user may prioritise well explained variance of models over bias (absolute error) when selecting a model, or may want to create a composite scoring function which considers both of these problems. In this case, a scoring function can be built to rank both variance and bias of a model, and create a composite score weighted by a ratio (e.g. variance: bias), used for further ranking. These filters use the metrics calculated by BruteReg for the analysis set. The different filters in BruteSis are shown below in Fig. 15, along with an example of how they can be used to create a workflow set.




```
def scoring_function(Rsq_train, Rsq_test, exvartrain,
                     exvartest, MAEtrain, MAEtest,
                     MSEtrain, MSEtest):

    ## arrange indicies in order of R**2 of the test set
    indicies = np.argsort(Rsq_test)

    ## select best 100 R2 values
    for i in indicies[(len(Rsq_test)-100):]:
        indicies2.append(i)

    ## select R2 value differences (train-test) of less than 10%
    for i in indicies2:
        if abs(Rsq_train[i] - Rsq_test[i]) < 0.1:
            indicies3.append(i)

    ## select methods with explained variance in top 10%
    for i in indicies3:
        if exvartest[i] > max(exvartest)/100*90:
            indicies4.append(i)

    indicies4 = indicies4[::-1]
    varience_score = []
    error_score = []

    ## create a varience score from R**2 and explained varience
    for i in indicies4:
        varience_score.append((Rsq_train[i] + Rsq_test[i])
                              + (exvartrain[i] + exvartest[i]))

        ## create an error score from MAE and MSE
        error_score.append(MAEtest[i] + MAEtrain[i]
                           + MSEtest[i] + MSEtest[i])

    ## rank both scores, and combine to final score
    varience_rank = np.array(varience_score).argsort().argsort()[::-1]
    error_rank = np.array(error_score).argsort().argsort()

    final_score = error_rank + varience_rank
    final_rank = (np.array(final_score).argsort().argsort())+1
    test = sorted(zip(final_rank, indicies4))
    ranked = [x[0] for x in test]
    indicies5 = [x[1] for x in test]

    return ranked, indicies5
```

Figure 15 - The filters available (left) in BruteSis (as functions), and an example of a workflow that may be employed to create a composite scoring/ ranking filter (right)

Model re-construction. After filtering the results, the user can also choose to re-construct the fitted model, and export and import it as a model for future use. This means that the user can load a developed and selected model into BruteSis, and calculate the model property (target) for structures not in the development or evaluation set, either on a case-by-case basis, or with a set

of compounds, by SMILES strings. For this, the descriptor calculation method from BruteReg is used to calculate the relevant descriptors for the unseen structures from the SMILES string, and the imported model is used to predict the response values.

Visualisation tools. BruteSis also includes a set of functions that allow the user to visualise data. For example, the regression plots for predicted vs. true values can be produced (with or without training data). The user can also print out equations from the linear models, or plot feature importances from ensemble models. Plotting of metrics for a group of models to be analysed is also possible. An example of some of the plots which may be generated by BruteSis are shown below in Fig. 16.

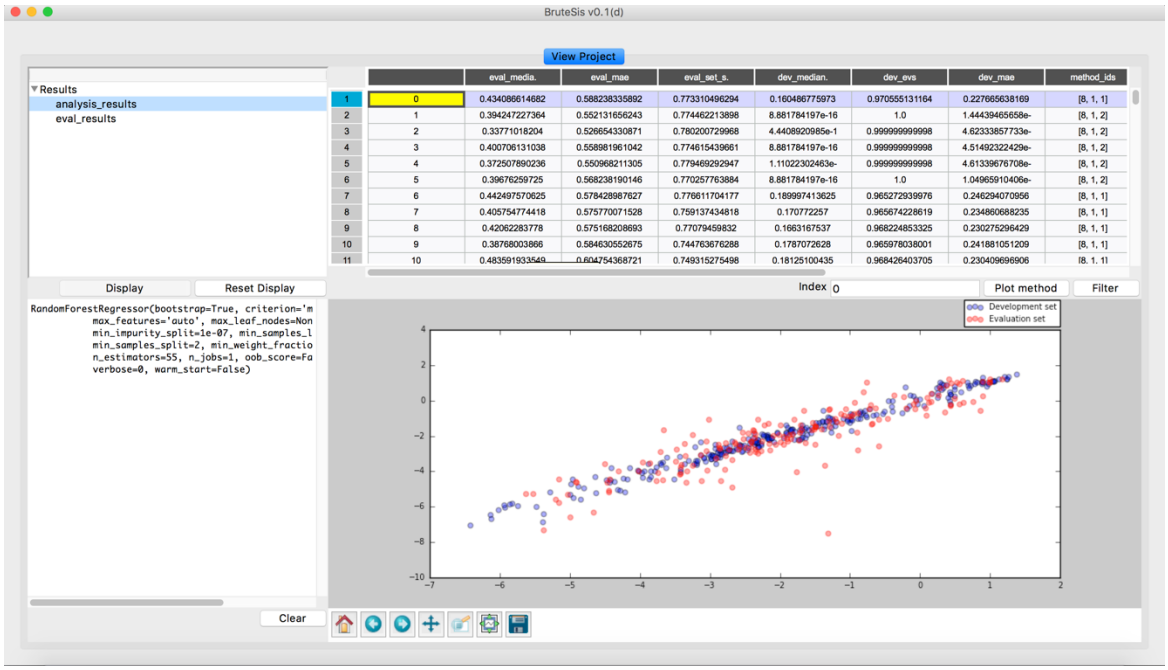


Figure 16 – An example of some of the visualisation outputs enabled by BruteSis – a table of the grid search method results, and a plot of the method described by row 0 of the table.

3.3 Methods

3.3.1 Dataset compilation

We have searched a recent version of the CSD (version 5.36 – 2015) for single component drug-like structures (no Lipinski violations) with available T_m data and aqueous solubility data. This has been facilitated by CCDC's new python API. The API allows searching and manipulation tasks, previously available in the CCDC GUI software suite, to be performed with python scripts. T_m data is available directly from the CSD. Solubility data has been taken from four sources; three published datasets – Hou[108], Huuskonen[109] and Delaney[110], and a database search via the EPI (Estimation Programs Interface) Suite[111] (version 4.11 2012). Searching these data sources is also performed through python scripts. A data-reading python package, 'pandas', is used to read in all of the datasets, and SMILES strings for the structures identified in the CSD search, which are then canonicalised with rdkit[112] (version 3.1, 2015). Once canonicalised, the solubility dataset SMILES are searched for SMILES strings matching those from the CSD search data.

A problem with automating this method is that sets of structures which are identical, isomeric (structural or conformational isomers), or polymorphic are identified by the same SMILES strings. Thus, any structures with identical SMILES strings have been identified and inspected manually. Where structures have been identified as identical, the 'best' structure (lowest R-factor and most sensible bond lengths/angles) has been selected, and other structures removed from the dataset. For isomeric systems, the literature has been searched to identify which isomer the dataset solubility value applies to, and for polymorphic systems, the literature has been searched for the most energetically favourable polymorph. Occasionally, solubility data has been found in the literature for metastable systems, and where this has been the case, these values have been added to the dataset. Where solubility data is not available for metastable systems, they have been removed from the dataset. The final dataset consists of 448 individual drug-like crystal structures and isomer/polymorph sets (total 452 structures, 8 polymorph/isomer sets) which have experimental T_m and solubility data available.

3.3.2 Assessment of the ability of the GSE to predict solubility

An initial investigation was conducted into the performance of the GSE on the dataset we have collated in order to establish its applicability. For this, three different algorithms were used to predict log P for each compound. Alog P was calculated using the CCDC API, and Mlog P – prediction of log P based on the number of carbon and hetero atoms - and Xlog P were calculated with the Chemistry Development Kit (CDK)[113]. Initially, the performance of the GSE was assessed by predicting log S for each compound according to equation 3.1, and calculating the residual sum of squares (RSS) and r^2 with respect to experimental log S.

Following the assessment of the GSE on our dataset, an ordinary least-squares linear regression (LR) was also conducted for each log P calculation algorithm using log P and T_m (-25^{o}C) as predictors, with training and test data selected randomly, in order to refit the intercept value and coefficients for T_m and log P.

3.3.3 Extrapolating meaning from molecular descriptors

In order to establish whether any additional structure or estimated property information could further enhance the prediction of solubility by means of a linear equation similar to the GSE, an investigation into the correlation of descriptors with log S, including regression of each descriptor to find an optimal relationship was conducted. Molecular descriptors were calculated using the rdkit python package[112], along with the molecular weight, and hydrogen bond donor and acceptor counts for each compound, which were calculated with the CCDC python API (total = 195). A list of the descriptors used is available in the rdkit documentation[112] (some descriptors involve multiple values).

In order to expose correlations between the descriptors used and experimental solubility (log S), a simple linear regression of each descriptor was performed. R^2 was also calculated for each descriptor with Log S from non-regressed data. Here, we discuss the most highly correlated descriptors.

3.3.4 Brute-force generation of regression models for logS

The dataset described in 3.3.1 was split into a development set (70%) and an evaluation set (30%). Following this, the evaluation set was manipulated with the k-best feature selection algorithm to generate the k-best sets of 10-90 features, increasing by 10 features (a total of 9 k-values) with all three algorithms implemented, as described in 3.1.1, generating a total of 27 different feature-label sets (3 algorithms x 9 k-values). These feature sets were then constructed on-the-fly, and fed into the grid search algorithm for all of the default estimators, with the default parameter grids, resulting in a total of 22830 models evaluated for the initial evaluation set. After filtering out poor models, and removing models which were extremely similar a total of 165 models were saved to the analysis set for further analysis.

3.4 Results & discussion

3.4.1 Assessment of the ability of the GSE to predict solubility

It was found that the GSE prediction for solubility was relatively poor for our dataset, regardless of which method of log P prediction was applied. The regression models applying Alog P and Xlog P predictions performed similarly, with r^2 vales = 0.60 and 0.66 respectively, and Mlog P had an r^2 = 0.14, suggesting it is a poor method for the calculation of log P.

The following equations were found for the three log P prediction methods when the GSE was refitted for the intercept value and coefficients for T_m and log P;

$$\log S = 0.082 - 0.007(T_m - 25^o) - 0.948(\text{Alog } P) \quad [3.2]$$

$$\log S = 1.926 - 0.004(T_m - 25^o) - 1.859(\text{Mlog } P) \quad [3.3]$$

$$\log S = 0.327 - 0.007(T_m - 25^o) - 0.976(\text{Xlog } P) \quad [3.4]$$

The results from these methods is depicted in Fig. 17. It can be seen from the above equations that the model for Xlog P predictions has the most similar intercept and coefficients to the original GSE, with a similar equation (differing by ~0.173 for the intercept, ~0.003 for the T_m coefficient and ~0.024 for the Xlog P coefficient). The RSS values for each log P prediction method (Alog P, Mlog P, Xlog P) with the original GSE equation were; 1.50, 2.71 and 1.14 and the r^2 values were; 0.60, 0.14 and 1.14 respectively. The RSS values for each log P prediction method (Alog P, Mlog P, Xlog P) with the modified LR equations were; 1.39, 2.21 and 1.03, and the r^2 values were; 0.63, 0.30 and 0.69 respectively, for the test data. The reduction of RSS for the new regression models indicates an overall improvement according to the aim of ordinary least-squares regression. The increase in r^2 represents an improvement in the explanation of variance.

Previous work by Ali *et al.*[31] (2012) has found that the GSE is able to predict log S within 1 log unit for a dataset of 1265 compounds. However, when applied to a subset of the data without values found in sparsely populated regions of the dataset, this reduces to 75%. This represents how sparsely populated regions of data-sets can significantly skew results. This may also be occurring

for our dataset, where there are large areas of solubility containing small numbers of structures at the extremities of the data, with our dataset having a large range of solubilities.

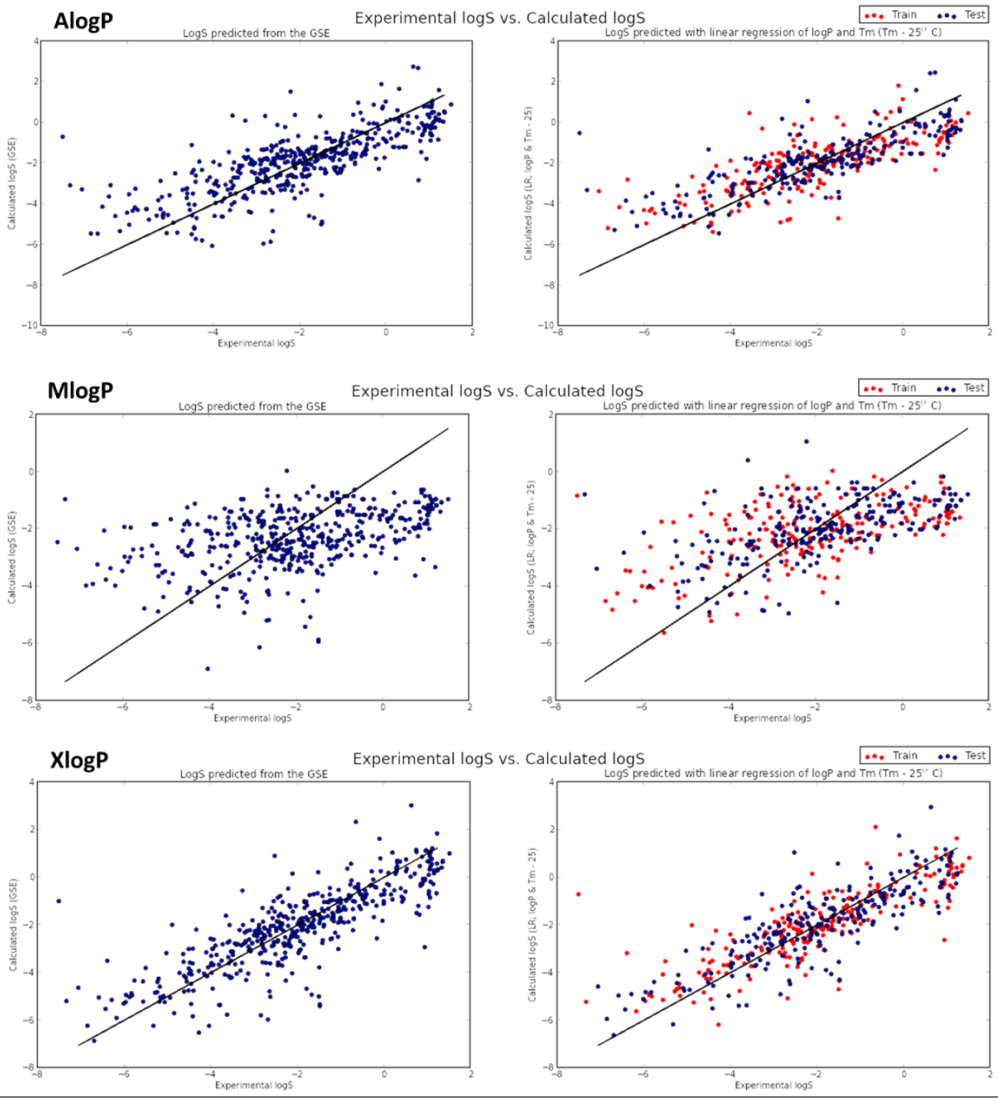


Figure 17 - Experimental vs. Calculated log S using three different prediction methods of log P with the GSE (left) and with ordinary least-squares linear regression models used to retrain the intercepts and coefficients of the GSE for our data (right – training data in red, test data in blue)

3.4.2 Extrapolating meaning from molecular descriptors

The best correlated descriptor (not including Xlog P) to log S is molMR (molar molecular refractivity – a measure of the total polarizability of one mole of a molecule calculated from summation of atomic contributions) with an R^2 = -0.659 and r^2 = 0.432. This descriptor infers information about molecular size and polarizability. This is unsurprising given that the aqueous solubility of a compound is known to be linked to its polarizability. However, because the descriptor also contains information about molecular size, it is important to consider its relation to other descriptors describing molecular size. Indeed, its correlation with molecular weight R^2 = 0.923. Thus, around 6% of the molMR descriptor value can be extrapolated as an indication of polarizability with no bias toward molecular weight, potentially allowing the comparison of relative polarizability of molecules of different sizes.

The next best correlated descriptor is chi0v (Kier and Hall Chi atomic valence connectivity index). These indices are represented by;

$$^{m}\chi_{q} = \sum_{k=1}^{k}\left(\prod_{a=1}^{n}\delta_{a}\right)_{k}^{-1/2} \quad [3.5]$$

where *m* is the order (number of verteces), *q* is a letter representing the connectivity index type (path, chain etc.), *k* is the total number of the m^{th} order subgraphs. The $\prod_{a=1}^{n}\delta_{a}$ term refers to the sum of the simple vertex degrees, which is calculated for each *k* subgraph. The vertex degree refers to the number of edges incident to the vertex (from graph theory), in this context, the number of bonds (edges) to an atom (vertex). For the case of indices relating to valence electrons, the δ_{a} term – the valence connectivity for the k^{th} atom – is calculated by;

$$\delta_{a} = \frac{(Z_{k}^{v} - H_{k})}{(Z_{k} - Z_{k}^{v} - 1)} \quad [3.6]$$

where Z refers to the number of atoms in the k^{th} atom, H the number of hydrogen atoms, and superscript v denotes valence electrons. Thus, the overall descriptor value is simply a sum of all of the numbers of atoms connected to each atom in the molecules, converted to the reciprocal square root; and indicates connectivity of atoms within the molecular graph. More saturated, large molecules will therefore have larger chi0v values. This may suggest that increased atomic saturation contributes to increased solubility. This finding falls in line with the work of Wassvik *et al.*[102] and Lovering *et al.*[105] described earlier.

LabuteASA (Labute's approximate surface area) is an approximation of the van der Waals surface area. The correlation to log S, R^2 = -0.62 and r^2 = 0.383, making it the third most correlated descriptor. This descriptor is another representation of molecular size, corroborating that this sort of descriptor is a good indicator of solubility, as represented by the other well correlated descriptors.

The most highly ranked descriptors in the set (in terms of correlation to log S) almost always contain an inference of molecular size or complexity. However, this presents a problem for the development of an appropriate regression model, as it is important to select descriptors that are orthogonal to each other in order to avoid overfitting to one determinative property or feature. The top five correlated descriptors to log S are shown in table 1, with their inter-correlation values represented by R^2. It is clear that these descriptors are highly correlated, and would therefore be an inappropriate choice as combined descriptors in a regression model.

Table 2- Inter-correlations expressed as R^2 between the best five correlated descriptors to log S

	chi0v	**LabuteASA**	**MW**	**chi1v**
molMR	0.978	0.983	0.923	0.919
chi0v		0.978	0.945	0.956
LabuteASA			0.958	0.901
MW				0.877

3.4.3 The analysis and evaluation of models calculated with BruteReg

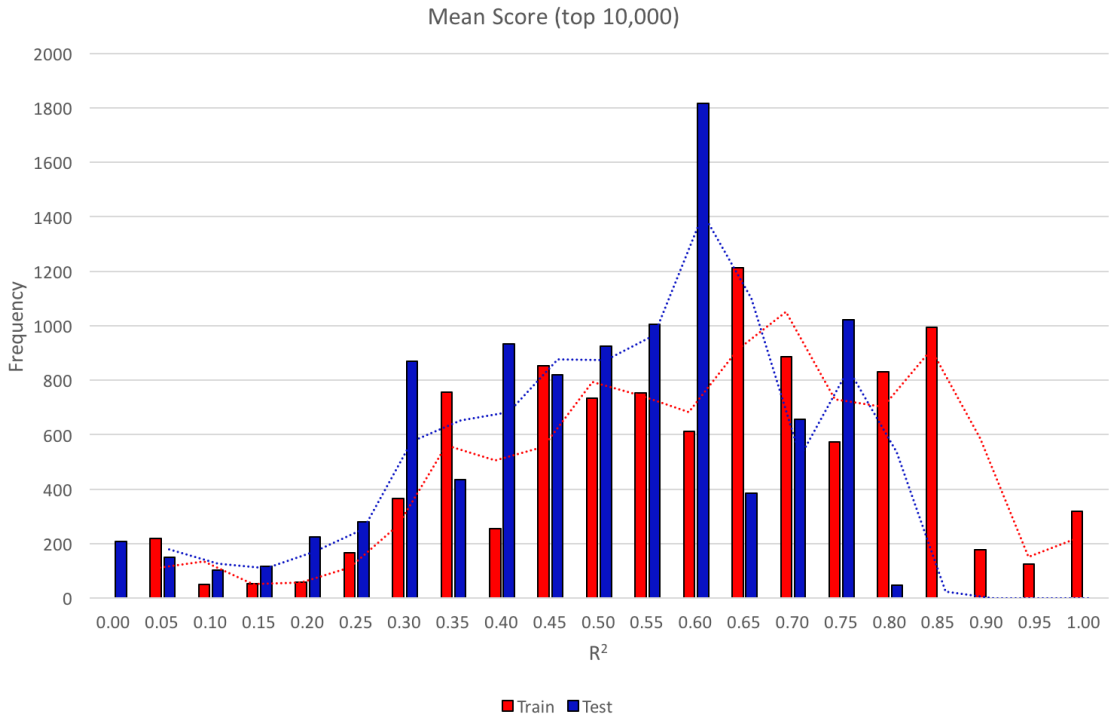


Figure 18 – A histogram showing the frequency of models with different R^2 scores, for the training sets (red) in the grid search CV, and the test sets (blue) produced in the grid search CV splits.

The initial run of BruteReg, with the default options described above, produced 22,829 models. The models were initially ranked by their mean R^2 score for the test sets used in the CV splits of the grid search. At this point, only the development set is seen by any of the models.

In Fig. 18, a histogram shows the distribution of the R^2 scores for the top 10,000 models. For the training sets in the grid search, the distribution of R^2 scores is broad. This corresponds to the broadness of the models produced by the grid search method, and their quality. It is noteworthy that there are models produced that have R^2 scores of 1.0 (frequency: 319). These scores correspond to models which are extremely over-fitted.

Following the generation of the evaluation set, BruteSis was used to further investigate a subset of the models created by BruteReg. In order to reduce the size of the model samples to evaluate, the filters described for generation of the analysis set were used, with the criterion that the R^2 difference between the test and training sets must be below 0.15, and the R^2 value of the training set must be above 0.75. The justification for these filter values is discussed below, and demonstrated in Fig. 19.

A plot of mean test scores vs. mean train scores is shown in Fig. 19 for the grid search CV method, which highlights a few interesting features. The most generalisable models, by definition, are those which have very small differences in scores between the training and test sets. For our data, these models can be seen as those closest to the black central line in Fig. 19. There are a large number of models which have this feature, but as the R^2 score for the test set gets larger, less

generalisable models are found, as indicated by the sparsity of points along the central black line at higher values along the x-axis. Two additional diagonal lines have also been plotted. These correspond to the filter criteria where the R^2 difference between the test and training sets must be below 0.15. This filter is imposed in order to ensure models that are further investigated in the analysis set have good generalisability. As can be seen, a good number of models are selected by this criterion. However, a number of these also have poor scores, thus an additional filter is imposed in the form of a minimum R^2 value of the training set. The training set is used rather than the test set, as there may be a limit on the possible R^2 value for the test set, which is unknown to the user at the initial model development stage, but there are usually models with up to nearly perfect, or perfect scores in the training set. The additional filter imposed is that the R^2 value of the training set must be above 0.75, plotted in Fig. 19 as a black horizontal line. The green area on the graph indicates the region in which models from the grid search are selected for the analysis set.

From the analysis set, the best models, according to R^2 of the test set (now the evaluation set rather than an average of the CV splits of the development set) were investigated. The best model found is shown in Fig. 20. This model is an elastic net model, with cross validation used to select the descriptors in the final model. The hyperparameters found by BruteReg included normalisation of the descriptors, fitting the intercept of the regression equation, using no alphas (along the regularisation path), and a 0.8 L_1 ratio (the ratio of the regulariser solving the L_1 rather than L_2 prior). The R^2 of the development set was 0.819, and of the evaluation set was 0.787; the MSE of the development set was 0.721, and of the evaluation set was 0.695. Although this model appears to be quite good, the final regression equation given contained in excess of 20 descriptors. This makes the model difficult to interpret. In fact, a vast number of models in the top results of the analysis set contained a large number of descriptors.

The minimum number of descriptors (or features) found in any model was 5, and the maximum was 81. Usually, non-general regression methods, such as random forest, will contain more features. However, such methods tend to overfit the data in a majority of cases, and most of the best methods were general linear regression methods. The overfitting in the case of random forest models may be attributable to insufficient specification of hyperparameter grids in the initial methodology.

As such, an additional filter was applied to the analysis set in order to select those models which were good performers, but were more interpretable. The minimum number of descriptors was set to 5, and the maximum to 15. This left a set of 37 models for further analysis, which are summarised in table 3. Interestingly, all of the models in this set were produced by either elastic net with CV, lasso lars with IC, or lasso with CV. This could be because all of these models had better defined default options for the hyperparameter grids used in BruteReg. However, as all of these models aim to reduce the number of features selected, and use CV or IC to select those features effectively, it is likely that it is the lasso and elastic net methods themselves that produce the best models for the log S regression, in addition to the constraint of fewer descriptors omitting alternative estimators.

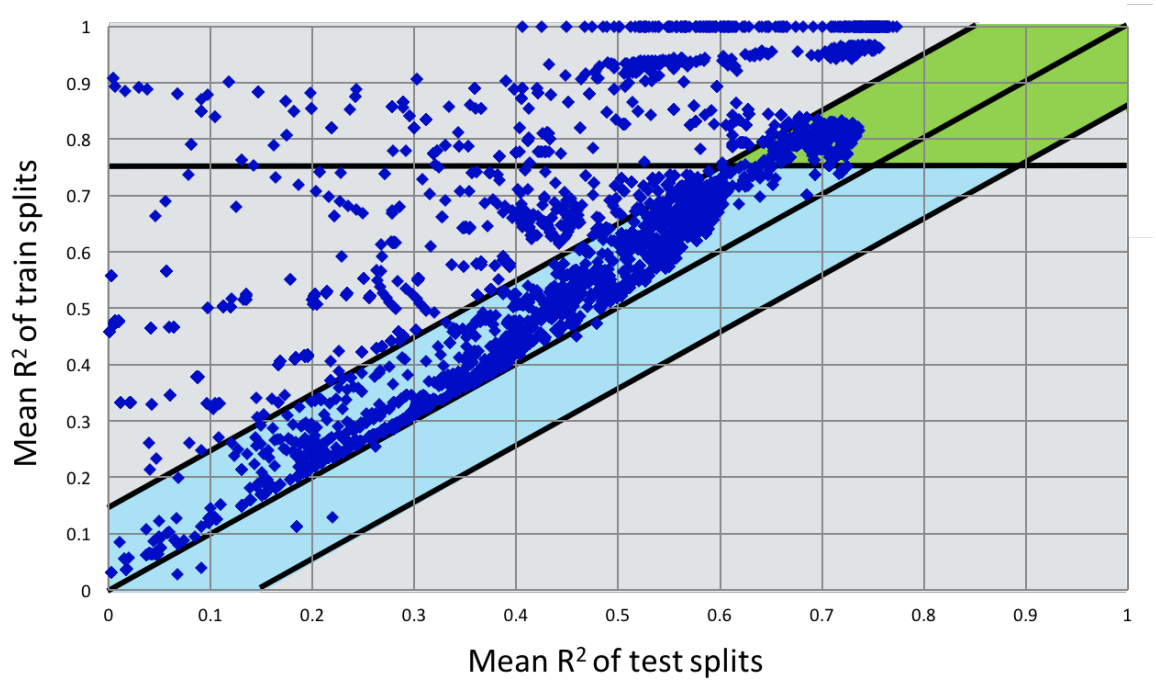


Figure 19 – A scatter plot of the mean R^2 scores of the test set from the grid search CV splits (x-axis) vs the mean training score from the grid search CV splits (y-axis). The three black diagonal lines indicate the filter criterion for generalisability of the model produced, with the central line indicating no difference in performance of the model between the training and test sets, and each additional line representing a difference in performance represented by +/- 0.15 R^2. The horizontal line represents the criterion that the mean R^2 of the training set must be > 0.75.

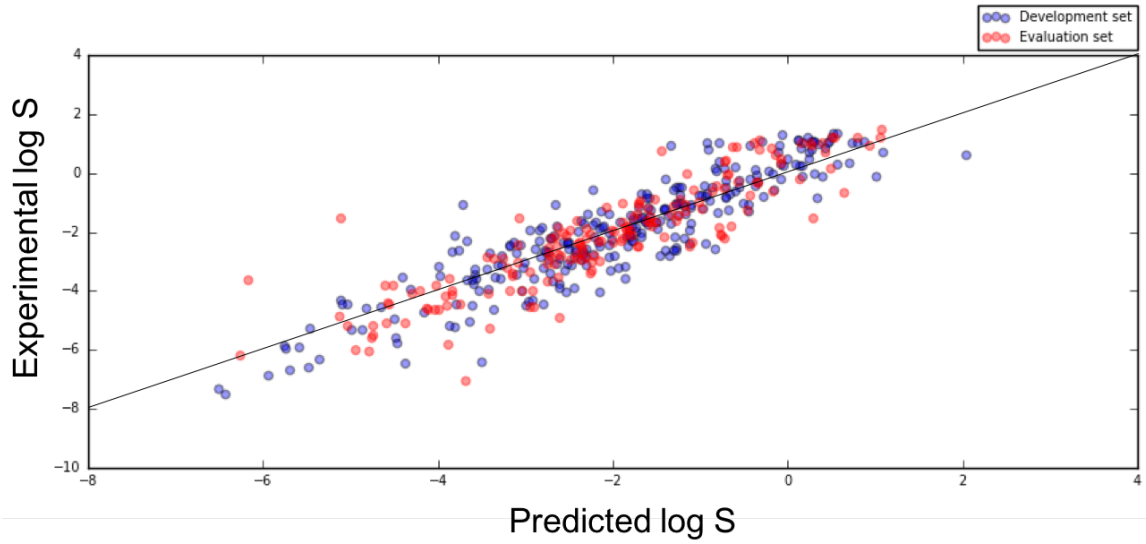


Figure 20 - The best regression method from the initial analysis set, plotted with the estimator refit to the full development set (blue), and tested on the evaluation set (red)

Table 3 - A table of results after filtering the analysis set to give models only containing between 5 and 10 descriptors, with their original ranks from the initial analysis set (by R^2). The three numbers in method id [i,j,k] are used to describe: i – the number of descriptors selected by initial feature reduction in BruteReg, where 0 corresponds to 10, with each additional integer equal to adding 10 (e.g. 1 = 20 descriptors, 3 = 40 descriptors); j – the feature reduction algorithm where 0 = f-regression, 1 = mutual information regression, and 2 = ExtraTrees; k – the estimator where 7=lassoCV, 9=LassoLarsIC, 11=ElasticNetCV (defined by sklearn).

	Development set					Evaluation set						
Original rank	Explained variance	MAE	Median AE	MSE	R^2	Explained variance	MAE	Median AE	MSE	R^2	**Method id**	**Parameters**
21	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 160, 'fit_intercept': True}
23	0.71	0.65	0.45	0.71	0.80	0.71	0.64	0.46	0.76	0.78	[4, 2, 9]	{'normalize': True, 'positive': False, 'criterion': 'bic', 'fit_intercept': True}
25	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 7]	{'normalize': True, 'n_alphas': 260, 'fit_intercept': True}
27	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 7]	{'normalize': True, 'n_alphas': 310, 'fit_intercept': True}
29	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 360, 'fit_intercept': True}
31	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 410, 'fit_intercept': True}
33	0.71	0.65	0.46	0.72	0.80	0.71	0.64	0.47	0.76	0.78	[3, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 510, 'fit_intercept': True}
35	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.76	0.78	[3, 2, 7]	{'normalize': True, 'n_alphas': 60, 'fit_intercept': True}
39	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.46	0.76	0.77	[4, 2, 7]	{'normalize': True, 'n_alphas': 10, 'fit_intercept': True}
41	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.46	0.76	0.77	[4, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 160, 'fit_intercept': True}
42	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.76	0.77	[5, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 260, 'fit_intercept': True}
43	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.76	0.77	[4, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 410, 'fit_intercept': True}
44	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.76	0.77	[5, 1, 7]	{'normalize': True, 'n_alphas': 360, 'fit_intercept': True}
46	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.76	0.77	[5, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 410, 'fit_intercept': True}
48	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.76	0.77	[4, 2, 7]	{'normalize': True, 'n_alphas': 360, 'fit_intercept': True}
49	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.76	0.77	[5, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 460, 'fit_intercept': True}
50	0.71	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.76	0.77	[4, 2, 7]	{'normalize': True, 'n_alphas': 310, 'fit_intercept': True}
51	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.76	0.77	[5, 1, 7]	{'normalize': True, 'n_alphas': 60, 'fit_intercept': True}
53	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.77	0.77	[5, 1, 7]	{'normalize': True, 'n_alphas': 110, 'fit_intercept': True}
54	0.70	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.77	0.77	[4, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 460, 'fit_intercept': True}
55	0.64	0.67	0.49	0.85	0.76	0.70	0.64	0.48	0.77	0.77	[5, 1, 7]	{'normalize': True, 'n_alphas': 210, 'fit_intercept': True}
57	0.70	0.65	0.46	0.72	0.80	0.70	0.64	0.47	0.77	0.77	[4, 2, 7]	{'normalize': True, 'n_alphas': 210, 'fit_intercept': True}
60	0.69	0.67	0.50	0.74	0.79	0.69	0.65	0.48	0.77	0.77	[7, 2, 9]	{'normalize': True, 'positive': False, 'criterion': 'bic', 'fit_intercept': True}
62	0.69	0.66	0.48	0.74	0.79	0.69	0.65	0.48	0.77	0.77	[5, 2, 7]	{'normalize': True, 'n_alphas': 110, 'fit_intercept': True}
64	0.69	0.66	0.48	0.74	0.79	0.69	0.65	0.48	0.77	0.77	[5, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 210, 'fit_intercept': True}
66	0.69	0.66	0.48	0.74	0.79	0.69	0.65	0.48	0.77	0.77	[5, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 410, 'fit_intercept': True}
68	0.69	0.66	0.48	0.74	0.79	0.69	0.65	0.48	0.77	0.77	[5, 2, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 60, 'fit_intercept': True}
90	0.64	0.68	0.56	0.83	0.76	0.68	0.67	0.51	0.79	0.77	[6, 1, 7]	{'normalize': True, 'n_alphas': 160, 'fit_intercept': True}
92	0.63	0.68	0.56	0.84	0.76	0.68	0.67	0.51	0.79	0.77	[6, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 310, 'fit_intercept': True}
94	0.63	0.68	0.56	0.84	0.76	0.68	0.67	0.51	0.79	0.77	[6, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 410, 'fit_intercept': True}
96	0.63	0.68	0.56	0.84	0.76	0.68	0.67	0.51	0.79	0.77	[6, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 260, 'fit_intercept': True}
98	0.63	0.68	0.56	0.84	0.76	0.68	0.67	0.52	0.79	0.77	[6, 1, 11]	{'normalize': True, 'l1_ratio': 1.0, 'n_alphas': 110, 'fit_intercept': True}
101	0.66	0.68	0.53	0.76	0.78	0.66	0.67	0.53	0.80	0.77	[6, 0, 9]	{'normalize': True, 'positive': False, 'criterion': 'bic', 'fit_intercept': True}
129	0.54	0.72	0.60	0.93	0.74	0.62	0.69	0.55	0.83	0.76	[8, 1, 7]	{'normalize': True, 'n_alphas': 10, 'fit_intercept': True}

The best model from the new set (after descriptor filtering) was also produced by an elastic net estimator with CV for feature selection, and is shown below in Fig. 21. The hyperparameters found by BruteReg included normalisation of the descriptors, fitting the intercept of the regression equation, using no alphas (along the regularisation path), and a 1.0 L_1 ratio (the ratio of the regulariser solving the L_1 rather than L_2 prior). The R^2 of the development set was 0.797, and of the evaluation set was 0.776; the MSE of the development set was 0.718, and of the evaluation set was 0.760. These statistical measures do not differ much from the initial analysis sets' best result. However, the new best model contains only 10 descriptors, rather than an excessive number. The regression equation produced was:

$$\begin{aligned} logS = \ & 0.945 - 0.00115T_m - 0.180fr_{benzene} - 2.815fr_{nitroso} - 0.00329EState_{VSA3} \\ & - 0.00357ExactMolWt + 0.00676EState_{VSA4} - 0.0348Chi1v \\ & - 0.000161VSA_{EState8} - 0.753xlogp - 0.0103EState_{VSA8} - 0.0618fr_{bicyclic} \\ & - 0.0345EState_{VSA10} - 0.00358PEOE_{VSA12} - 0.0134NumAromaticRings \end{aligned}$$

[3.7]

Although this equation still includes the terms for T_m and log P seen in the original and refitted GSE equations in this work, the statistical measures indicate an improvement justified by the addition of descriptors, of around 10% for both the development and training sets from the refitted GSE. Interestingly, a large number of the new descriptors include fragment counts or other structural information. This corresponds to the findings of other investigations that have discussed the inclusion of structural information, such as those discussed in the introduction to this chapter.

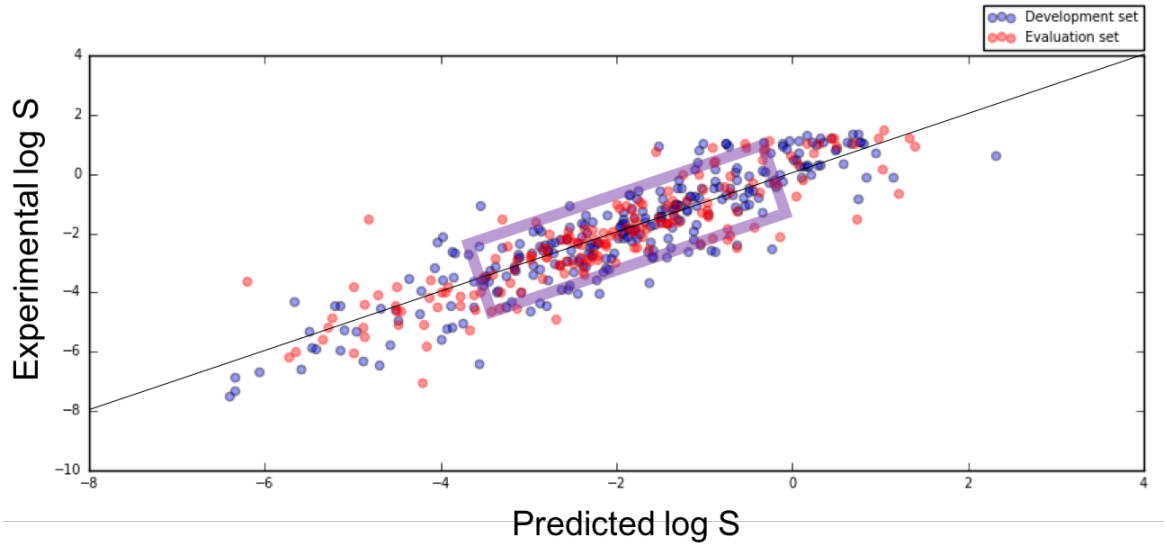


Figure 21 - The best regression method from the descriptor filtered analysis, plotted with the estimator refit to the full development set (blue), and tested on the evaluation set (red). The region highlighted by the purple rectangle corresponds to the best predictions.

The appearance of elastic net and lasso based methods in the best models selected is not surprising. The choice to reduce the number of descriptors in the model means that ridge regression models will not appear amongst the selected results, as ridge does not remove descriptors completely from the model (see 2.3.3). Elastic net models select descriptors more efficiently than lasso models, as they mix the l_1 and l_2 regularisers (see 2.3.3), meaning that

important descriptors are not left out of the model as a by-product of random feature selection in grouping, as can be the case for lasso models. Addition of selection criteria (either CV, AIC or BIC) to the lasso model can further alleviate this problem, by selecting the best descriptors from a group of highly correlated descriptors, which explains why the LassoCV and LassoIC models also appear in the best models.

In section 3.1, the importance of structure and solid-state effects on solubility was highlighted. The appearance of fragment count descriptors in the best models for log S prediction selected by BruteSis suggests that there are subtle solute-solvent interactions that are structure specific, and not well described by the balance between the solid-state effects represented by T_m and the lipophilicity of the molecule described by log P in the standard GSE. Although the GSE has been found to perform well in the past, evaluating its performance with the diverse set of structures in the dataset we have compiled has exposed some of its potential limitations. In the following chapters, we will develop the idea of using specific structural data to develop a solvation model in a knowledge-based approach. Using existing structural data from experiment we will aim to add this specific information to existing approaches, and in doing so attempt to improve them.

Chapter 4

Probing the Average Distribution of Water in Organic Hydrate Crystal Structures with Radial Distribution Functions (RDFs)

*Parts of this chapter are published in; Skyner et al. CrystEngComm., 2017, **19**, 641-652*
The programs developed and described in this chapter are available in Electronic Appendix II

4.1 Introduction

The abundance of hydrates in the CSD reflects the common role of solvent in the crystallisation process. An understanding of this is therefore of paramount importance for crystal engineering, with solvent choice often influencing the crystal structure and properties; either by formation of a solvate or hydrate, by directing the molecular conformation, or by favouring a particular crystal packing.

CSD surveys are applicable to the investigation of non-covalent interactions, such as hydrogen bonding within organic hydrate crystal structures. The systematic analysis of hydrates was, until recently, often confined to inorganic structures[114].

Recent surveying of organic hydrates has served primarily as a tool for the classification of the role of water within the crystallisation process and in overall structure. A commonly accepted classification system organises water sites within crystal structures into three categories and several sub-categories. The primary categories are isolated lattice sites, lattice channels and metal-ion coordinated water[115]. This classification system is summarised in table 4. Other survey studies have also considered the driving force for hydrate formation[116–118].

A further method of classification specifically orientated at the distribution of water within crystal structures was recently developed by Infantes and Motherwell[119]. Their work encompasses the use of the CSD search program, ConQuest[120], to obtain a dataset of crystal structures, with a final set of 1424 structures. The stringent search criteria of their survey specify that structures must have at least a single water-water contact with an O···O distance < sum of vdW radii (3.04 Å), no disorder or errors, and with an R-factor of < 10%. From the data analysed, water networks were defined in terms of clusters. Those identified were primarily described as discrete rings and chains, infinite chains and tapes and layer structures. Rings and chains are primarily described as patterns of 4-membered water rings, and chains with a repeat motif of four waters. Tapes primarily consist of linked 5-membered rings with one shared ring edge, and alternate 4,6-membered rings sharing a single edge.

Table 4 - Descriptions of Morris' Crystal Hydrate Classification System

Class	*Category*	*Description*
1	**Isolated lattice sites**	Water molecules are isolated from interaction with other water molecules by the intervention of other molecules.
2	**Lattice channels**	Water forms lattice channels. Water molecules lie in columns along unit-cell axis, forming channels.
2a	**Expanded channels**	Channels within these structures may take up extra moisture when exposed to high humidity.
2b	**Lattice planes**	Water occurs in a 2D plane.
2c	**Dehydrated hydrates**	May in principle belong to any other class. Crystals in this class dehydrate on removal from the mother liquor.
3	**Metal-ion coordinated water**	Contain metal-ion coordinated water.

Further work by Infantes and Motherwell[121] describes extended motifs constructed from water and chemical functional groups in organic molecular crystals. Infantes and Motherwell's work describes extended patterns of hydrogen bonding between chemical groups and water. They conclude that the ring, chain, tape, and layer patterns discussed in previous work are also predominant in larger hydrogen bonded networks with further bond donors and acceptors investigated. The work also discusses the specific role of H-bond (electron pair) donors and acceptors in the networks, which ties in to a secondary purpose of CSD surveys; the investigation of physiochemical features and structure-property relationships.

Desiraju[116] (1991) investigated the ratio of hydrogen bond donors and acceptors within a dataset of 411 crystal structures, toward the prediction of the likelihood of an organic hydrate crystallising. The majority of structures that formed hydrates were found to have a higher ratio of donor groups than acceptor groups, and thus it was concluded that hydrate formation probably compensates for this ratio mismatch.

A study directed at more 'biological' molecules was conducted by Jeffrey and Maluszynska[122] focusing on the stereochemistry of water molecules in the hydrate structures of small biological molecules. The dataset of 311 molecules, including the hydrates of amino acids, peptides, carbohydrates, purines and pyrimidines, and nucleosides and nucleotides, revealed a multitude

of hydrogen bonded interactions at a distance of ~3.0 Å from the water oxygen. Hydrogen bond acceptors of one hydrogen bond were found to be more prevalent than those accepting two. Only nine examples of water not acting as a hydrogen bond acceptor were found, with only one not donating two hydrogen bonds. An overall total of 16 hydrogen bond configurations were found, with water found to be a stronger acceptor than donor.

A recent discussion by Mascal *et al.*[123] considers novel coordination environments, specifically in relation to hydrates. Emphasis is placed on the abundance of hydrates within the CSD, implying that any discussion of hydrates should first consult the CSD. It is suggested that existing work directed toward characterization of water motifs adequately describes the variety of possible motifs to an appropriate standard of notation, and the authors refer to their own work[119,123]. This assumption is supported by the classification of apparently novel motifs by the authors' own classification system and the classification of organic hydrates seems possible in the forms of either a three-category or a cluster-based approach. These methods of characterization are commonly accepted and cited within the literature.

Van de Streek and Motherwell have noted that "statistical surveys into the behaviour of hydrates are difficult due to the severe bias that is introduced at many levels[117]", however there may be scope within similar surveying techniques for the building of predictive models. For example, Galek *et al*[124] have utilised data available in the CSD to develop statistical models for hydrogen bond coordination behaviour (not limited to the study of hydrates). Their work describes the hydrogen bonding behaviour of over 70 unique atom types, and begins to make assessments of structural stability of hydrogen bonding environments in known crystal structures, showing potential for application of empirically or statistically derived models.

In this work we develop a method for the statistical analysis of organic hydrate crystal structures. Our model combines the radial distribution functions (RDFs) of multiple atom pairs from numerous organic hydrate crystal structures. We also compare water oxygen (OW) and water hydrogen (HW) RDFs to the work of Soper[125]. Soper evaluated neutron diffraction data for water and ice at a range of temperatures (220K to 673K) and pressures (up to 400 MPa) in the form of OO, OH and HH partial structure factors. Fourier transformation of these partial structure factors produces site-site RDFs. However, the presence of systematic uncertainties arising from diffraction experiments means that this transformation is not as intuitively straightforward as expected. Soper uses empirical potential structure refinement (EPSR) in order to fit a 3D computational water model as closely as possible to the pre-determined experimental structure factors, improving the reliability of the extracted RDFs. Preliminary comparison of our own data with all of Soper's water and ice functions showed that our functions fit best (from visual overlay) with ice at 220K, and water at 298K, both under ambient pressure. Thus, comparisons between these two models and our own RDF will be discussed in depth.

4.2 Methods

4.2.1 Calculation of RDFs

In order to test the predictive power of a RDF model applied to non-crystalline phases, we included atom positions in a cumulative plot. We used the common atom-typing algorithm of the AMBER forcefield, and calculated RDFs for all atom types found within small-molecule organic hydrates.

The dataset for building of RDFs was obtained from a search for any structure containing water as an independent entity in the CSD (CSD version 5.34, 2013).[126] Structures included in the dataset were selected with the following restrictions; 3D coordinates determined, R ≤ 0.05, not disordered, no errors, not polymeric, no powder structures, and only organic. All hydrogen positions were normalised according to the following criteria; C-H = 1.089 Å, N-H = 1.015 Å, O-H = 0.993 Å. The final dataset contained 5922 structures in total.

We developed a programmatic approach within MATLAB in order to automate the processing of the dataset, and to collate the results effectively for the building of RDFs.

The developed program's primary operation can be summarised by the workflow in Fig. 22, or as follows:

- Determine atom types according to AMBER forcefield definitions for a crystal structure .pdb file with Antechamber[127,128]
- Apply all crystallographic algorithms necessary to produce symmetry equivalent atom positions and to expand the lattice by one unit cell in each direction
- Sort all atoms for each structure into individual arrays
- Move the structure coordinate system origin to a target atom nucleus position (either water oxygen or hydrogen)
- Convert to a spherical polar coordinate system
- Calculate distance, azimuth and elevation for all atom pairs within a specified cut-off distance (15 Å)
- Repeat, moving origin for every target atom in the system
- Save data as a MATLAB workspace for manipulation with further routines

The libraries for all information relating to symmetry operations were developed from the existing Fortran library CrysFML[129], the Bilbao Crystallographic Server[130–132], and the International Tables[61]. Routines for RDF calculations were developed from I.S.A.A.C.S[133] and from Allen and Tildesley[134]. Atom type assignment is performed as an external routine through Antechamber[127,128]. Schematic representations of the atom types used in this study are shown in Fig. 23.

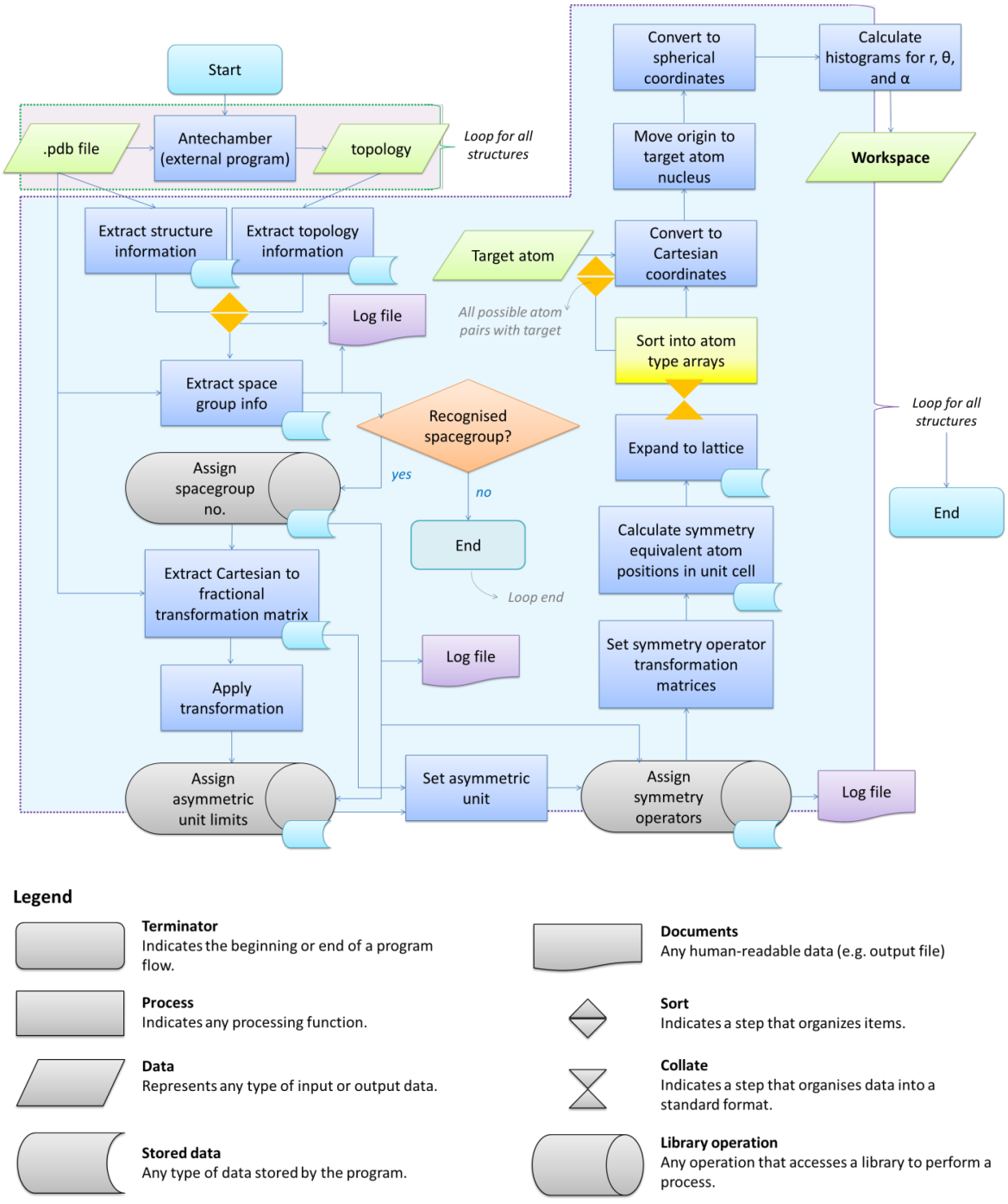


Figure 22 - Flow process diagram for program used to calculate information for RDFs

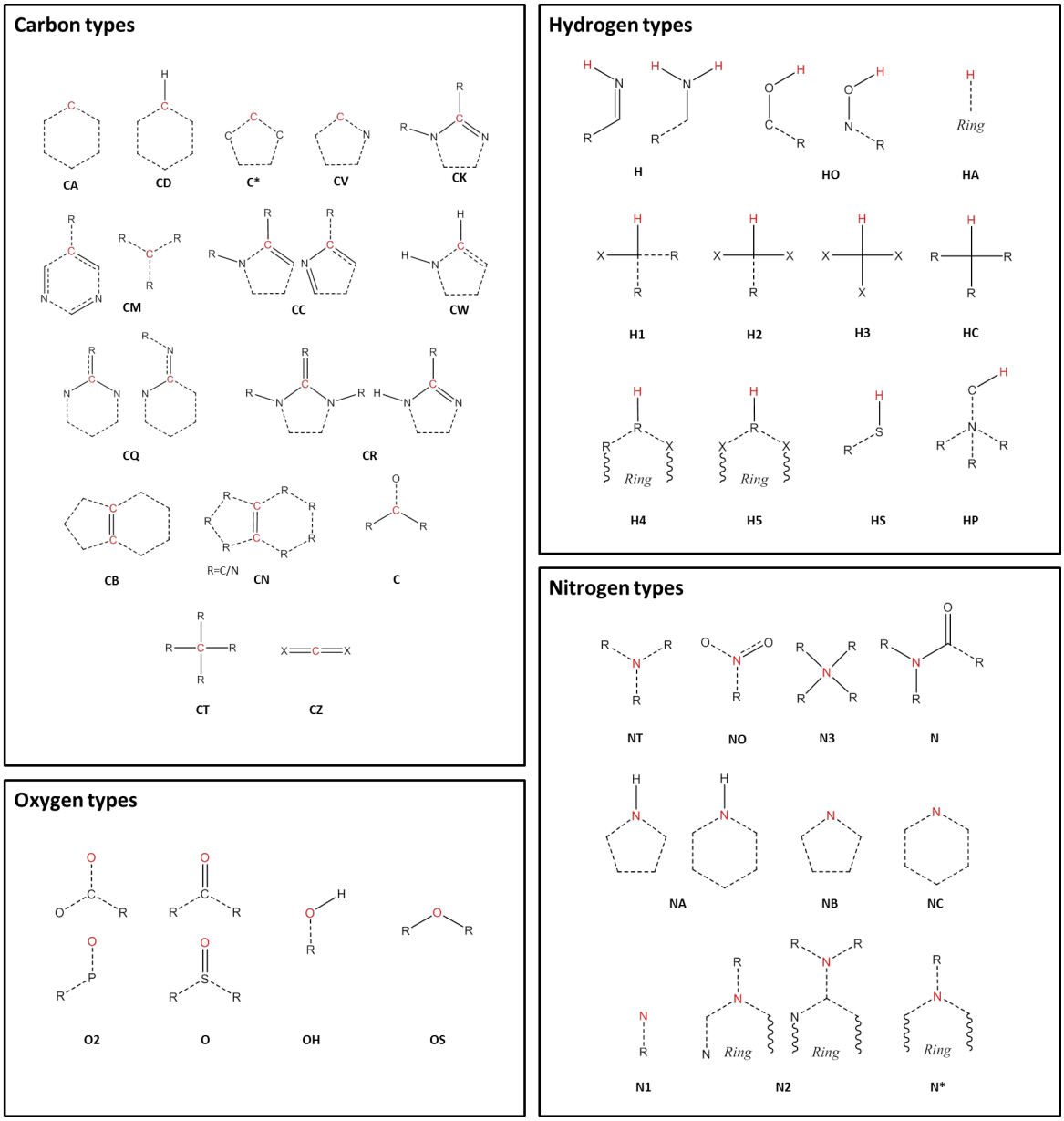


Figure 23 - Schematic representations of AMBER atom types. The red atom represents the atom that is being typed. The code below each schematic refers to the code assigned by the AMBER routine. R groups represent any atom, and X groups represent either N or O. Dotted lines represent undefined bond order, and solid lines represent conventional nomenclature of bonds.

4.2.2 Deconvolution of water RDF by water motif

In order to break down the contribution of particular arrangements of water (within organic hydrate crystal structures) to the average distribution of HW···OW, as represented by our RDF, an investigation into the specific motifs present within our dataset was conducted.

The identification of motifs (as defined by Infantes and Motherwell[119]) was conducted using the CSD-Materials module, available in the current release of Mercury.[135] The selected motifs are represented in Fig. 24. The motifs can be separated into: infinite chains, discrete chains, discrete rings, and infinite tapes in one dimension.

The search criteria for water motifs ignores specific hydrogen bonding interactions, and simply defines a network by an O···O distance < sum vdW radii + 1 Å. Therefore, quantification of the intermolecular pair distances (H···W) is not directly possible from the search results themselves. In order to assess these interactions, the pair count histograms were selected from the original dataset, and a new RDF calculated for each motif.

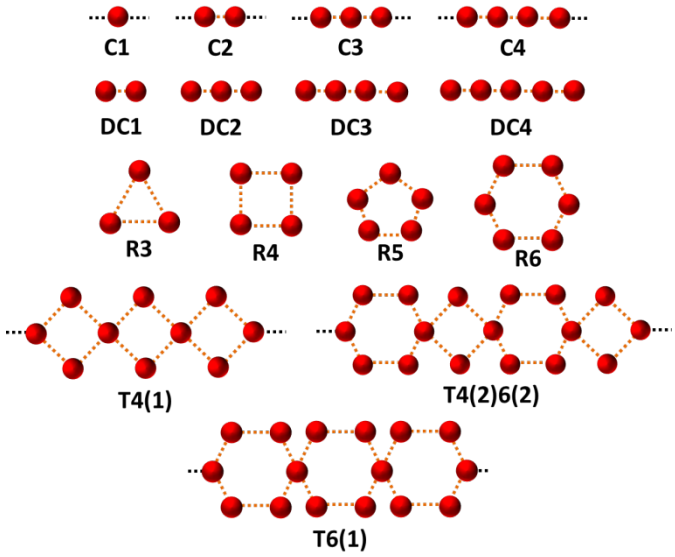


Figure 24 - The 15 water motifs used in this work. The motifs can be separated into; infinite chains (C1, C2, C3, C4; where the number represents the number of unique waters present before the motif is repeated), discrete chains (DC1, DC2, DC3, DC4; where the number represents the number of contacts between waters in the chain), discrete rings (R3, R4, R5, R6; where the number represents the number of waters in the ring), and infinite tapes in one dimension involving rings (T4(1), T4(2)6(2), T6(1); where a number outside of brackets represents the number of waters in the ring motif, and a number inside of brackets represents the number of waters from this ring also involved in a neighbouring ring). Nomenclature adapted from Infantes and Motherwell[119]

4.3 Theory

RDFs are simply calculable from crystal structures by evaluating all interatomic distances of atom pairs, binning them into a histogram, and then normalising with respect to an unbiased distribution of the same number of atoms – hence accounting for the intrinsically increasing numbers of pairs at larger values of *r*. This is demonstrated for a heterogeneous system in the equation below;

$$g_{\alpha\beta}(r) = \frac{dn_{\alpha\beta}(r)}{4\pi r^2 dr \rho_{\alpha\beta}} \quad [4.1]$$

where $\rho_{\alpha\beta}$ represents the number density of pairs in the entire system volume, and $n_{\alpha\beta}$ represents the number of pairs comprising atoms of species α and β. This function gives the probability of finding an atom of species β at a distance *r* from an atom of species α. The RDF for a particular material is often described graphically as a function of distance, *r*, with respect to the reference particle. The overall profiles of the plots of RDFs differ, depending on phase of matter, and the

order present. For RDF plots of a crystal structure, *g(r)* is represented by a series of short spikes, which indicate the existence of particles at specific and definite locations. This regularity can be extended almost infinitely until the crystal edge, illustrating the long-range order that, at least ideally, symmetry imparts to crystal structures.

The profile of a liquid radial distribution function differs greatly. The function represents an average of particle locations, conversely to the precise positions depicted in crystal structures. When a crystal melts to liquid, long-range order is lost, and at large distances there is an equal probability of finding a second particle in any shell of equal volume. However, at short distances close to the reference particle there may be some remaining order, a vestige of that found in the crystal phase. The nearest neighbours of the reference particle may still approximately occupy their original positions. Thus, it is often possible to identify an average sphere of nearest neighbours in the first and perhaps the second shell r_1 and r_2 from the reference particle[136].

A useful description of the energetics of a solution can be extracted from the Potential of Mean Force[137] (PMF), which describes free energy changes of the system as a function of a coordinate or coordinates. A popular choice for the coordinate is the distance *r,* due to the simplicity of calculation.

For a given *r* between two molecules, the PMF describes an average over all orientations of the surrounding solvent molecules. RDFs are directly related to the PMF $w^{(2)}(r)$ by;

$$g(r) = exp\left(-\frac{w^{(2)}(r)}{kT}\right) \quad [4.2]$$

where (2) denotes the number of atoms or particles to be considered. Thus;

$$w^{(2)}(r) = -kT \ln g(r) \quad [4.3]$$

The Helmholtz free energy *A(r)* can be expressed as;

$$A(r) = -kT \ln g(r) + a \quad [4.4]$$

where *a* is a constant chosen so that the most probable distribution between two particles gives a free energy of 0.

The PMF can be used to describe the energetics of the whole system. An appropriate weighting scheme applied to empirically parameterised RDFs can then be utilised within computational algorithms for the simulation of systems in solution. This reduces the computational cost associated with explicit solvent models, whilst improving some of the inaccuracies that implicit solvation models suffer due to their inherent approximations.

4.4 Results

4.4.1 Structure of water in hydrates

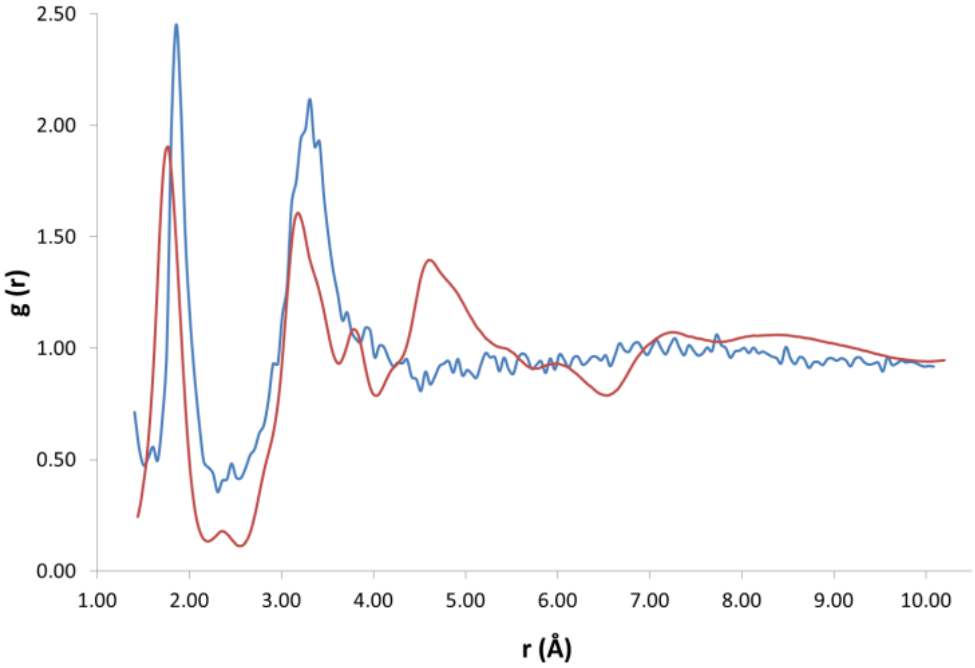


Figure 25 – Soper's OW···HW EPSR RDF of ice at 220K (red) and our OW···HW RDF model (blue), both with the OW···HW intramolecular interaction peak removed.

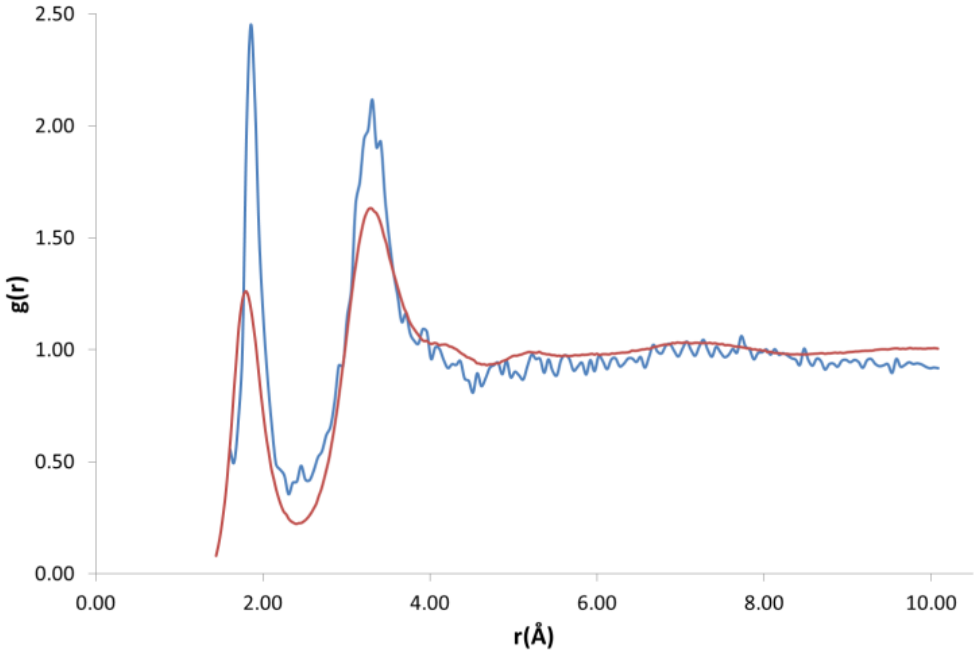


Figure 26 – Soper's OW···HW EPSR RDF of water at 298K (red) and our OW···HW RDF model (blue), both with the OW···HW intramolecular interaction peak removed.

Our initial expectations were that only the direct intermolecular interactions (equivalent to the first solvation shell) would be deducible from the calculated RDFs, and that difficulties would arise in relating the distributions to the equivalent solution phase information. However, a comparison of our RDF for HW and OW with Soper's RDFs for ice (220K; Fig. 25) and water (298K; Fig. 26) does show some interesting correlations beyond the first solvation shell.

It is important to determine whether the discrete features observable in the RDF are in fact noise, or signal. There are two possible scenarios: A) The features present are noise, due to an insufficient amount of data, meaning the distribution is not entirely representative of a smooth and average distribution within hydrates; B) The features present are signal, comprising a number of discrete peaks occurring due to the complexity of the water networks or motifs found in organic hydrates.

Fig. 27 (bottom) shows Soper's EPSR model for ice at 220K parameterised from neutron diffraction data (red), and our RDF (original: dotted black line, smoothed function: blue) resulting from all water oxygen to water hydrogen pair distances found within our dataset (5922 structures). It can be seen that there is a shift of the first two observable peaks to higher values of *r*, and the absence of the third peak observable in Soper's function. The peaks and troughs of the RDF profile also occur at different values of g(r). This difference is highly relevant if the model data from our RDF data are to be applied to predictive models in the future, particularly in the conversion of RDFs to PMFs, as the logarithmic relationship between g(r) and w(r) means that a small change in free energy (a small multiple of kT) can correspond to a change in g(r) of an order of magnitude from its expected or most likely value. However, one structural feature unique to the Soper ice RDF, which doesn't occur in the Soper water RDF, also appears to be present in our RDF; namely, the presence of a small peak in the trough between the two large peaks representing the first and second hydration shells, between 2-3 Å.

Overlaying the OW···HW RDF with Soper's model of water (298K) provides a better fit in terms of peak positions, as shown in Fig. 27 (top; original: dotted black line, smoothed function: blue). However, discrete features unique to the solid state of ice are not present in Soper's liquid water function.

If the RDF model is compared to this subtle peak in Soper's water model, it can be seen that the maxima of the peaks in its profile, although quite noisy, fit the shape of the water profile well. No smoothing function has been applied as part our own method, however Soper fitted his data to inherently smooth computational models of water and ice.

A visual comparison of the short-range interactions discussed above is also summarised in Fig. 27. In both images, we have applied the Savitzky-Golay smoothing algorithm[138] to our data (shown as a blue line, with the original data as a black dotted line) simply for the purpose of producing this figure, in order to increase the signal-to-noise ratio without unduly distorting the original data. In the top image, we compare this to Soper's 298K water model, and highlight three areas where our own RDF displays features that are not explained by the water model. Namely, a large shoulder on the right of the first interaction peak, at ~2.15Å, a smaller shoulder on the left of a second interaction peak, at ~2.85Å, and a third small but independent peak at ~4.16Å. We have also indicated peaks that are explained by the water RDF, as indicated by the blue and red arrows, highlighting the peaks in their respective plot colours.

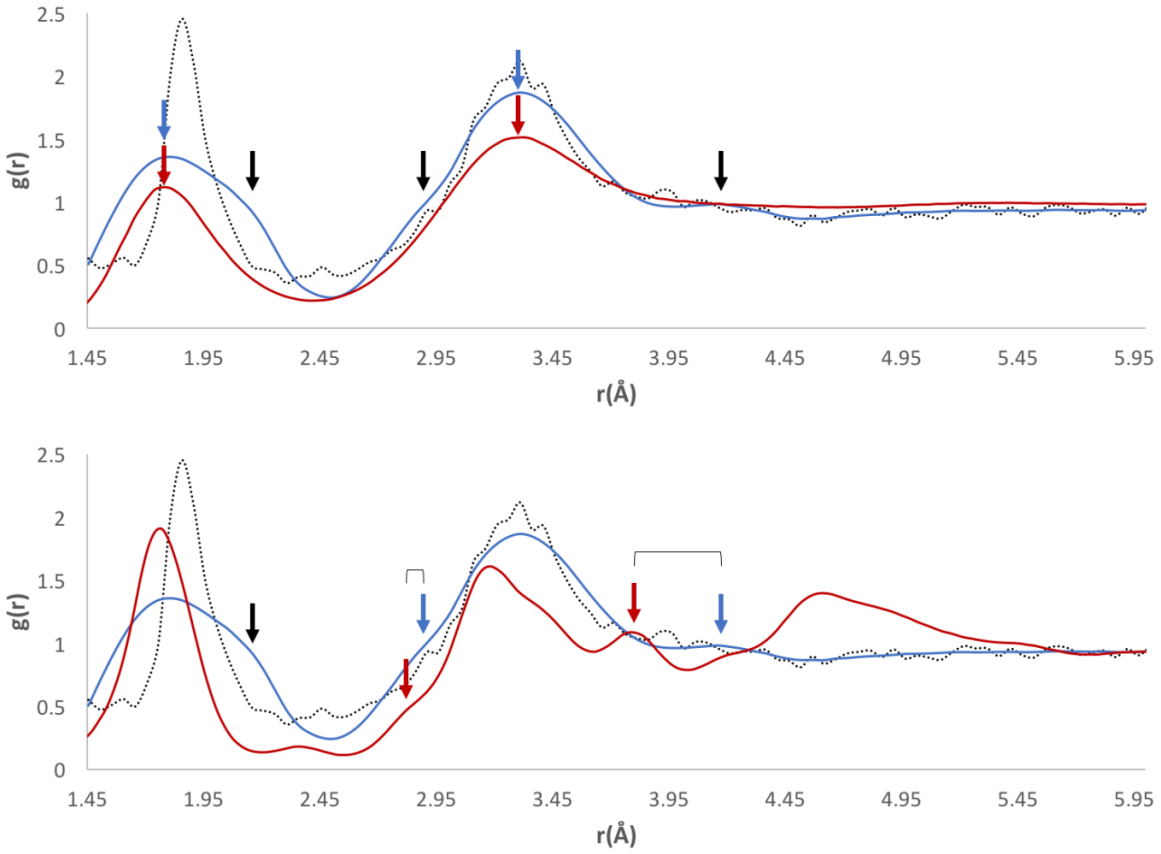


Figure 27 – A comparison of the short-range interactions in our RDF for OW···HW pairs (original data shown as dotted black lines, smoothed data shown in blue) with Soper's RDF of water at 298K (shown in red on the top plot) and ice at 220K (shown in red on the bottom plot). The black arrows on both plots represent peaks or features in our RDF which cannot be explained by the comparative Soper plot. The blue and red arrows indicate comparable peaks, with their colour corresponding to the same coloured plot line.

In the bottom image, we compare our smoothed profile (blue) to Soper's ice RDF (red), and attempt to indicate sources for the unexplainable peaks from the ice profile, as indicated above. The first shoulder, indicated by the only black arrow in the bottom image, is not confidently explained by either of Soper's distributions, and is probably due to the broad distribution of data in the first solvation shell, and between the first solvation shell and the second solvation shell.

The overall shape of our profile correlates well to that of Soper's water profile. However, certain features present in Soper's ice RDF also appear in our RDF; i) a peak at 2.9Å that becomes a shoulder on the peak at 3.3Å when a smoothing algorithm is applied, corresponding to a similar feature of Soper's 220K ice function, at 2.8Å and ii) a peak at 4.1Å, which is emphasised upon the application of a smoothing algorithm, corresponding to the third solvation shell, present in Soper's 220K ice function at 3.8Å. This suggests that some order found in a typical ice model is also present in the overall structure of water in organic hydrates. In liquid water, this order is lost, meaning that Soper's water model no longer contains these interactions. However, the peak positions in our RDF correspond more closely to those present in Soper's liquid water model than to the ice model.

The presence of peaks in similar positions to Soper's water function in our RDF may suggest that our data are most representative of systems at 298K, implying that water networks within hydrates have similar interaction distances to liquid water. This may result from the measurement temperature of the original data; over half of the contributing structures (3659) were measured above 261K. However, it could also be an indication of peak broadening in the RDF due to the diversity of structures within our dataset. Beyond the second solvation shell, the RDF appears to be noisy.

Additional consideration was given to the measurement temperature at which the crystallographic data were obtained. The data were separated into three 50K temperature intervals, and one interval where the temperature was above 261K. These intervals were chosen based upon the distribution of measurement temperatures across the whole dataset, with a large number of structures (over half of the dataset) being measured at ~298K. Next, the OW···HW RDFs were recalculated for each temperature interval. The resulting functions are shown in Fig. 28.

The positions of the peak maxima representative of the first and second solvation shells do not change, unlike the Soper functions. This is because of the normalisation of hydrogen bond lengths, done because hydrogen positions are notoriously difficult to assign in crystal structure solution and refinement. Unfortunately, this means that subtle differences in the data, reflecting the variation in lengths of covalent bonds to hydrogen, may occasionally be lost. However, it is unlikely that the data would be any more accurate or reliable should the hydrogen bond lengths not be normalised, and perhaps more errors would be incorporated into the data from unreliable bond lengths due to the unreliable assignment of hydrogen positions in the experimental data.

The only observable difference between the measurement temperature separated data are the values of g(r) at which the peak maxima occur, although there is no observable pattern to explain this. The number of contributing data were considered as a cause, but recalculating the functions with the same number of contributing structures for each temperature range produced similar results. The larger oscillations seen in the results at 211-260K are due to there being fewer data in this range than in other intervals.

In order to determine whether discrete features at both short and long range were due to specific arrangements of water, further analyses of specific motifs were carried out.

We observe a better fit of the long-range pair distances to Soper's water model in comparison to the ice model. However, there is still a considerable amount of 'noise' present at long-range distances. This was investigated further by the overlay of the RDF with an RDF (calculated in I.S.A.A.C.S[133]) for Bernal's hexagonal ice structure[139]. However, statistical analysis of the long-range pair distances (> 4Å) for both of the Soper functions and also for the hexagonal ice function (Table 1) showed that the profile of water (298K) fits best, followed by ice (220K) and finally hexagonal ice. The statistical analysis (methods described in chapter 2) is shown in table 5.

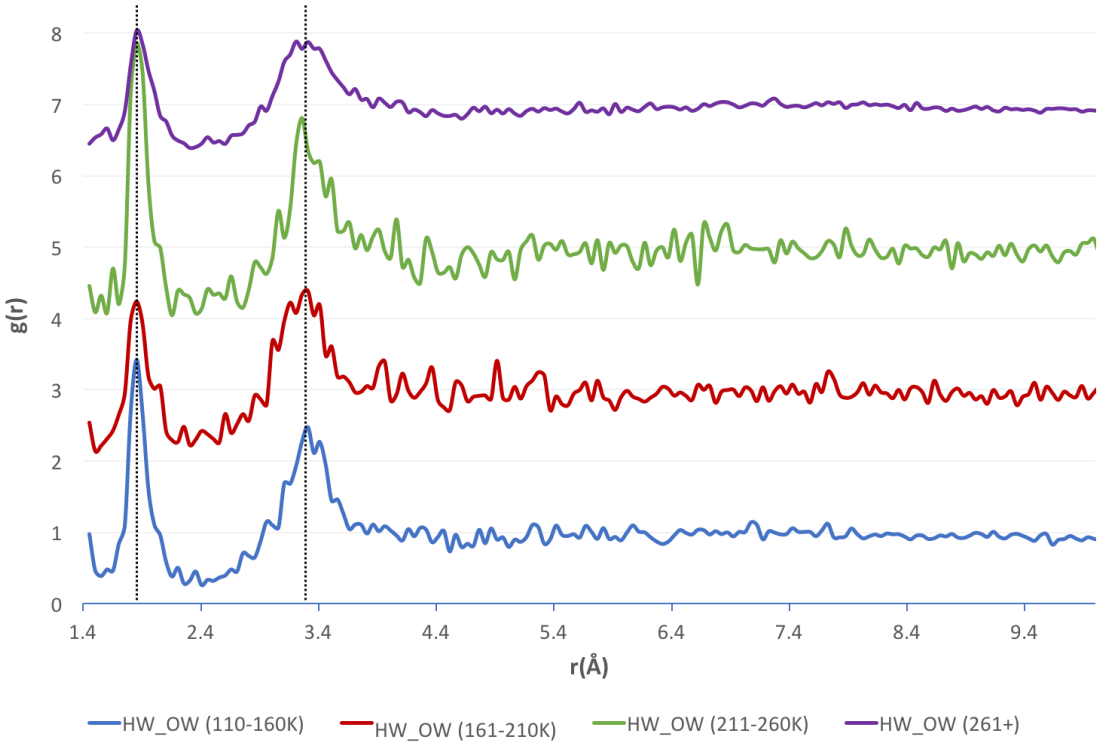


Figure 28 – The OW···HW RDFs for water, separated by temperature ranges, as indicated by the legend (bottom) with the functions stacked in order of increasing temperature.

Table 5 – A summary of the statistical analysis of goodness of fit (GOF) for the long-range pair distances of the OW···HW RDF with hexagonal ice, water (298K) and ice (220K) models.

	Hexagonal Ice	***Water (298K)***	***Ice (220K)***
RMSE	8.7	0.57	0.62
ln(L)	-640	-154	-170
AIC	1287	314	345
BIC	1297	324	355

4.4.2 Deconvolution of water RDF by water motif

A breakdown of the frequency and number of structures found for each motif investigated is shown in Table 6. Similarly to Infantes and Motherwell[119], the most frequently occurring motif type for our dataset was the discrete chain motif (17.4%), followed by infinite chains (10.4%), discrete rings (6.1%), and finally infinite tapes (0.96%). Part of the difference in frequencies found for each motif within our dataset is due to the more extensive set of motifs used in the original study (we have only used a small subset of common motifs for exemplary purposes). Other differences in the methodology include dataset size, and the method of motif assignment. The Infantes and Motherwell[119] study involved the manual identification of water motifs, whereas our own methodology used the CCDC's Mercury[135] software to automate the process, meaning that

the two processes use slightly different criteria to select examples of a given motif. Such differences may arise due to acceptance of discrepant ranges of site-site distances.

The purpose of recalculating RDFs for specific water motifs was to identify whether discrete features within the overall OW···HW RDF could be specific to a particular arrangement of water in organic hydrates observable in RDF plots. Initial analysis of the likelihood of this was performed by a simple overlay of each recalculated motif RDF with the original OW···HW RDF. It was found that peaks unique to the profile of particular motifs were also distinctly present in the original function. An example of this is shown in Fig 29.

In order to quantify the likelihood of these distinct features correlating to the features present in the original RDF (omitting r < 1.6Å), a statistical analysis of the goodness-of-fit (GOF) of each motif to the original RDF was conducted. The results of this analysis are shown in Table 6.

The following statistical measures were employed; Root Mean Squared Error (RMSE), R^2, ln(L), the AIC, and BIC. Here, we treat the original RDF as the 'true' model, and the motif RDFs as approximating models.

From the results of AIC and BIC analysis, the GOF for each motif was ranked (the same ranking applies for both AIC and BIC), as shown in Table 6. It was found that the DC1 motif fitted most closely with the overall RDF. It might be expected that this would be the case, as DC1 motifs appear most frequently in our original dataset. However, a regression of the AIC and BIC scores against the frequency of occurrence for all motifs found no correlation to suggest this.

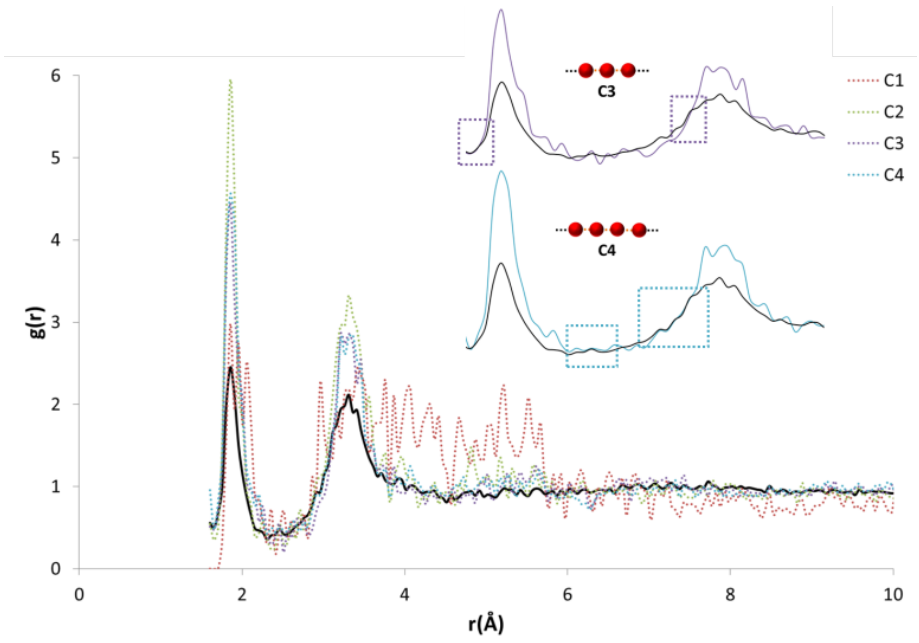


Figure 29 – An example of the initial overlay analysis of motif RDFs with the original OW···HW RDF. Discrete features for both the C3 (purple) and C4 motif (blue) appear to be present in the original function. Other discrete chain motifs are also represented here, as indicated by the legend (top right).

Table 6 – A summary of the motif search of our dataset, showing the frequency of occurrence (out of 5921 structures) and the number of structures found, and the results of the statistical analysis conducted to quantify the likelihood of distinct features in motif RDFs correlating to the features present in the original RDF.

Motif Type	Motif	Frequency (%)	Number of Structures	RMSE	R^2	ln(L)	AIC	BIC	Rank
Infinite Chain	C1	2.9	169	2.0	0.99	-361	727	736	13
	C2	3.9	229	1.6	0.99	-325	655	665	12
	C3	1.8	106	1.0	1.00	-249	505	514	6
	C4	1.9	112	1.3	0.99	-285	576	585	10
Discrete Chain	DC1	10.5	623	0.1	1.00	213	-420	-410	1
	DC2	2.8	164	0.4	1.00	-77	160	169	2
	DC3	2.6	155	1.0	1.00	-236	478	487	5
	DC4	1.5	89	1.1	0.99	-252	511	520	7
Discrete Ring	R3	0.4	24	2.3	0.98	-382	770	780	15
	R4	3.1	184	1.2	0.99	-273	551	560	9
	R5	0.8	49	1.2	0.99	-265	537	546	8
	R6	1.7	103	1.4	0.99	-296	597	607	11
Infinite Tapes	T4(1)	0.2	13	0.6	1.00	-163	332	341	3
	T4(2)6(2)	0.6	33	0.9	1.00	-229	463	473	4
	T6(1)	0.2	11	2.3	0.98	-379	764	774	14

4.4.3 Qualitative interpretation of RDFs

The values of g(r) and *r* found for each atom type are plotted against each other in bar charts in Fig. 30. Comparison of the most prominent peak positions for each atom type with OW vs each atom type with HW identifies whether, on average, the atom type is in closer proximity to the OW or HW of water. Comparison of the relative values of g(r) also gives an indication of which atom types are most likely to be in close proximity to water.

Carbon atom types. The calculated RDF profiles for carbon atom types generally show broad peak areas for pairs calculated with HW and OW, reflecting the lack of specific intermolecular interaction of water with carbon, and no definite orientation of water with respect to carbon. However, carbon atom types describing carbon in close proximity to an oxygen or nitrogen atom produced RDF profiles reflecting nearby interactions. For example, in the profile of the C atom type (Fig. 31), describing either an sp^2 carbonyl carbon or else an aromatic carbon with a hydroxyl substituent in tyrosine, the RDF maximum g(r) peak for C with HW occurs at lower *r* than the OW peak, indicative of the C-O···HW hydrogen bonding interaction (r = 2.86 Å; g(r) = 1.84). The profile also shows a secondary HW peak after an OW peak at *r* = 4.26Å, with a separation of HW peaks = 1.40 Å, roughly corresponding to the average distance separating the hydrogens within a water molecule. This suggests that the average orientation of water in relation to C-O occurs with HW-OW along the C-O vector.

A comparison of the profiles of the CC and the CK atom types (Fig. 32) gives an example of how using a sophisticated atom-typing algorithm may offer an advantage over using traditional element labels. Both atom types represent a carbon adjacent to a nitrogen in a five-membered

ring. The CC atom type can have any substituent, whereas the CK atom type has a hydrogen substituent (see Fig. 23). The first immediate difference between the CC and CK RDFs is the overall likelihood of finding carbon to water pairs.

The addition of a non-hydrogen substituent (*i.e.* In the CK RDF) produces a significant peak for CK···HW pairs that is not present in the CC···HW profile (*r* = 2.95 Å, g(r) = 2.63), as indicated by the peak highlighted in Fig. 32. This difference may seem intrinsic; however, these results exemplify how the atom-typing method is able to describe the major differences in water distribution introduced in the average case of substituent changes. This again corroborates the postulate that atom typing algorithms are useful in a quantitative survey of hydrate distributions, as conventional atom labels based on atomic number alone would not have identified this change in distribution.

Where substituent effects are not considered, there is little more to be learned from the RDFs of carbon atom types, as the distribution of water around such atoms is expectedly broad, and does not show significant patterns which cannot be observed within the RDFs describing substituent atoms of terminal ligands.

Nitrogen atom types. The peak analysis of nitrogen atom types revealed a distinct difference in the profiles of nitrogen atoms participating in N-H···OW and N···HW interactions. The profile of nitrogen groups participating in H-bond donor N-H···OW interactions show the highest g(r) OW peak to occur before the highest g(r) HW peak, as expected, and include the following atom types; N, N2, N3, NA and NT. Nitrogen atom types with profiles indicative of H-bond acceptor behaviour included N1, NB, and NC.

Oxygen atom types. The peak analysis of oxygen atom type RDFs revealed more distinct differences in profiles than those found in nitrogen atom type RDFs. For two of the oxygen atom types, O (Fig. 33) and O2 (Fig. 34), representing carbonyl and carboxylate oxygen respectively, the overall profile of peaks were similar to those found for the H-bond acceptor groups in nitrogen atom type RDFs. The primary difference between the O and O2 RDFs is the comparative g(r) values of the HW and OW highest peaks. For the O atom type, the maximum g(r) value for OW is greater than for HW, whereas for the O2 atom type, both the OW and HW peaks have similar values of g(r).

The RDF profile for the OH (Fig. 35) atom type, representing alcohol oxygen, differs somewhat from the O and O2 atom types, reflecting the ability of an alcohol group to participate in both H-bond donor and acceptor interactions with water.

The first obvious difference in the OH RDF occurs for OH···HW pairs, where a definite intermolecular interaction is represented by a sharp and narrow peak. This peak represents the alcohol oxygen participating in H-bond acceptor behaviour, O···HW. Two further peaks are also present at r similar to those found in the O and O2···HW pair RDFs (r = 1.86Å and 3.21 Å). These peaks are increasingly broadened, suggesting less definite positions and orientations of water as r increases. A high g(r) value peak occurs in the OH···OW RDF at r = 2.81 Å, which is the same r for the highest peak found in the O2···OW RDF, suggesting a similar mode of interaction.

Interestingly, for the OS atom type, the largest peak in the RDF for HW is found at a distance (~4.6Å) not indicative of hydrogen bond formation. The OS atom type represents an ether or ester

oxygen. It is known that there are few examples of ester hydrogen bonding in the CSD.[140] A study[141] into ether and ester hydrogen bond formation found that ester oxygen hardly participates in hydrogen bonding. For (E)-esters, this is because of competition with the adjacent carbonyl group. For (Z)-esters, this is because of destabilization due to a repulsive electrostatic interaction by the carbonyl group. Ethers were found to form hydrogen bonds at longer distances than expected, suggesting the bond is readily elongated by competing interactions.

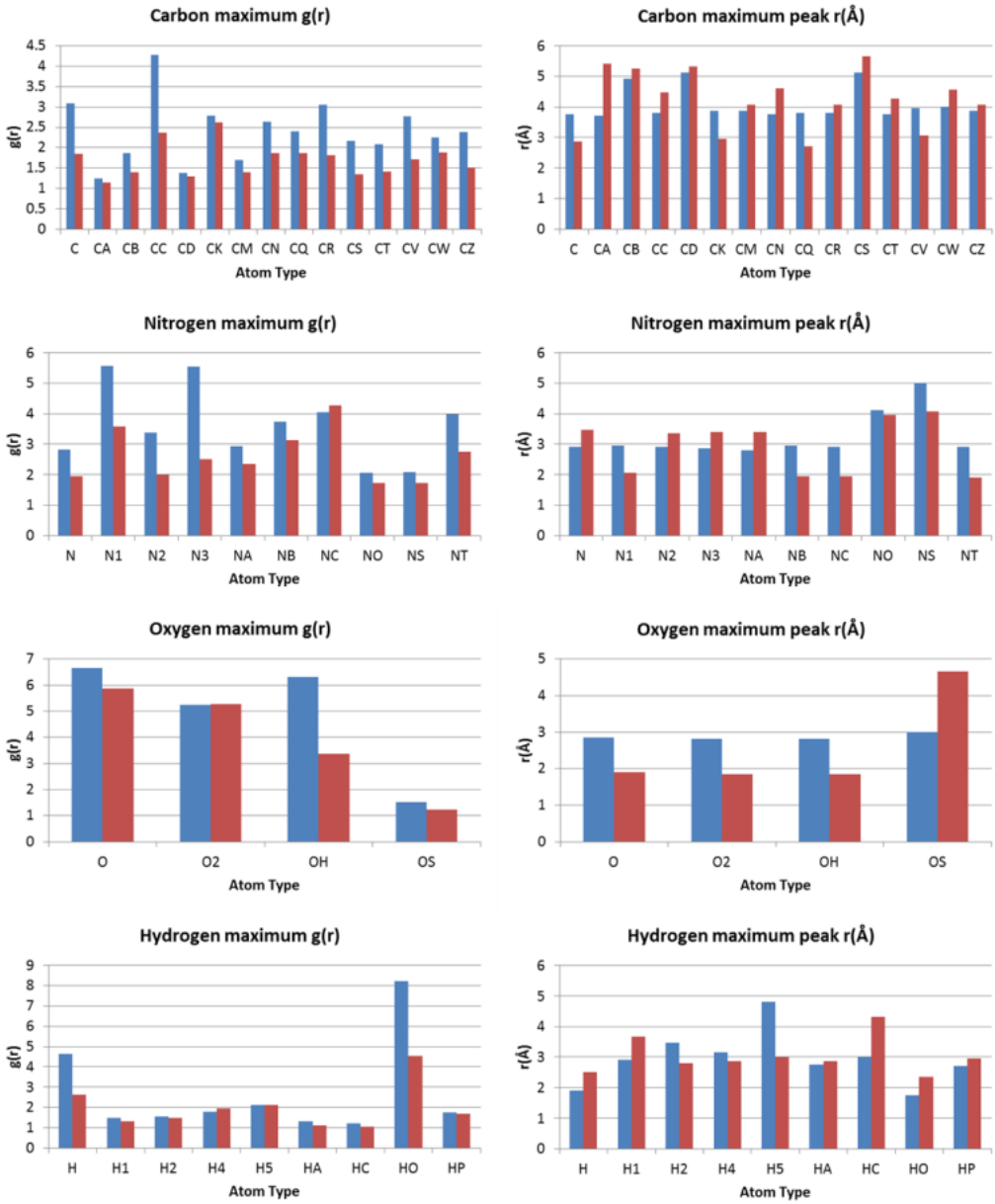


Figure 30 - The maximum peak value of g(r) for each RDF pair profile (each atom type with HW and OW) was determined. These bar graphs show the g(r) value for the maximum peak of each atom type with OW (blue bars) and HW (red bars) on the left, with the distance at which these peaks were found plotted on the bar graphs on the right.

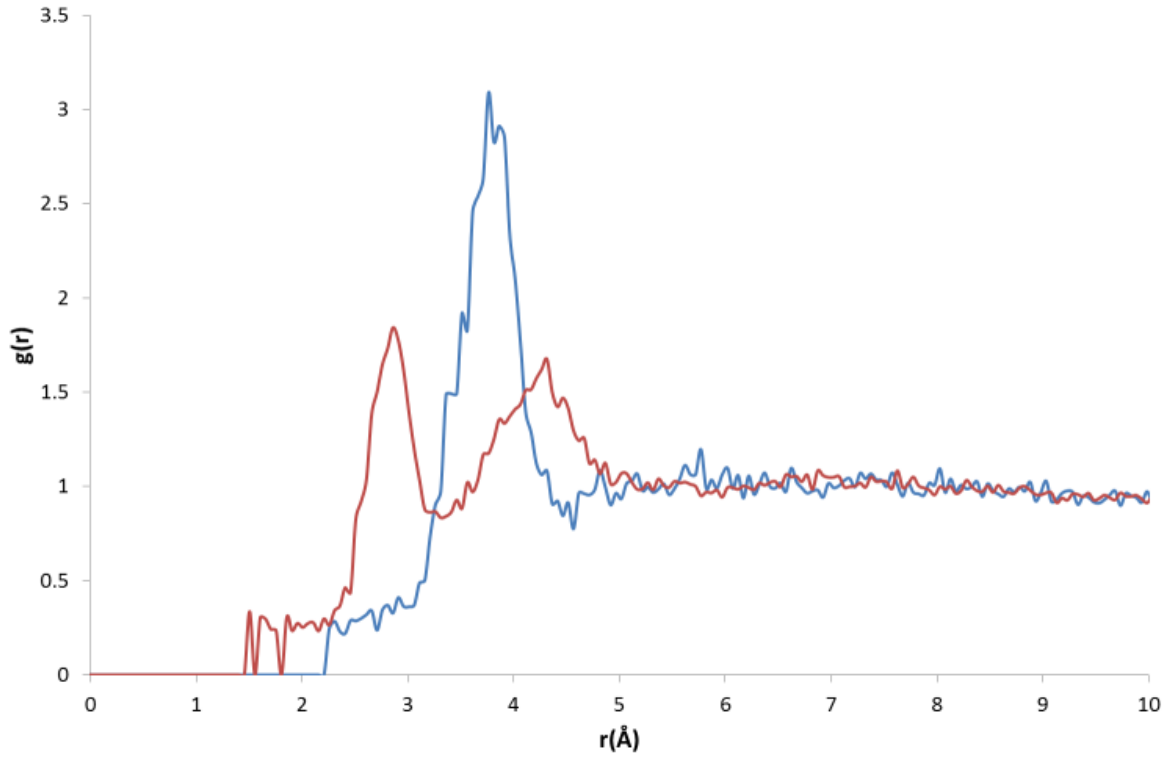


Figure 31 - AMBER RDF profiles for atom pairs of C with OW (blue) and HW (red).

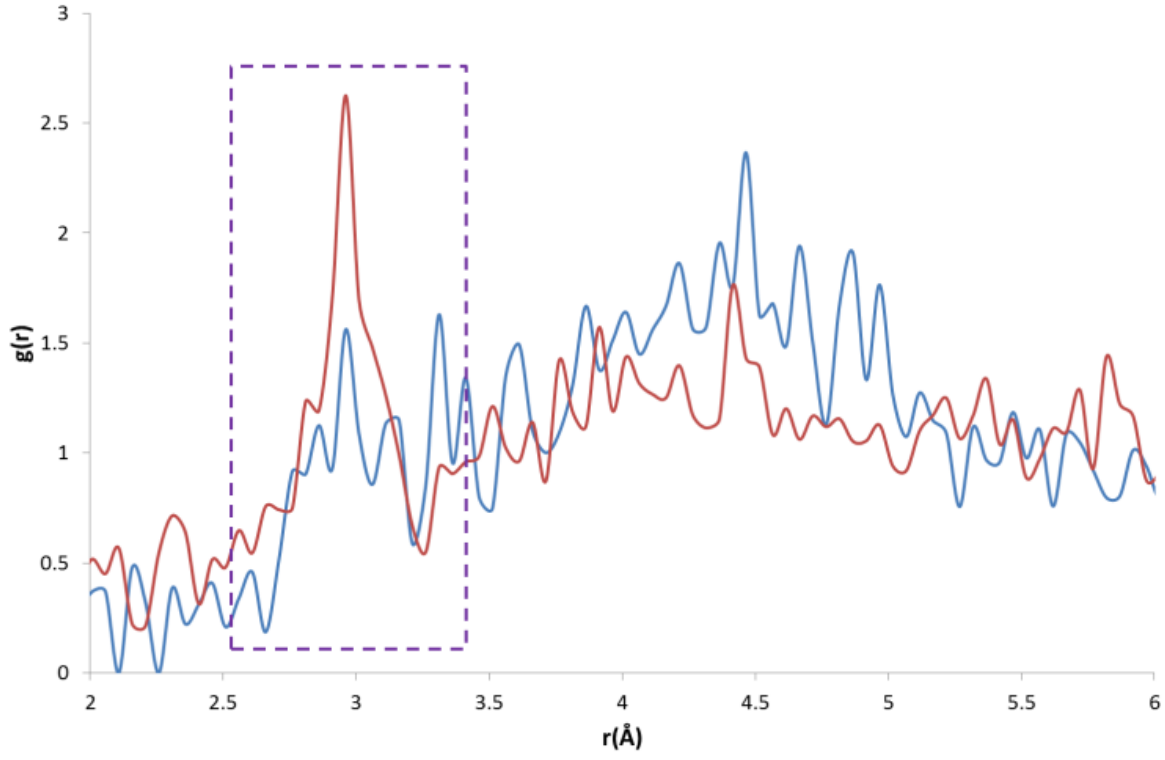


Figure 32 – CC···HW RDF (blue) and CK···HW RDF (red) with a much larger peak apparent at ~3Å in the CK···HW profile (outlined in purple).

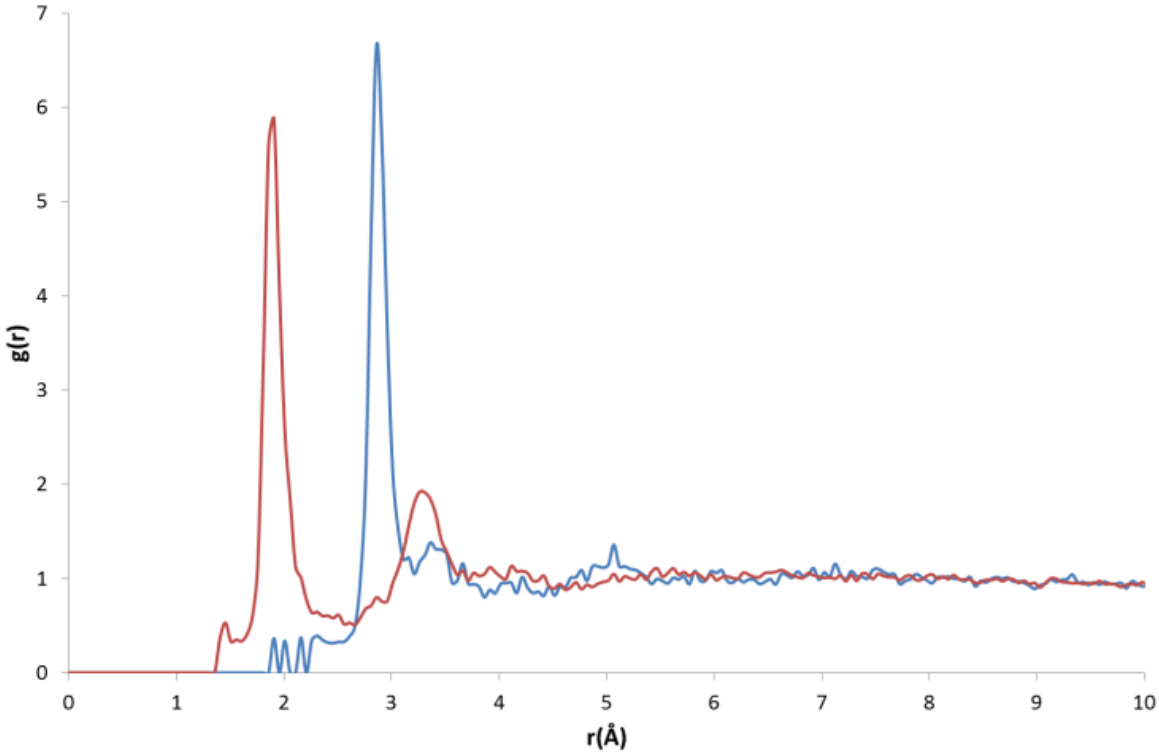


Figure 33 - O atom type with OW (blue) RDF and HW (red) RDF

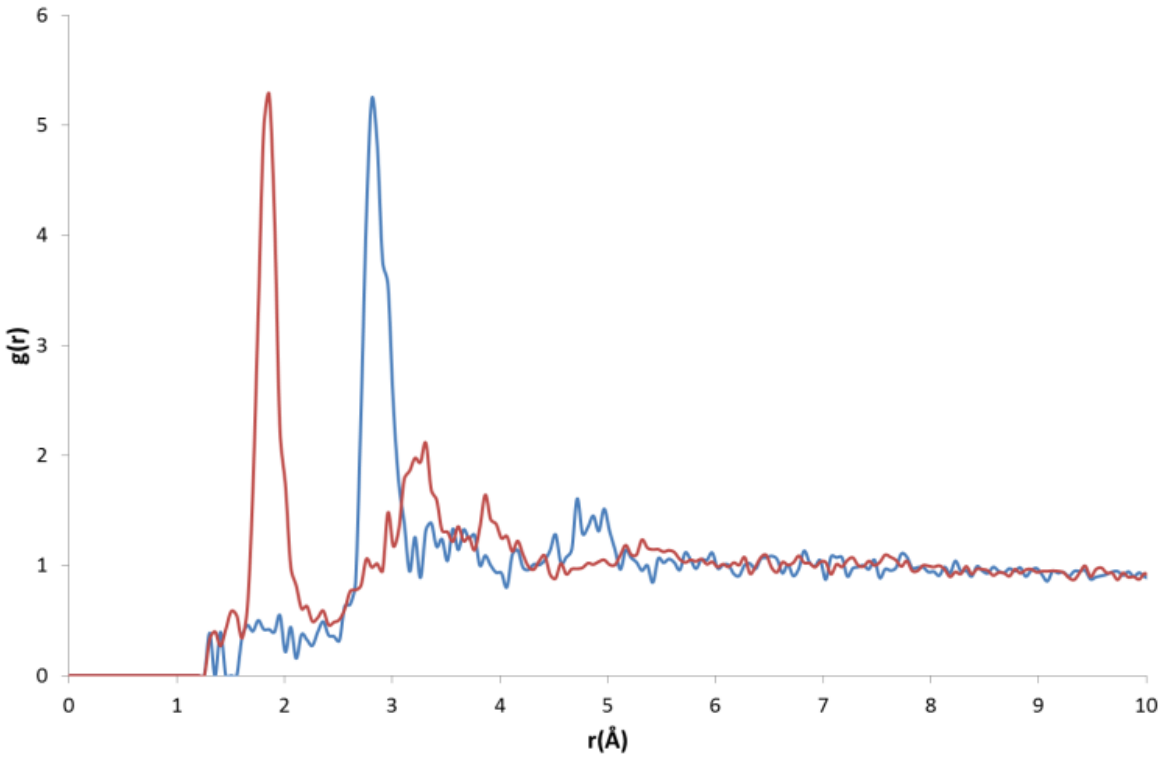


Figure 34 - O2 atom type with OW (blue) RDF and HW (red) RDF

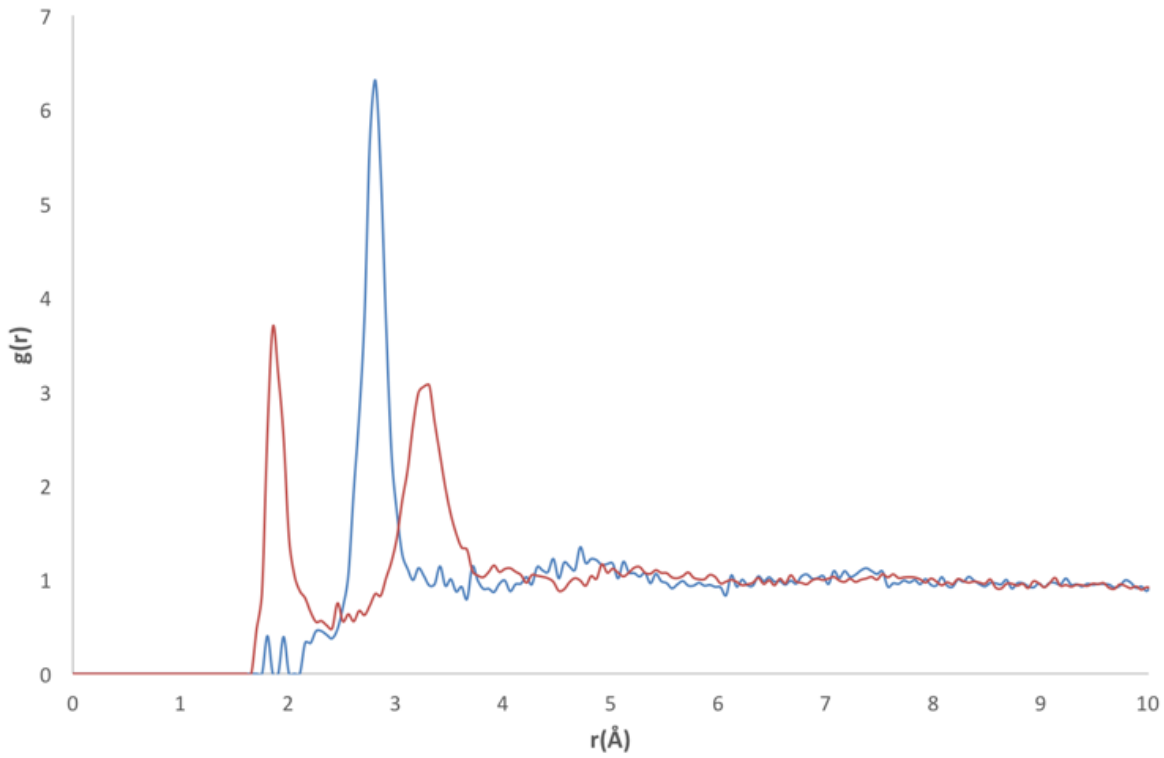


Figure 35 - OH atom type with OW (blue) RDF and HW (red) RDF

Hydrogen atom types. Peak analysis of RDFs describing hydrogen atom type pairs with OW and HW revealed two distinct overall profiles. The first type of profile has sharp and narrow peaks, indicating direct interaction with water, with a well described average orientation of water around the respective atom types. The second profile shape represents no direct interaction of water with the respective hydrogen atom types, and presents as broad peaks at low values of g(r), suggesting fewer similarities between the pairs found in the structures used to build the RDFs, and less definition in the average orientation of water. Only two of the nine investigated hydrogen atom types showed profiles with distinct narrow peaks; H, representing hydrogen in an amide or imino group, and HO, representing hydroxyl hydrogen. Both profiles indicate distinct H...OW pairs for interactions, characterised by the appearance of a peak in the hydrogen HW RDF before a hydrogen OW peak.

4.5 Discussion

The analysis of the contribution to the overall profile of water (via interpretation of OW···HW RDFs) of individual motifs of water within hydrate structures showed that discrete features appear in the RDFs, even at long distances. This is indicative of their ability to capture 'real' interactions. It was expected that long-range pair distances would mostly comprise noise, as an artefact of the most commonly occurring symmetrically equivalent atom positions; therefore, the distinguishing of signal within these regions, attributable to particular arrangements of water, is promising for the application of RDFs in predictive methods.

Chapter 5

Developing Solvation Models: Application of RDFs

5.1 Introduction

In chapter three, it was shown that even when attempting to predict solubility from simple regression models, the inclusion of information about the specific structure – using either an atomic or a functional group description – of the solute is important. In chapter four, it was shown that a simple atom typing algorithm and RDFs calculated from organic hydrate crystal structures can describe the structure of solvent well, at least in the case of water-water interactions (pairs). In this chapter, we investigate whether the atomic information that is included in such RDFs can be applied to an existing theoretical model, namely 1D-RISM, as discussed in chapter two, to improve the calculation of Hydration Free Energies (HFEs).

Previous studies have also considered the inclusion of structural information not implicit in the RISM theory. For example, Ratkova *et al*[142] (2010) used several empirical corrections to RISM to estimate the HFE for a number of organic molecules. This combination is referred to as the structural descriptors correction (SDC). In the SDC model, the structural information included was in the form of structural descriptors: excluded volume, branch, double bond, benzene ring, hydroxyl group, halogen atom, aldehyde group, ketone group, ether group and phenol fragment descriptors. HFE values were compiled from a number of different sources, for 185 compounds in nine classes: alkyl, alkenyl, phenyl, hydroxyl, halo, aldehyde, carbonyl, and ether, and a final separate distinction of a phenol fragment. Molecules consisting of a single class of these fragments are referred to as simple solutes, and molecules consisting two or more fragment types are referred to as polyfragment solutes. 65 simple solutes were used as a training set for SDC calibration, with 120 molecules; 60 simple and 60 polyfragment, used as a test set. The differences between the experimental and RISM calculated HFEs are then targeted by a regression model including structural descriptors.

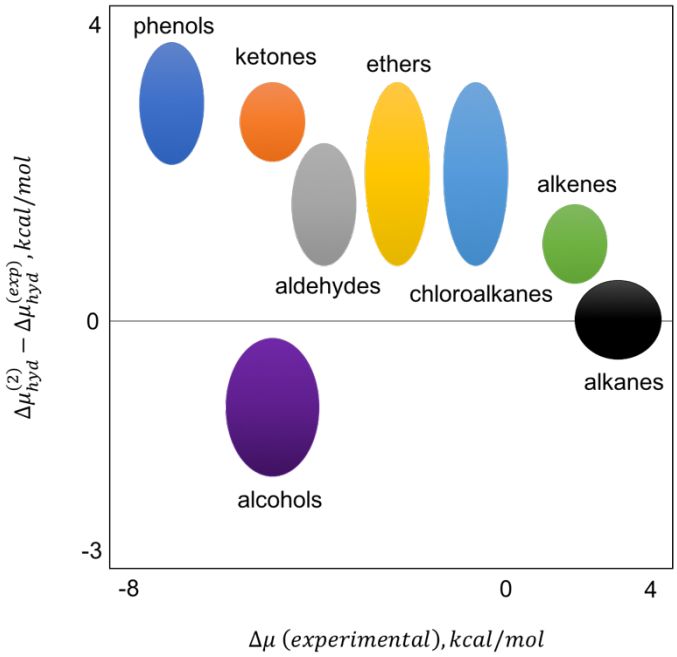


Figure 36 - A schematic of the results of the Ratkova *et al*[142] RISM with SDC correction, showing the grouping of compound classes (different coloured shapes), which Ratkova *et al*[142] aimed to correct with fragment corrections.

Fig. 36 shows a schematic of a graph from Ratkova *et al*[142], demonstrating the difference between the experimental HFE (x-axis) and the error term (y-axis) for the training set. Apart from the alkanes, all classes of compounds were found to be biased with respect to zero, but with a small standard deviation in error for compounds of the same class. With this observation, it was assumed that fragment corrections could be used to remove the error bias for each class of compounds. An additional observation is that systematic errors inherent in the RISM methodology are made apparent by the introduction of functional group information, explaining why HFE values of a certain functional group containing compound are over- or underestimated by RISM.

The conventional RISM scheme, with no additional corrections, only allows for qualitative descriptions of hydration, due to the number of approximations made. Specifically for 1D-RISM, errors in HFE have been identified as being caused by an overestimation of the cavitation energy required to place the solute into the solvent, and by an underestimation of the energy involved in hydrogen bond formation[143].

In addition to models which include specific structural corrections, such as the SDC model described above, a number of different corrections have been attempted in order to minimise the effect of the errors inherited by RISM due to its approximations. For example, a model designed to give quick estimations of HFE was developed by Palmer *et al*[144], combining 1D-RISM with molecular informatics, referred to as RISM-MOL-INF. RISM-MOL-INF firstly calculates the distribution function *g(r)* with 1D-RISM (with different closure relations), and then uses the value of *g(r)* at values of *r* as input descriptors to regression models. Partial least squares (PLS) and random forest estimators were considered for the regression problem. It was found that the four regression models, based upon different closure relations, predicted HFEs more accurately than the corresponding 1D-RISM models employing the same closure relations. It is possible that this improvement occurs because some of the approximations made in the final integral (e.g. the

exclusion of the bridge function) are removed by the use of a regression model. The molecules used to train the models are all organic drug-like molecules, therefore although they may be diverse under that definition, they are limited to a certain number of combinations of molecular functionality. It may be the case that the interactions that occur for this limited functionality within water are captured within the specific reaction coordinates accounted for in the final regression models, and that only their propensity (i.e. the descriptor value) is important, and excluded reaction coordinates are not often encountered as interaction distances.

The computational cost associated with improving methods also needs to be considered when adding corrections to 1D-RISM. For example, Ratkova[60] has shown that the SDC model can be further improved by introducing QM-derived partial charges into the initial RISM calculation scheme, rather than using OPLS based charges. However, this strategy significantly increases computational cost.

Another investigation focused on the improvement of HFE calculation with the RISM methodology was conducted by Freedman and Truong.[145] Their study focuses on the implementation of MD or MC distribution functions, based upon the observation that RDFs obtained from MD or MC simulations are more accurate than those obtained from RISM. For the usual RISM scheme, the description of the solvent is in the form of the bulk solvent susceptibility function, calculated from dielectrically consistent 1D-RISM[146]. The models investigated within Freedman and Truongs' work aim to use the distribution functions from MD or MC simulations within the RISM formalism to more accurately calculate HFEs. The results are compared to those found with the conventional RISM/HNC formalism. For a small set of organic molecules, good HFEs which were an improvement upon the original formalism were found.

Considering that further improvement of the RISM methodology by addition of information such as structural descriptors not only involves an increased cost for an increase of accuracy, but also involves modification of the RISM formalism, which already contains a number of inherent corrections, this chapter focuses upon the implementation of distribution functions not calculated within the RISM methodology, similarly to the Freedman and Truong study[145].

As the distribution functions (as described in the previous chapter) we have calculated are based upon an average distribution of water around various atom types with no consideration of orientation, a scheme is needed to account for this. The most obvious implementation of this is an atom weighting scheme that sufficiently describes each atom's contribution to the solvation energy. In this light, a property that depends upon the specific 3D structure of the solute molecule is also desirable. It has previously been shown that the solvent accessible surface area (SASA) is directly related to the cavitation energy, with molecules having larger SASAs being more insoluble[34]. Therefore, in this chapter, we aim to demonstrate the application of our RDFs, weighted by SASA to the calculation of HFE within the RISM formalism, as described theoretically below for the simplest possible case. We also consider a number of correction schemes.

5.2 Theory

5.2.1 Calculating the direct correlation function

Within the RISM formalism, the HNC closure relation (as described in chapter 2) describes the RDF between two particles α and γ as:

$$g_{\alpha\gamma}(r) = \exp\left(-\beta u_{\alpha\gamma}(r) + h_{\alpha\gamma}(r) - c_{\alpha\gamma}(r)\right) \quad [5.1]$$

Rearranging this equation to solve for the direct correlation function $c_{\alpha\gamma}(r)$ gives:

$$e^{c_{\alpha\gamma}(r)} = \frac{e^{-\beta u_{\alpha\gamma}(r)} e^{h_{\alpha\gamma}(r)}}{g_{\alpha\gamma}(r)}$$

$$c_{\alpha\gamma}(r) = -\beta u_{\alpha\gamma}(r) + h_{\alpha\gamma}(r) - \ln\left(g_{\alpha\gamma}(r)\right) \quad [5.2]$$

Expanding this expression, using the relationship $g_{\alpha\gamma}(r) = h_{\alpha\gamma}(r) + 1$ gives:

$$c_{\alpha\gamma}(r) = -\beta u_{\alpha\gamma}(r) + \left(g_{\alpha\gamma}(r) - 1\right) - \ln\left(g_{\alpha\gamma}(r)\right) \quad [5.3]$$

Using the potential of mean force, which is directly related to the RDF as $w^{(2)}(r) = -kT \ln g(r)$ (see chapter 4) as an estimate of the pair potential $u_{\alpha\gamma}$:

$$c_{\alpha\gamma}(r) = -\beta\left(-k_B T \ln g_{\alpha\gamma}(r)\right) + \left(g_{\alpha\gamma}(r) - 1\right) - \ln\ g_{\alpha\gamma}(r) \quad [5.4]$$

and substituting the expression for $\beta = 1/k_B T$:

$$c_{\alpha\gamma}(r) = -\left(\frac{1}{k_B T}\right)\left(-k_B T \ln g_{\alpha\gamma}(r)\right) + \left(g_{\alpha\gamma}(r) - 1\right) - \ln\ g_{\alpha\gamma}(r)$$

$$c_{\alpha\gamma}(r) = g_{\alpha\gamma}(r) - 1 = h_{\alpha\gamma}(r) \quad [5.5]$$

In model equation 5.5, the simplification for the expression for $c_{\alpha\gamma}(r)$ results in a relationship whereby $c_{\alpha\gamma}(r)$ and $h_{\alpha\gamma}(r)$ are equal. By definition, $c_{\alpha\gamma}(r)$ remains finite in the volume integral, and so does not become long-ranged (i.e. it describes direct correlations), whereas $h_{\alpha\gamma}(r)$ describes many body correlations, and therefore is not finite in its volume integral, and includes information about long-range correlation. The difference between $c_{\alpha\gamma}(r)$ and $h_{\alpha\gamma}(r)$ can be expressed as an expansion of graphs (from which the original functions are an infinite sum of), where the graphs have well defined topological features, as expressed in equation 5.2. Although addition of the PMF to describe $u_{\alpha\gamma}(r)$ greatly simplifies the expression for finding $c_{\alpha\gamma}(r)$, in this work, $c_{\alpha\gamma}(r)$ is explicitly calculated from equation 5.2.

5.2.2 HFE expressions

HNC. The HNC[67] free energy expression relates $c_{\alpha\gamma}(r)$ and $h_{\alpha\gamma}(r)$, as calculated from equation 5.1 to the free energy as:

$$\Delta G_{HNC} = 2\pi\rho kT \sum_{\alpha\gamma} \int_0^\infty \left[-2c_{\alpha\gamma}(r) - h_{\alpha\gamma}(r)\left(c_{\alpha\gamma}(r) - h_{\alpha\gamma}(r)\right)\right] r^2 dr \quad [5.6]$$

Repulsive bridge correction and HNCB HFE expression. The repulsive bridge correction, developed and applied to the HNC closure by Kovalenko and Hirata,[147] treats the overestimation of water ordering around hydrophobic solutes in the HNC RISM formalism. More specifically, it treats the entropic component of HFE. The repulsive bridge correction ($-B^R_{\alpha\gamma}(r)$) is calculated as:

$$\exp\left(-B^R_{\alpha\gamma}(r)\right) = \prod_{\nu\neq\gamma} \left\langle w^{bulk}_{\gamma\nu} \times \exp\left(-\beta\varepsilon_{\alpha\nu}\left(\frac{\sigma_{\alpha\nu}}{r}\right)^{12}\right)\right\rangle \quad [5.7]$$

where $w^{bulk}_{\gamma\nu}$ is the bulk solvent intramolecular correlation function, and $\varepsilon_{\alpha\nu}$ and $\sigma_{\alpha\nu}$ are Lennard-Jones parameters. The HNCB (HNC with repulsive bridge correction) expression for free energy is given by:

$$\Delta G_{HNCB} = \Delta G_{HNC} + 4\pi\rho kT \sum_{\alpha\gamma} \int_0^\infty \left(h_{\alpha\gamma}(r) + 1\right)\left(e^{-B^R_{\alpha\gamma}(r)} - 1\right) r^2 dr \quad [5.8]$$

Gaussian Fluctuations approximation (GF). The GF[148,149] free energy expression assumes Gaussian fluctuations of the solvent. The closure relation from which $c_{\alpha\gamma}(r)$ and $h_{\alpha\gamma}(r)$ are calculated is not specific to the GF method (i.e. any closure relation can be used). GF gives the free energy as:

$$\Delta G_{GF} = 2\pi\rho kT \sum_{\alpha\gamma} \int_0^\infty \left(-2c_{\alpha\gamma}(r) - c_{\alpha\gamma}(r)h_{\alpha\gamma}(r)\right) r^2 dr \quad [5.9]$$

5.2.3 Relation of the partial molar volume (PMV) to g(r)

The Kirkwood-Buff (KB) solution theory provides a framework for evaluating thermodynamic quantities of a liquid mixture in terms of $g_{\alpha\gamma}(r)$. In this theory, the PMV ($\bar{V}$) is given by:

$$\bar{V} = kT\chi_T - \int_0^\infty \left(g_{\alpha\gamma}(r) - 1\right) 4\pi r^2 dr \quad [5.10]$$

where χ_T is the isothermal compressibility of solution. The PMV is easily obtained if $g_{\alpha\gamma}(r)$ has been calculated.

5.3 Methods

5.3.1 Dataset compilation

The experimental HFEs for a dataset of 70 structures were taken from the FreeSolv[150] database, which sources the existing experimental literature (experimental measurement of HFEs can be facilitated by methods such as calorimetry). The 70 structures each have a known crystal structure in the CSD (structures with Z'>1 were omitted). The solute geometry was taken as the asymmetric unit of the corresponding crystal structure, with a local geometry minimisation performed using the CSD python API[151]. This minimisation uses the TRIPOS forcefield[152], but also uses known valence bond lengths and angles based upon distributions found within the CSD. The RDF, volume and density terms were then calculated as outlined below. The final integration and implementation of calculation schemes, described in the narratives in section 5.4, were performed using a collection of simple python scripts, described briefly in section 5.3.4, which are available in Electronic Appendix III.

5.3.2 Solute RDF calculation

The solute molecule is initially atom-typed with antechamber, using AMBER types, as was done during the calculation of the RDFs described in chapter 4[128]. Next, the atomic contributions to the solvent-accessible surface area (SASA) are calculated with the Lee and Richards[153] method, via freeSASA[154]. The solute molecule is then treated as a single interaction site. The atomic contributions for SASA are used to weight the empirically calculated RDFs (from organic hydrates) described in chapter 4, and the sum of SASA weighted RDFs for each solute molecule atom is normalised by the total molecule SASA, giving two RDFs; $g_{mol}^{OW}(r)$ and $g_{mol}^{HW}(r)$. These two RDFs are smoothed using a Savitzky-Golay algorithm[138]. The calculation of solute RDF was attempted both with and without the inclusion of hydrogen atoms. This is because it is likely that the information contained about hydrogen atom type RDFs will also be contained in the RDFs of the atom types to which they are attached. This is discussed further in the results section (5.4). This method is depicted in Fig. 37.

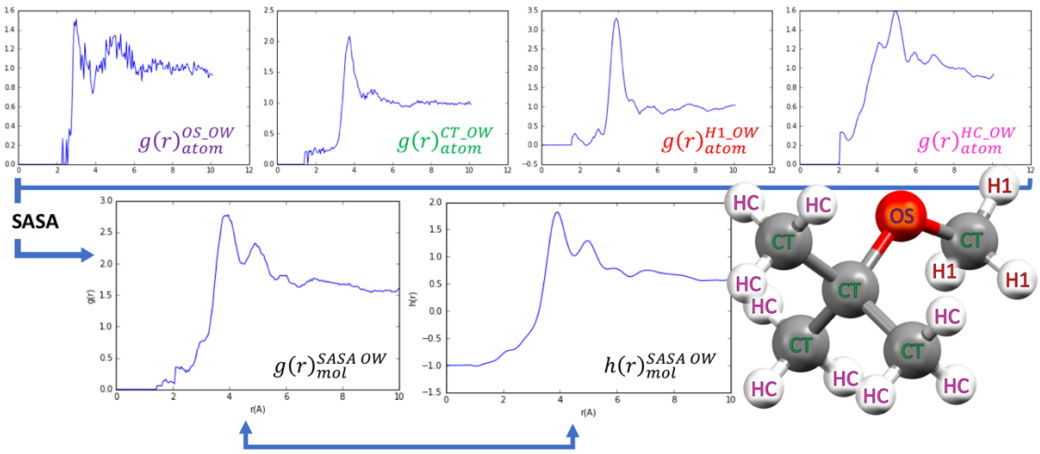


Figure 37 - The SASA atomic contributions are calculated for each atom in the solute molecule, along with each atom AMBER type (see molecule, right). From this, the corresponding empirically calculated RDFs (top) can be weighted by SASA per atom, and used along with SASA weighting to produce molecular RDFs (bottom left), and finally $\boldsymbol{h_{\alpha\gamma}(r)}$ (with a smoothing algorithm).

5.3.3 Energy expressions

The HFEs in this work were calculated using a custom set of python scripts, including the solute RDF calculations described in 5.3.2. The algorithms used for the final energy terms were written to follow the algorithmic methodology of RISM-MOL[155], in order to allow for comparison of features between our own methods and 1D-RISM, upon which they are based.

5.3.4 Regression methods: descriptors vs. calculated terms

In chapter 3, it was shown that an ExtraTrees regressor works well for feature reduction and selection before a regression problem is attempted. Considering this, the importance of molecular descriptors in relation to HFE was considered. The importance of descriptors was also then re-evaluated with the inclusion of the developed PMV and HFE terms (but not the corrected terms), as described in the narrative in sections 5.4.1, 5.4.2 and 5.4.3.

The available data were then split into training and test sets containing the same molecular structures, and the ExtraTrees regressor was used to select the 15 most important descriptors from the training set. Finally, a variety of Lasso (Lasso, LassoLarsCV and LassoLarsIC) estimators were used to regress: (a) the experimental HFE with molecular descriptors only, and (b) the experimental HFE with molecular descriptors, and the developed HFE and PMV terms.

5.4 Results & discussion

5.4.1 HNC HFE expression

Initially, the HFE energy term was calculated with the hydrogen atom types accounted for (ΔG^{H}_{HNC}), and with them removed (ΔG_{HNC}). It was found that the correlation with experimental HFEs was better for ΔG_{HNC}(with hydrogen RMSE = 113.63 kcal/mol, r^2 = 0.17; without hydrogen discussed below). However, ΔG_{HNC} gave HFEs that were too negative, by approximately an order of magnitude. This is shown in the top left graph in Fig. 38. A line of best fit through the calculated energies gave an expression:

$$\Delta G_{HNC} = -2.45\ \Delta G_{experimental} - 58.34 \qquad [5.11]$$

with an R^2 = 0.23. With respect to a line fitted where $\Delta G_{HNC} = \Delta G_{experimental}$ the r^2 = 0.47, and RMSE = 66.39 kcal/mol. It is known that the PMV correlates well to the errors obtained within the conventional RISM formalism. In order to deduce whether the PMV correlates to the error for our own model, we plotted the error term from ΔG_{HNC} against the expression obtained for the PMV, calculated with equation 5.10. The PMV term was calculated both from $g_{\alpha\gamma}(r)$ with and without the hydrogen atoms included. It was found that there was not a good correlation of the ΔG_{HNC} error with the PMV calculated from $g_{\alpha\gamma}(r)$ with hydrogen atoms removed. The correlation of the ΔG_{HNC} error with PMV from $g_{\alpha\gamma}(r)$ including hydrogen atoms was extremely high (R^2 = 0.96). This is shown in the top right graph in figure 38. The linear equation for the best fit of correlation was used to add a PMV correction term to ΔG_{HNC}, as shown in the middle left graph in figure 38. This correction gives the expression:

$$\Delta G_{HNC}^{PMV} = \Delta G_{HNC} - 0.0001\,\bar{V} + 72.87 \quad [5.12]$$

Although the calculated HFEs obtained with ΔG_{HNC}^{PMV} are no longer incorrect by an order of magnitude (reducing the RMSE), the results obtained are actually less correlated to $\Delta G_{experimental}$ (best fit R^2 = 0.21; $\Delta G_{HNC}^{PMV} = \Delta G_{experimental}$ r^2 = 0.45, RMSE = 6.83 kcal/mol). The equation of best fit through the results give the expression:

$$\Delta G_{HNC}^{PMV} = 0.91\,\Delta G_{experimental} + 2.36 \quad [5.13]$$

It is noteworthy that the PMV correction applied to ΔG_{HNC} included information about the hydrogen atoms that were initially removed from the calculation. This may suggest that although their removal gives a better correlation to $\Delta G_{experimental}$, it is necessary to represent them in some other way, as they are essential to the error term. In order to further probe this, the error term for ΔG_{HNC}^{PMV} was plotted against ΔG_{HNC}^{H} to identify any correlation, as shown in the middle right graph in Fig. 37. It was found that there was a correlation of R^2 = 0.41, with the expression:

$$\Delta G_{HNC}^{PMV}\ error = -0.14\,\Delta G_{HNC}^{H} - 19.03 \quad [5.14]$$

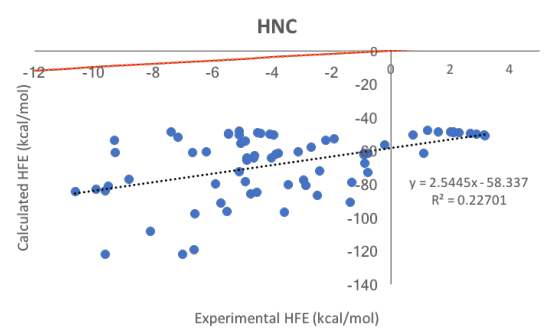
This error correction term was applied to ΔG_{HNC}^{PMV} in the same fashion as the PMV correction in equation 5.12, to give the $\Delta G_{HNC\ H}^{PMV}$ model shown in the bottom graph of Fig. 38. The addition of the hydrogen correction once again improved the RMSE, making it 4.83 with respect to $\Delta G_{HNC\ H}^{PMV} = \Delta G_{experimental}$ with r^2 = 0.47, which is almost identical to the ΔG_{HNC} model. This demonstrates that although the error from ΔG_{HNC} correlated well with PMV, the order of magnitude of the error means it is very difficult to accurately correct it. In addition to this, if the error is not corrected for such a strong correlation, there may be issues inherent in the methodology; either due to the atom-type RDFs in their raw form, or due to the subsequent weighting and normalisation scheme applied to them. Another consideration is that the HNC method may not be theoretically robust enough to give accurate predictions. Indeed, it has already been shown that the HNC method can be improved by a repulsive bridge correction (HNCB), shown in equations 5.8 and 5.9.

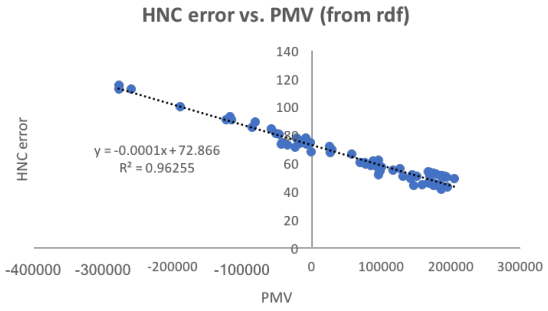
5.4.2 HNCB HFE expression

In order to establish whether the correction methodology employed for the HNC HFE expression would be more valid given a more robust methodology, the work described in section 5.4.1 was repeated for HFEs calculated with the HNCB model (ΔG_{HNCB}), as shown in Fig. 39.

As found for the ΔG_{HNC} method, the HFEs initially calculated by ΔG_{HNCB} are incorrect by an order of magnitude. In addition to this, the range of energies calculated is also much broader. However, ΔG_{HNCB} gives a better correlation to $\Delta G_{experimental}$ than ΔG_{HNC} (best fit R^2 = 0.27; $\Delta G_{HNCB} = \Delta G_{experimental}$ r^2 = 0.51, RMSE = 94.25 kcal/mol). The equation for the linear best fit of ΔG_{HNCB} to $\Delta G_{experimental}$ is:

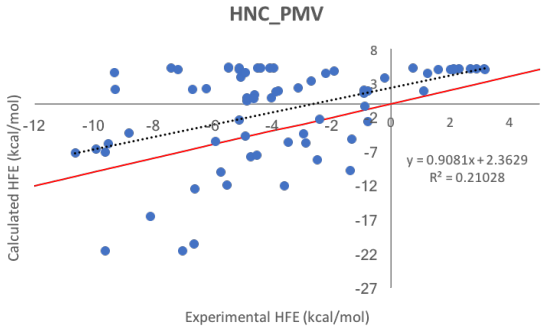
$$\Delta G_{HNCB} = -14.84\,\Delta G_{experimental} + 76.29 \quad [5.15]$$

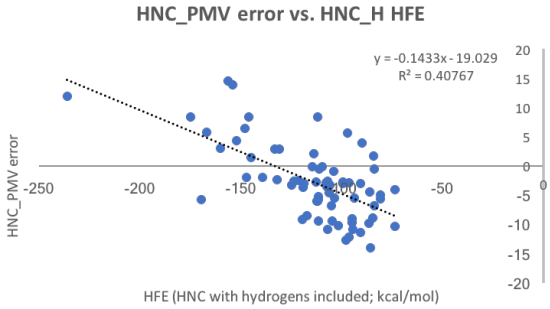




$$\Delta G_{HNC} = 2\pi\rho kT \sum_{\alpha\gamma} \int_0^\infty \left[-2c_{\alpha\gamma}(r) - h_{\alpha\gamma}(r)\left(c_{\alpha\gamma}(r) - h_{\alpha\gamma}(r)\right)\right] r^2 dr$$

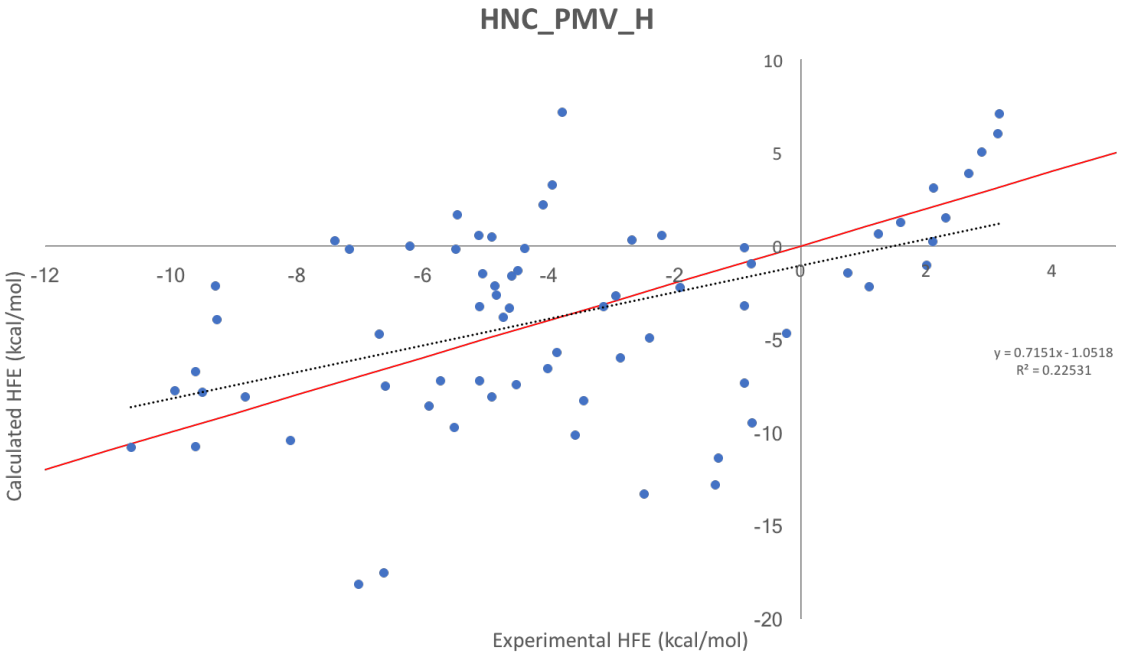
$$\bar{V} = kT\chi_T - \int_0^\infty \left(g_{\alpha\gamma}(r) - 1\right) 4\pi r^2 dr$$





$$\Delta G_{HNC}^{PMV} = \Delta G_{HNC} + (-0.0001 \times \bar{V} + 72.886)$$

$$\Delta G_{HNC}^{H}$$



$$\Delta G_{HNC\,H}^{PMV} = \Delta G_{HNC}^{PMV} + (-0.1433 \times \Delta G_{HNC}^{H} - 19.029)$$

Figure 38 - The HFEs calculated with the HNC free energy expression are correlated to the PMV. Applying the expression for this correlation to the HNC energy significantly improves the correlation between experimental and calculated HFES. Finally, a weak correlation between the error of the new energy term with the HFE calculated with hydrogens included is found. Applying this correlation as an additional correction also improves the energy error between the experimental and calculated HFEs.

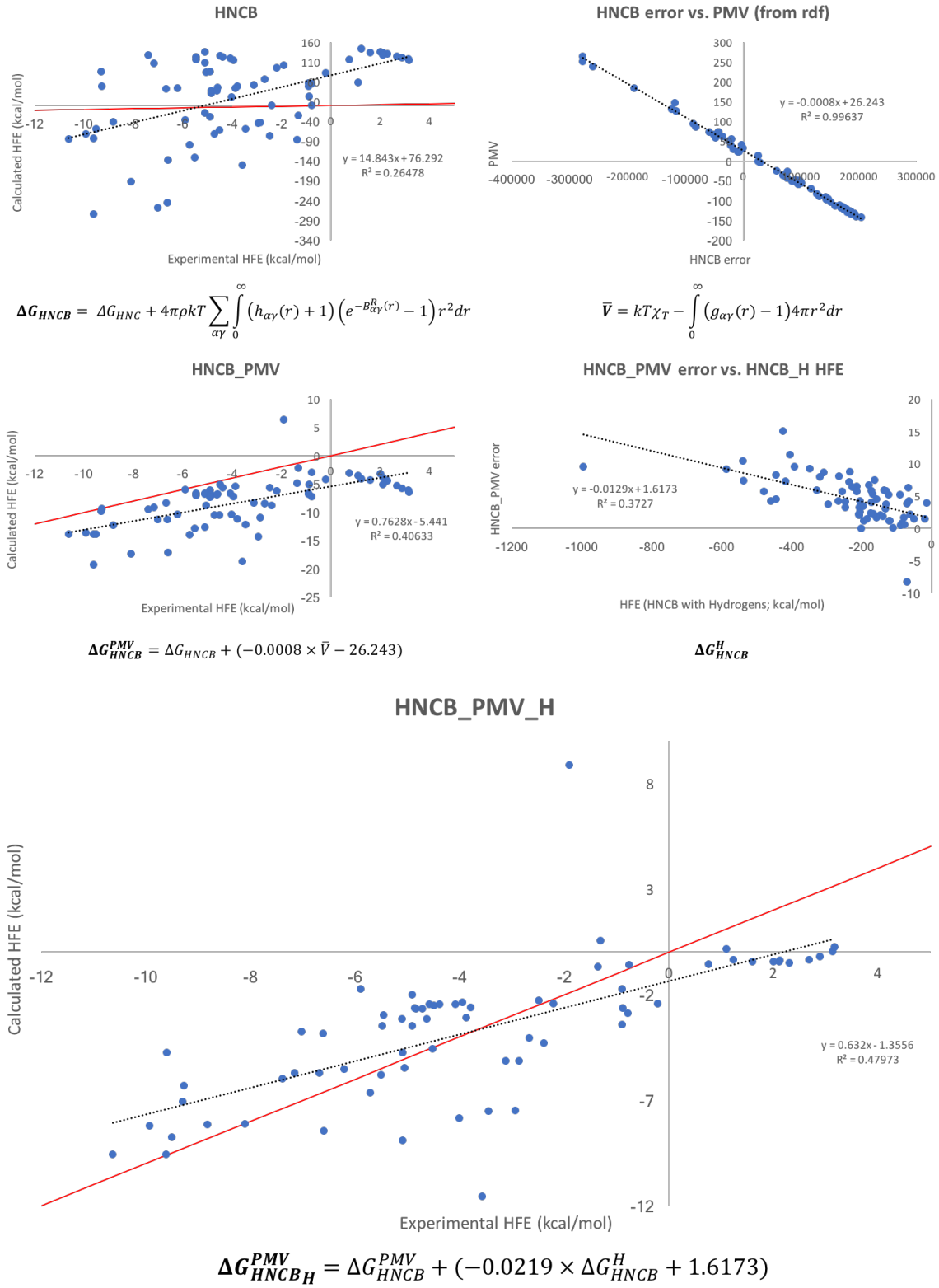


Figure 39 - The HFEs calculated with the HNCB free energy expression are correlated to the PMV. Applying the expression for this correlation to the HNCB energy significantly improves the correlation between experimental and calculated HFES. Finally, a weak correlation between the error of the new energy term with the HFE calculated with hydrogens included is found. Applying this correlation as an additional correction also improves the energy error between the experimental and calculated HFEs.

The correlation of the ΔG_{HNCB} error to the PMV (with hydrogens included) was also extremely strong, and a better correlation was found for ΔG_{HNCB} than for ΔG_{HNC}, with R^2 = 0.96. The equation found relating the error to the PMV was used to correct ΔG_{HNCB}, giving:

$$\Delta G_{HNCB}^{PMV} = \Delta G_{HNCB} - 0.0008\,\bar{V} + 26.24 \quad [5.16]$$

ΔG_{HNCB}^{PMV} is shown in the middle left graph in Fig. 39. The PMV correction significantly improves the range at which the calculated energies occur. In addition to this, the correction improves the fit of the calculated energies to the experimental values (best fit R^2 = 0.41; $\Delta G_{HNCB}^{PMV} = \Delta G_{experimental}$ r^2 = 0.63, RMSE = 5.69 kcal/mol), unlike ΔG_{HNC}^{PMV} where the PMV correction resulted in a worse correlation.

Finally, the correlation between ΔG_{HNCB}^{PMV} error and ΔG_{HNCB}^{H} was explored. This is shown in the middle right graph in Fig. 39. As with the PMV correction, ΔG_{HNCB}^{H} correlated better to the ΔG_{HNCB}^{PMV} error than the equivalent HNC correction (best fit R^2 = 0.37), and the following expression gives the appropriate ΔG_{HNCB}^{H} correction:

$$\Delta G_{HNCB_H}^{PMV} = \Delta G_{HNCB}^{PMV} - 0.022\,\Delta G_{HNCB}^{H} + 1.62 \quad [5.17]$$

The model shown in equation 5.17 is shown in the bottom graph in Fig. 39. This model gave the best correlation of all the models tested, derived from the 1D-RISM formalisms described in section 5.2.2 (best fit R^2 = 0.48; $\Delta G_{HNCB_H}^{PMV} = \Delta G_{experimental}$ r^2 = 0.68, RMSE = 2.69 kcal/mol).

5.4.3 GF HFE expression

ΔG_{GF} gave the poorest results of all the energy expressions tested in this work (best fit R^2 = 0.19; $\Delta G_{GF} = \Delta G_{experimental}$ r^2 = 0.43, RMSE = 73.15 kcal/mol). The correlation of the error of ΔG_{GF} with the PMV was better fitted to an exponential expression (R^2 = 0.91), unlike the linear correlations found in sections 5.4.1 and 5.4.2. The expression for this correlation was:

$$\Delta G_{GF}^{PMV} = 76.615\, e^{-2\times10^{-6}\bar{V}} \quad [5.18]$$

The application of this correction significantly decreases the quality of calculated HFEs (best fit R^2 = 0.0088; $\Delta G_{GF} = \Delta G_{experimental}$ r^2 = 0.09, RMSE = 4.64 kcal/mol), thus no further correction schemes were applied to the GF HFE expression, as the results no longer show any correlation.

5.4.4 Regression methods: descriptors vs. calculated terms

Although our calculated HFEs ($\Delta G_{HNC}, \Delta G_{HNC}^{H}, \Delta G_{HNCB}, \Delta G_{HNCB}^{H}, \Delta G_{GF}$ and ΔG_{GF}^{H}) are only weakly correlated to the experimental HFE (see table 7, below), it is possible that they may be significantly improved by molecular descriptors. The reason for low correlations may be due to some information that is lost by using the RDF construction method described in 5.3.2, which may be recovered by the inclusion of molecular descriptors. In order to test this, the method described in section 5.3.5 was followed.

Table 7 - Summary of r^2 and RMSE values for the models discussed in the narratives in sections 5.4.1, 5.4.2 and 5.4.3

	Experimental HFE		
Model	**r^2**	**MAE (kcal/mol)**	**RMSE (kcal/mol)**
ΔG^{H}_{HNC}	0.17	110.24	113.63
ΔG_{HNC}	0.47	64.02	66.39
ΔG^{PMV}_{HNC}	0.45	2.70	6.83
$\Delta G^{PMV}_{HNC\ H}$	0.47	0.00	4.83
ΔG^{H}_{HNCB}	0.22	224.94	275.84
ΔG_{HNCB}	0.51	25.39	94.25
ΔG^{PMV}_{HNCB}	0.63	4.57	5.69
$\Delta G^{PMV}_{HNCB\ H}$	0.68	0.00	2.69
ΔG^{H}_{GF}	0.23	140.21	160.92
ΔG_{GF}	0.43	70.45	73.15
ΔG^{PMV}_{GF}	0.09	0.55	4.64

The statistical measures corresponding to the regression models calculated by Lasso, LassoLarsCV and LassoLarsIC estimators are summarised in table 8, and a visualisation of the experimental vs. predicted HFEs for each method is given in Fig. 40, for the following regression equations;

Lasso – Molecular descriptors only:

$$\Delta G = -1.86 - 0.10\,TPSA - 0.032\,SlogP_{VSA11} + 0.063\,SlogP_{VSA5} - 0.11\,SMR_{VSA9} - 0.058\,SlogP_{VSA2} - 0.20\,PEOE_{VSA10}$$

[5.19]

Lasso – Molecular descriptors and calculated HFE and PMV terms:

$$\Delta G = -1.78 - 0.10\,TPSA + 0.0074\,\Delta G_{HNCB} + 0.036\,SlogP_{VSA5} + 0.0039\,SMR_{VSA5} - 0.11\,PEOE_{VSA1} - 0.04\,SlogP_{VSA2}$$

[5.20]

LassoLarsCV – Molecular descriptors only:

$$\Delta G = -1.26 - 0.019\,TPSA - 0.57\,NumHAcceptors + 4.62\,MinPartialCharge + 0.0081\,SlogP_{VSA5} - 0.96\,NumHDonors + 0.091\,FractionCSP3 - 0.050\,PEOE_{VSA10}$$

[5.21]

LassoLarsCV – Molecular descriptors and calculated HFE and PMV terms:

$$\Delta G = -1.19 - 0.012\,TPSA + 3.83\,MinPartialCharge + 0.0030\,\Delta G_{HNCB} + 0.0049\,SlogP_{VSA5} - 0.91\,NumHAcceptors + 0.012\,SMR_{VSA5} - 1.51\,NumHDonors$$

[5.22]

LassoLarsIC – Molecular descriptors only:

$$\Delta G = 0.40 - 1.22\ NumHAcceptors + 2.52\ MinPartialCharge + 0.39\ SlogP_{VSA11} - 0.31\ SMR_{VSA5} + 0.38\ SlogP_{VSA5} - 2.07\ NumHDonors - 0.0040\ Kappa3 - 0.13\ SlogP_{VSA2} - 0.028\ Estate_{VSA8} - 0.75\ PEOE_{VSA10} - 0.50\ Chi1 + 1.37\ fr_{methoxy} + 0.14\ fr_{ether} + 0.25\ MolLogP + 0.22\ SMR_{VSA1}$$

[5.23]

LassoLarsIC – Molecular descriptors and calculated HFE and PMV terms:

$$\Delta G = -5.11 - 0.18\ SlogP_{VSA11} + 2.64\ MinPartialCharge + 0.014\ \Delta G_{HNCB} + 0.67\ fr_{ether} - 0.64\ NumHAcceptors - 0.041\ Kappa3 - 0.071\ SlogP_{VSA2} - 0.055\ \Delta G_{GF} - 2.73\ NumHDonors - 0.15\ PEOE_{VSA8}$$

[5.24]

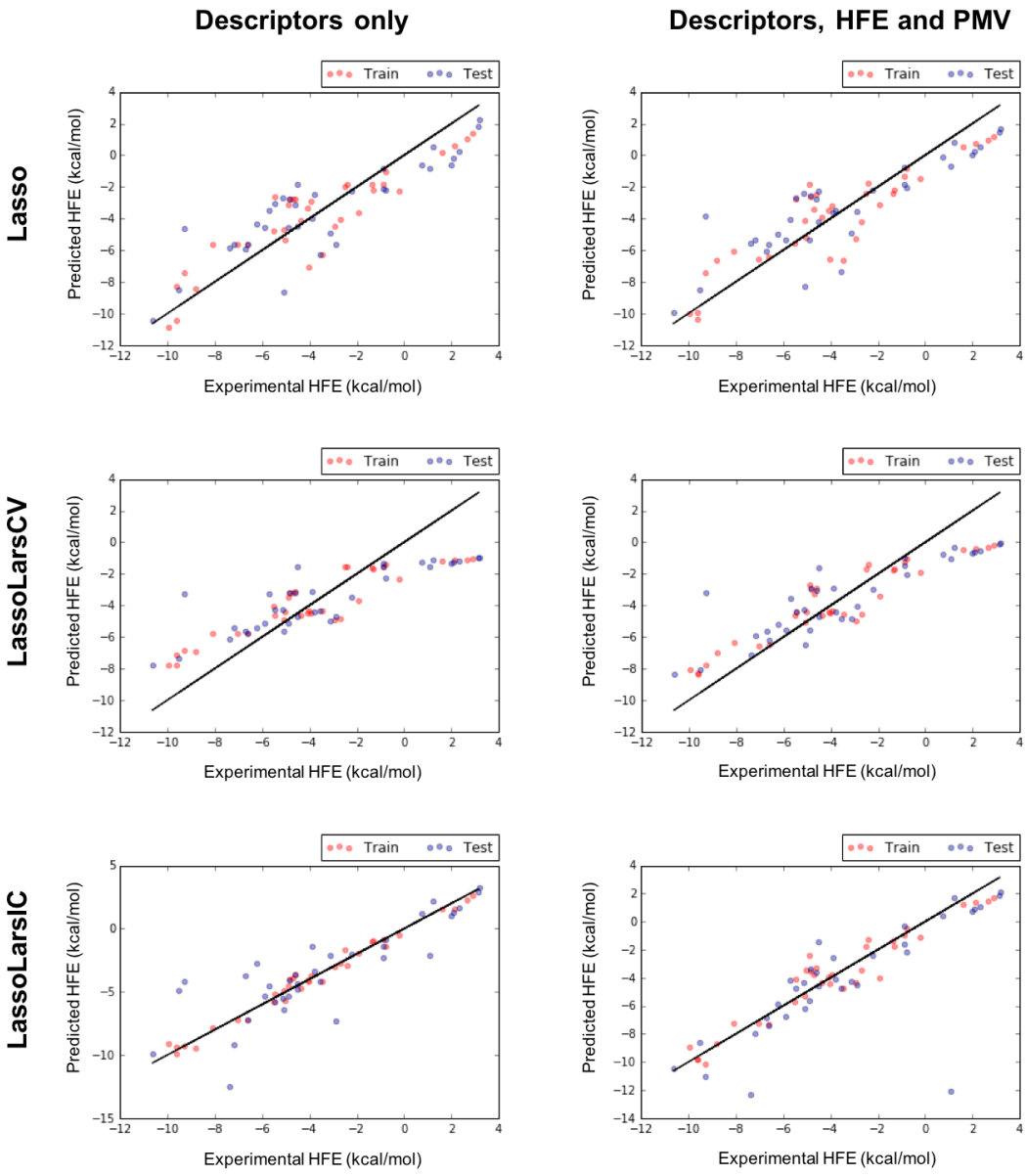


Figure 40 - Various Lasso regression models for the HFE, showing the comparison of prediction for the same training (red) and test (blue) sets when; (a) only molecular descriptors are included, and (b) when the calculated HFEs and PMV terms are included as descriptors.

Table 8 - A summary of the statistical measures calculated for the various Lasso estimator regression models for: (a) molecular descriptors only, and (b) molecular descriptors, and calculated HFE and PMV terms.

			Lasso	*LassoLarsCV*	*LassoLarsIC*
Molecular descriptors only	**Train**	**MAE (kcal/mol)**	1.32	1.49	0.38
		MSE (kcal/mol)	2.32	3.19	0.22
		R^2	0.80	0.71	0.98
	Test	**MAE (kcal/mol)**	1.59	1.83	1.41
		MSE (kcal/mol)	3.59	5.03	4.21
		R^2	0.73	0.63	0.69
Molecular descriptors, and calculated HFE and PMV terms	**Train**	**MAE (kcal/mol)**	1.24	1.25	0.77
		MSE (kcal/mol)	2.35	2.22	0.95
		R^2	0.8	0.81	0.92
	Test	**MAE (kcal/mol)**	1.5	1.58	1.42
		MSE (kcal/mol)	3.49	3.86	6.83
		R^2	0.74	0.71	0.49

For both the case where the HFE is predicted from only molecular descriptors, and its equivalent model with calculated HFE and PMV descriptors included, the best model (as defined by the best test R^2) is obtained from a Lasso estimator. These two models also have the fewest descriptors. However, as seen in figure 40 (Lasso, descriptors only) there are clusters of structures which are biased in their error. These clusters roughly correspond to an area between 2 kcal/mol < $\Delta G_{experimental}$ < 4 kcal/mol, which covers a number of alkanes of varying chain length; and a second cluster between -6 kcal/mol < $\Delta G_{experimental}$ < -3 kcal/mol covering a variety of phenol derivatives. This bias is also present for the equivalent Lasso model including the ΔG_{HCNB} term (Fig. 40; Lasso, descriptors, HFE and PMV). However, the relationship between these biased clusters and the black $x = y$ line plotted appears to be more correlated, where the clusters appear to be transformed to more linear shapes, and are parallel to the $x = y$ line. This a similar observation to the structure-based bias found by Ratkova *et al*[142] (Fig. 36).

Although the Lasso models described above are statistically the best, a visual comparison of the plots in Fig. 40 shows that for the models with the calculated HFE and PMV included, the LassoLarsIC estimator gives a model which has a better fit and less structure-specific bias in the error; although there are two large outliers in the test set that affect the statistical measures. It may be the case that if the dataset were increased in size, the number of outliers would also increase. As the number of samples in the dataset is quite small, this is not known. However, an analysis of the importances of the descriptors for the whole dataset (rather than fitting on half of the data for a test set) may give more insight as to what is important for either predicting HFE purely from molecular descriptors; or to suggest what information may be missing from the models described in sections 5.4.1, 5.4.2, and 5.4.3. Additionally, this analysis may give more insight as to what is modelled well. In order to perform this analysis, the method for feature importance determination described in section 5.3.4 was used.

Fig. 41 shows the ExtraTrees feature importance of the descriptors (not including calculated HFE and PMV) for both the descriptor set for all structures with only molecular descriptors and the descriptor set for all structures with the HFE and PMV terms included in the importance fitting.

The most important feature found for both cases is the *NOCount* descriptor, which is a simple count of the number of nitrogen and oxygen atoms in the structure. This suggests that the HFE of structures containing nitrogen and oxygen atoms is particularly well defined by our models. This correlates to the findings that descriptors relating to polar terms (see below) are also important. Furthermore, the importance of this descriptor increases when the HFE and PMV terms are included. This suggests that either; the atoms are under-represented or not well described by the atom type descriptions in the RDFs used to calculate the solute RDFs (see section 5.3.2), or that their interactions with water in solution are very different to their interaction with water in hydrates.

Following this, for the inclusion of HFE and PMV terms, TPSA is the next most important feature. This feature is also important for models containing descriptors only. The TPSA descriptor corresponds to the Topological Polar Surface Area. Its appearance as an important feature for the descriptor only model is hardly surprising, as it stands to reason that molecules with a larger surface area corresponding to polar atoms will be more soluble than those with a smaller polar surface area. It is also not unreasonable that the term appears to be important for the prediction of HFE when our calculated terms are not included. This is because it is known, as previously described, that RISM type models significantly underestimate HFEs for polar molecules. However, the reason for deriving new RISM-type methods in the fashion we have was in order to add more information about solute-solvent interactions, and it was hoped that this would include more information about these specific underestimations. It is therefore slightly surprising that the TPSA descriptor is more important when our calculated terms are included. However, the importances are calculated from a random forest fit, so the relationship between descriptors, and their importances, may be more sophisticated than this simple observation and resulting explanation.

Two other important descriptors for both cases are *MaxAbsPartialCharge* and *MinPartialCharge*. This may also relate to the polarity of the molecule, corroborating the importance of the TPSA of the molecule. Furthermore, these kinds of descriptor are almost certainly correlated to the strength of the electrostatic solute-solvent interaction. The electrostatic contribution to HFE corresponds to long-range electrostatic interactions, in the form of an electrostatic response of a solvent to the solute charge. RISM with self-consistent field (RISM-SCF) methods determine electronic structure and solvent distribution consistently, with the solute atomic partial charges determined in the SCF step, and plugged in to the RISM calculation. This process is repeated until consistent results are obtained, meaning the electronic structure of the solute and the solvent distribution are simultaneously optimised[156]. In our methods, no information about the partial charges of the atoms is included, and no optimisation of either the solute or solvent is conducted. An inclusion of a different $u_{\alpha\gamma}(r)$ term, calculated with the partial charges of the solute molecule and Lennard-Jones parameters (as in RISM-MOL), rather than from the SASA weighted PMF, may alleviate some of the issues related to partial charge found within our models. These errors are almost certainly related to an inadequate treatment of the solute in our models.

Fig. 42 shows the feature importances for the top 50 features of the determination when the calculated HFE and PMV terms are included. As expected from the results of the RISM-type HFE methods, ΔG_{HNCB} was determined as the most important of the terms included (none of the PMV or H corrected terms were included), as this energy expression had the best correlation to experimental HFEs. More surprisingly, ΔG_{GF} was the next most important of the included energy terms. This term had the poorest correlation with the corresponding experimental HFEs. However, there may be some information from the assumption of Gaussian fluctuations of the solvent that is missed by the other energy terms.

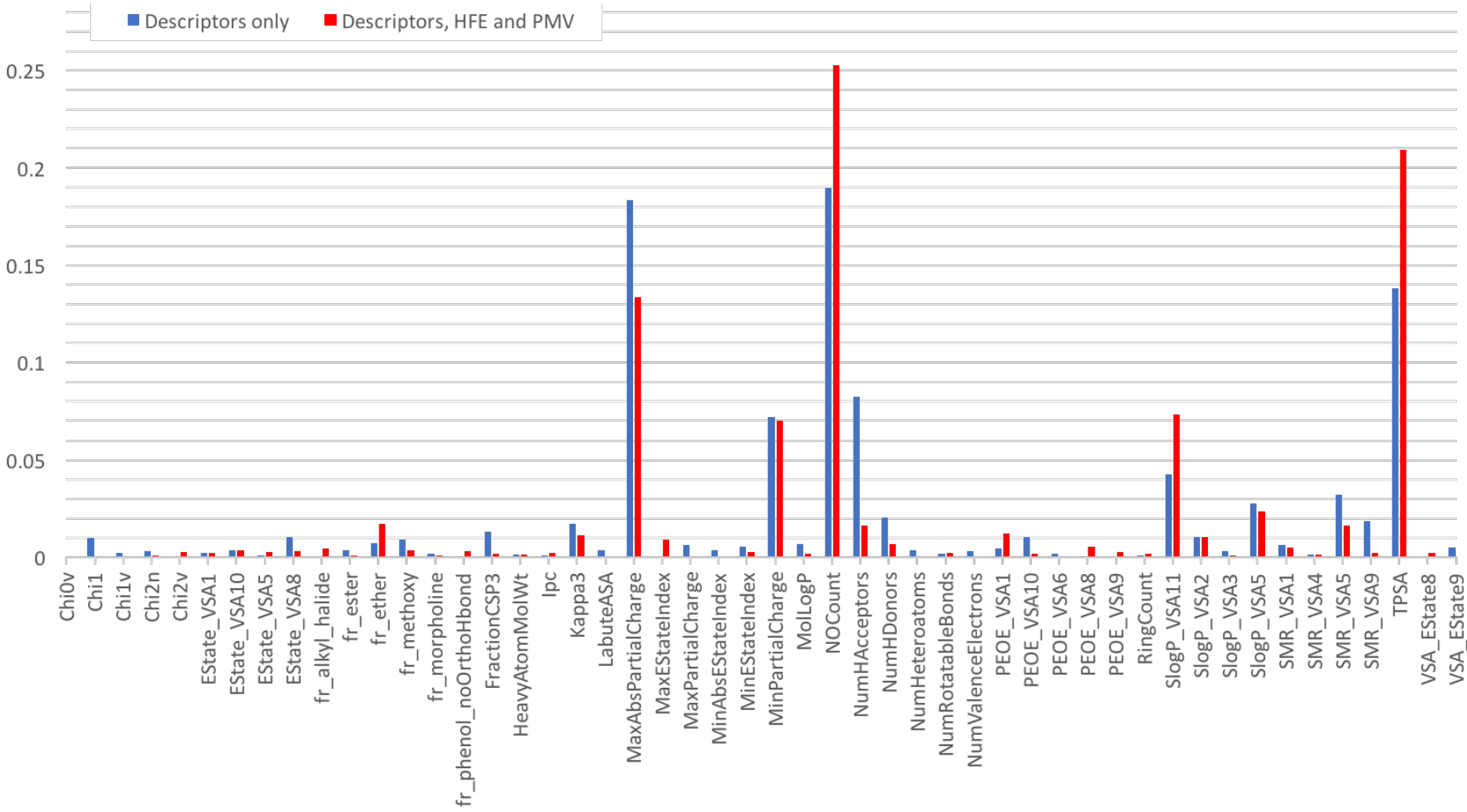


Figure 41 – Feature importances from an ExtraTrees regression of the top 50 (from sum) molecular descriptors for the prediction of HFE when; (a) only molecular descriptors are included (blue), and (b) when the calculated HFEs (with no corrections) and PMV (with no corrections) are included (red).

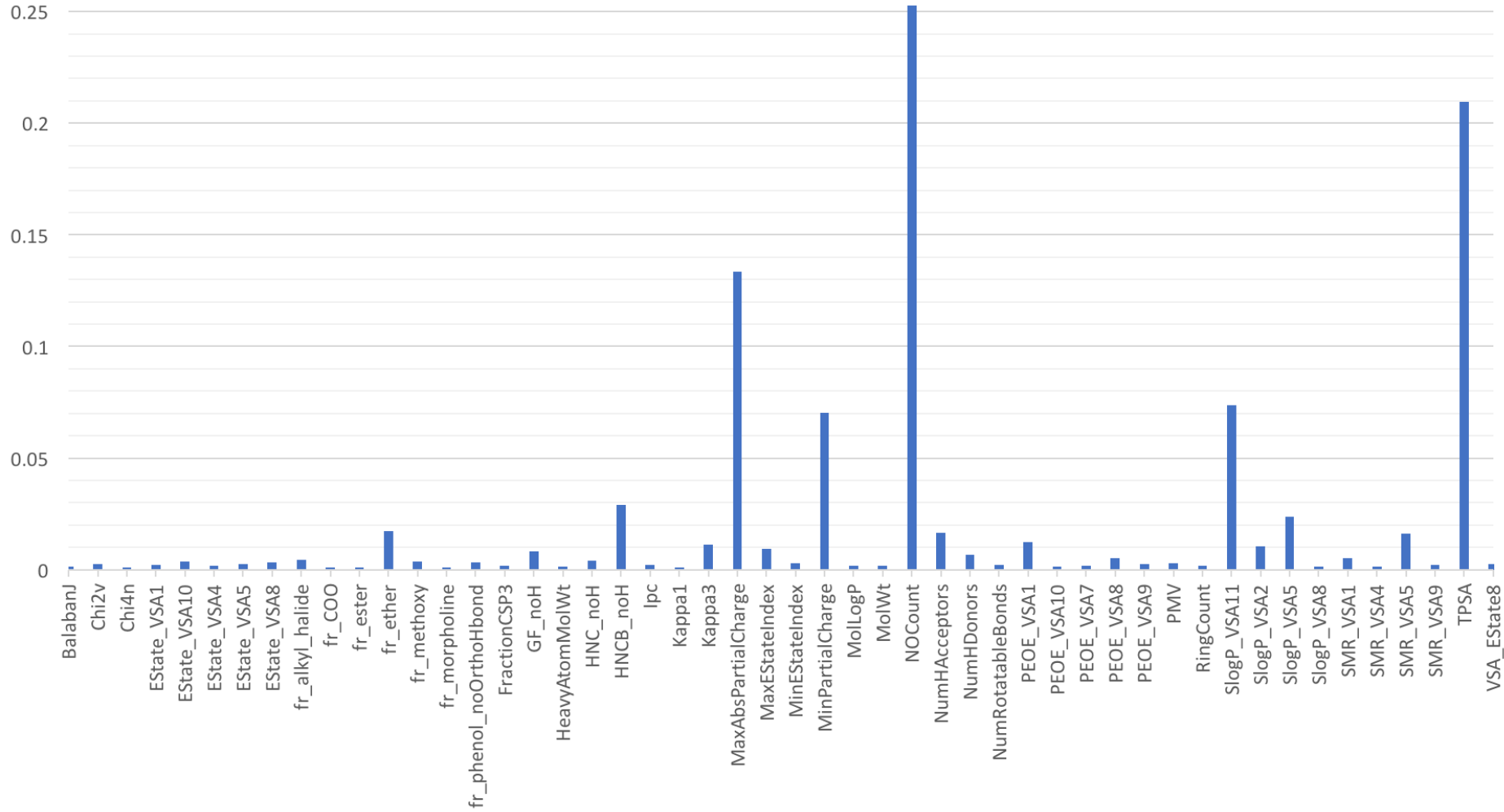


Figure 42 – Feature importances from an ExtraTrees regression of the top 50 molecular descriptors for the prediction of HFE when; when the calculated HFEs (with no corrections) and PMV (with no corrections) are included.

Chapter 6
Conclusions

6.1 Summary and conclusions

In this thesis, we investigate various approaches to solubility prediction and solvation model development, based on informatics and incorporation of empirical data. These approaches can be described as being of a knowledge-based approach or nature, and specifically incorporate structural information from the CSD.

The simplest model involving an empirical quantity was the GSE, discussed and investigated in **Chapter 3.** The GSE is a theoretically justified regression model, which involves the prediction of log S from a compound's melting point and log P. Previous work by others had found that the GSE was able to perform well, with $r^2 = 0.96$, for a set of 1026 organic compounds[101]. Our own dataset was tested with the GSE, and we found significantly poorer results, with the best log P prediction method resulting in a GSE prediction of log S with $r^2 = 0.60$. We suggest that this poor result occurs due to structures which lie in sparsely populated regions of log S; in line with the suggestions made by Ali *et al*[31].

Another consideration is that the melting point of a compound may not always be reported correctly. For our dataset, experimental melting points were taken from the CSD. An effort was instigated in 2015 by CCDC to correct the melting point values available for compounds. Prior to 2015, and at the time our work was conducted, a number of structures were found to contain incorrect melting points, or to have melting points recorded with incorrect units. Although efforts were made in this work to identify and correct these cases, there is still an ongoing effort to correctly identify all of these cases, so there is a possibility that not all of the data in our own dataset is reliable. Furthermore, the amount of melting point data available has been found to be limited[157].

Following the investigation of whether the GSE worked well for our own dataset, we performed a simple linear regression analysis of a number of molecular descriptors, calculated in rdkit, to investigate the correlation between them and log S. We found that descriptors related to

molecular size, shape, and complexity were the most correlated to log S. This suggests that errors in the GSE may be specific to molecular structure.

The final investigation in **Chapter 3** aimed to improve the prediction of log S by running a variety of regression models, with both molecular descriptors and the terms included in the GSE, to find the best possible model. This was done through a brute force methodology, which was facilitated through a CV grid search enabled with a number of scripts and programs (see Electronic Appendix I) wrapped around sklearn. From 22,829 different regression models in the initial grid search, 37 models were selected as models which were generalisable, interpretable, and have low bias and a good explained variance. These models were all fitted by either lasso or elastic net based estimators (see 2.3.3). The best overall model found contained both the melting point and log P terms of the GSE, but also included a number of descriptors based on either simple fragment counts or molecular complexity. The appearance of these descriptors suggests that specific structure-solute interactions or hydrogen bonding networks in the crystalline state may not be sufficiently described by the melting point and log P terms in the GSE. Therefore, we next aimed to investigate the specific interactions of different atoms, with some sort of description of their environment, with water.

This investigation is discussed in **Chapter 4.** In this chapter, we analyse the atom to water pair distributions of a number of atom types, which are defined by their immediate chemical environment according to AMBER atom types. The analysis of these pair distributions is enabled through a novel calculation of RDFs, which are averaged over a number of organic hydrate crystal structures. Each structure is initially atom typed, and the pair distances for each atom type with water are grouped together. The atom type groups from all structures (~6000) are then grouped by atom type, and an RDF is calculated for each atom type with water oxygen (OW) and water hydrogen (HW), as per the usual convention, where the histogram of pair distances is normalised against an unbiased distribution of the same number of particles to give the final *g(r)* function. This method gives qualitative information that is similar to that gained from other methods, such as CSD surveys. However, the relation of *g(r)* to a number of thermodynamic quantities infers the applicability of the functions to empirically parameterised or hybrid solvation models. This inference is justified by a comparison of the OW···HW RDF from our work with experimental RDFs. A comparison to Soper's RDF of OW···HW pairs in water at 298K showed the best correlation, both statistically and visually, to our own model, implying that our averaged RDFs gave information about water similar to that found in the solution phase.

Following this observation, we went on to investigate whether these functions could be reliably used to develop a new empirically parameterised solvation model, based upon the integral equation theory of liquids, specifically 1D-RISM, as discussed in **Chapter 5**. In this chapter, we apply a weighting scheme based upon SASA to the RDFs from **Chapter 4** to estimate solute-water distribution functions for a set of 70 molecules, with and without hydrogen atoms included for the solute. These functions can then be used to estimate the pair potential via the PMF. From the PMF and RDF, the direct and total correlation functions can also be estimated via a HNC closure relation. Finally, all of these functions can be implemented into various (HNC, HNCB and GF) free energy expressions to calculate the HFE. It was found that evaluation of the energy expressions alone was best for the functions where hydrogens were not included. However, the predicted HFEs were incorrect by at least an order of magnitude, and were not very well correlated to the

corresponding HFEs (HNC r^2 = 0.47, HNCB r^2 = 0.51, GF r^2 = 0.43). In order to improve HFE predictions, in accordance with previous findings, the errors of the HNC, HNCB and GF HFEs were regressed against PMV and the same functions with hydrogens included. The relevant corrections were made according to these regressions. It was found that PMV was extremely correlated to the error (HNC R^2 = 0.96, HNCB R^2 = 0.99, GF R^2 = 0.91), and application of a PMV correction improved the HNCB expression (r^2 = 0.63), but did not improve the HNC or GF methods, where the HNC model had the same r^2, and the GF model decreased in prediction accuracy. A final correction to the HNCB and HNC models was made for the RDFs calculated with hydrogen atoms included. The HNCB model prediction accuracy was improved by this correction (r^2 = 0.68, RMSE = 2.69) and was found to be the best model. Although the HNCB model was the best of those evaluated, there were still a large number of outliers found. These corresponded to clusters of similar structures, such as phenol based structures or hydrocarbon chains of varying length, suggesting that an insufficient amount of structural information was inferred by our RDFs. This probably occurs because the organic hydrate crystal structures used to build our RDFs are not similar to all of the structures in the dataset used in the RISM-type models, thus do not describe all of the solutes well.

In order to establish what sort of information our RISM-type models did not include, or may not entirely account for, we included the non-corrected (PMV and hydrogen correction) energy terms in regression models for HFE with molecular descriptors; also performing the same regressions for the same training and test sets without the calculated energy terms included. For two of the three selected (best) regression estimators, inclusion of these terms improved the regression (Lasso and LassoLarsCV). For the third regression estimator (LassoLarsIC), although the statistical measures did not indicate an improvement, a visual analysis of the regressions with and without the extra terms included showed a better agreement when the extra terms were included, but with two large outliers in the set.

Finally, an analysis of the most important features selected by an ExtraTrees regressor was conducted. It was shown that TPSA is the second most important descriptor both when the extra energy terms are included, and when they are not. This finding corroborates the previous observation that 1D-RISM does not describe polar molecules well[158]. Other important descriptors predominantly corresponded to fragment and surface area descriptors. As the conventional 1D-RISM formalism upon which our models are based does not include directional (i.e. specific positions and orientations) information about the distribution of the solvent, it is difficult to establish whether the errors in our model correspond to this, as previously found, or correspond to insufficient information from the RDFs used to produce them.

6.2 Further work

In **Chapter 3**, a variety of estimators were investigated in order to find the best possible regression method for predicting log S. However, more complicated estimators such as neural networks, and more complex (non-linear) SVM methods were not included in the methodology, as these were found to significantly increase the computational cost of the brute force method. Given the adequate resources and time, including these additional estimator types may find better models.

In **Chapter 4**, the atom types used were based upon the existing AMBER forcefield. Further work would aim to investigate different atom typing schemes for these RDFs, and their application to the RISM-type models in **Chapter 5**, in order to find the best possible scheme for HFE calculation. In addition to this, a RISM-type model based upon the 3D-RISM formalism would be preferable to the 1D-RISM formalism used in this work. However, calculating the appropriate RDFs with orientational and directional degrees of freedom included would be much more complex. This would require a scheme to determine how the origin of each system were to be calculated, so that the orientational and directional degrees of freedom are relative for every structure used in the final averaging. One promising solution would be to use something similar to the CCDC IsoStar[159] system, where functional group descriptions are used instead of atom types, and each functional group is least-squares superimposed upon the average geometry of the relevant group.

Bibliography

1 S. Basavaraj and G. V. Betageri, *Acta Pharm. Sin. B*, 2014, **4**, 3–17.

2 H. D. Williams, N. L. Trevaskis, S. A. Charman, R. M. Shanker, W. N. Charman, C. W. Pouton and C. J. H. Porter, *Pharmacol. Rev.*, 2013, **65**, 315–499.

3 C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models*, John Wiley & Sons, Chichester, 2013.

4 S. H. Yalkowsky and S. C. Valvani, *J. Pharm. Sci.*, 1980, **69**, 912–922.

5 A. G. Leach, H. D. Jones, D. A. Cosgrove, P. W. Kenny, L. Ruston, P. MacFaul, J. M. Wood, N. Colclough and B. Law, *J. Med. Chem.*, 2006, **49**, 6672–6682.

6 GSK BioDig, MedChemica, 2014.

7 J. Hussain, *GlaxoSmithKline*, 2011.

8 J. W. Ponder, C. Wu, P. Ren, V. S. Pande, J. D. Chodera, M. J. Schnieders, I. Haque, D. L. Mobley, D. S. Lambrecht, R. A. DiStasio, M. Head-Gordon, G. N. I. Clark, M. E. Johnson and T. Head-Gordon, *J. Phys. Chem. B*, 2010, **114**, 2549–64.

9 S. Y. Liem and P. L. A Popelier, *Phys. Chem. Chem. Phys.*, 2014, **16**, 4122–34.

10 A. A. Noyes and W. R. Whitney, *J. Am. Chem. Soc.*, 1897, **19**, 930–934.

11 A. Jouyban and M. A. A. Fakhree, *Toxicity and Drug Testing*, InTech, 2012.

12 G. Völgyi, E. Baka, K. J. Box, J. E. A Comer and K. Takács-Novák, *Anal. Chim. Acta*, 2010, **673**, 40–6.

13 J. L. McDonagh, N. Nath, L. De Ferrari, T. van Mourik and J. B. O. Mitchell, *J. Chem. Inf. Model.*, 2014, **54**, 844–856.

14 B. A. Hendriksen, M. V. S. Felix and M. B. Bolger, *AAPS PharmSci*, 2003, **5**, 35–49.

15 J. Bauer, S. Spanton, R. Henry, J. Quick, W. Dziki, W. Porter and J. Morris, *Pharm. Res.*, 2001, **18**, 859–866.

16 S. R. Chemburkar, J. Bauer, K. Deming, H. Spiwek, K. Patel, J. Morris, R. Henry, S. Spanton, W. Dziki, W. Porter, J. Quick, P. Bauer, J. Donaubauer, B. A. Narayanan, M. Soldani, D. Riley and K. Mcfarland, *Org. Process Res. Dev.*, 2000, **4**, 413–417.

17 D. S. Palmer, J. L. McDonagh, J. B. O. Mitchell, T. van Mourik and M. V. Fedorov, *J. Chem. Theory Comput.*, 2012, 3322–3337.

18 D. S. Palmer, A. Llinàs, I. Morao, G. M. Day, J. M. Goodman, R. C. Glen and J. B. O. Mitchell, *Mol. Pharm.*, 2008, **5**, 266–279.

19 S. H. Yalkowsky and S. C. Valvani, *J. Pharm. Sci.*, 1980, **69**, 912–922.

20 S. L. Price, M. Leslie, G. W. A. Welch, M. Habgood, L. S. Price, P. G. Karamertzanis and G. M. Day, *Phys. Chem. Chem. Phys.*, 2010, **12**, 8478–8490.

21 A. M. Reilly and A. Tkatchenko, *J. Phys. Chem. Lett.*, 2013, **4**, 1028–1033.

22 E. H. Lee, *Asian J. Pharm. Sci.*, 2014, **9**, 163–175.

23 A. R. Leach and V. J. Gillet, *An introduction to chemoinformatics*, Springer, Revised Ed., 2007.

24 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.

25 S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi and I. Pletnev, *J. Cheminform.*, 2013, **5**, 7.

26 A. R. Leach and V. J. Gillet, *An Introduction to Cheminformatics*, Springer, Dordrecht, 2007.

27 L. D. Hughes, D. S. Palmer, F. Nigsch and J. B. O. Mitchell, *J. Chem. Inf. Model.*, 2008, **48**, 220–232.

28 J. B. O. Mitchell, *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, 2014, **4**, 468–481.

29 Y. Ran and S. H. Yalkowsky, *J. Chem. Inf. Comput. Sci.*, 2001, **41**, 354–357.

30 Y. Ran, Y. He, G. Yang, J. L. H. Johnson and S. H. Yalkowsky, *Chemosphere*, 2002, **48**, 487–509.

31 J. Ali, P. Camilleri, M. B. Brown, A. J. Hutt and S. B. Kirton, *J. Chem. Inf. Model.*, 2012, **52**, 420–8.

32 J. Tomasi, B. Mennucci and R. Cammi, *Chem. Rev.*, 2005, **105**, 2999–3093.

33 L. Onsager, *J. Am. Chem. Soc.*, 1936, **58**, 1486–1493.

34 R. A. Pierotti, *Chem. Rev.*, 1975, **76**, 717–726.

35 S. Höfinger and F. Zerbetto, *J. Phys. Chem. A*, 2003, **107**, 11253–11257.

36 C. Colominas, F. J. Luque and M. Orozco, *Chem. Phys.*, 1999, **240**, 253–246.

37 J. Wang, T. Hou and X. Xu, *J. Chem. Inf. Model.*, 2009, **49**, 571–81.

38 R. M. Levy and E. Gallicchio, *Annu. Rev. Phys. Chem.*, 1998, **49**, 531–67.

39 D. L. Beveridge and F. M. DiCapua, *Annu. Rev. Biophys. Biophys. Chem.*, 1989, **18**, 431–492.

40 R. W. Zwanzig, *J. Chem. Phys.*, 1954, **22**, 1420–1426.

41 C. Chipot and A. Pohorille, in *Free Energy Calculations: Theory and Applications in Chemistry and Biology*, Springer, Berlin, Heidelberg, 2007, pp. 33–75.

42 K. Luder, L. Lindfors, J. Westergren, S. Nordholm and R. Kjellander, *J. Phys. Chem. B*, 2007, **111**, 1883–1892.

43 H. Liu, S. Dai and D. Jiang, *J. Phys. Chem. B*, 2014, **118**, 2719–25.

44 M. A. Wyczalkowski, A. Vitalis and R. V. Pappu, *J. Phys. Chem. B*, 2010, **114**, 8166–80.

45 A. Ahmed and S. I. Sandler, *J. Chem. Eng. Data*, 2015, **60**, 16–27.

46 R. A. Friesner and V. Guallar, *Annu. Rev. Phys. Chem.*, 2005, **56**, 389–427.

47 A. H. Steindal, K. Ruud, L. Frediani, K. Aidas and J. Kongsted, *J. Phys. Chem. B*, 2011, **115**, 3027–37.

48 P. Mark and L. Nilsson, *J. Phys. Chem. A*, 2001, **105**, 9954–9960.

49 M. W. Mahoney and W. L. Jorgensen, *J. Chem. Phys.*, 2000, **112**, 8910.

50 C. Vega and J. L. F. Abascal, *Phys. Chem. Chem. Phys.*, 2011, **13**, 19663–88.

51 H. W. Horn, W. C. Swope, J. W. Pitera, J. D. Madura, T. J. Dick, G. L. Hura and T. Head-Gordon, *J. Chem. Phys.*, 2004, **120**, 9665–78.

52 L. Wang, T. J. Martinez and V. S. Pande, *J. Phys. Chem. Lett.*, 2014, **5**, 1885–1891.

53 D. van der Spoel, P. J. van Maaren and H. J. C. Berendsen, *J. Chem. Phys.*, 1998, **108**, 10220.

54 A. Jones, F. Cipcigan, V. P. Sokhan, J. Crain and G. J. Martyna, *Phys. Rev. Lett.*, 2013, **110**, 227801.

55 D. S. Palmer, A. I. Frolov, E. L. Ratkova and M. V. Fedorov, *J. Phys. Condens. Matter*, 2010, **22**, 492101.

56 T. Luchko, S. Gusarov, D. R. Roe, C. Simmerling, D. A. Case, J. Tuszynski and A. Kovalenko, *J. Chem. Theory Comput.*, 2010, **6**, 607–624.

57 D. Chandler, J. D. McCoy and S. J. Singer, *J. Chem. Phys.*, 1986, **85**, 5971.

58 A. Kovalenko and F. Hirata, *J. Chem. Phys.*, 1999, **110**, 10095.

59 H. C. Andersen and D. Chandler, *J. Chem. Phys.*, 1972, **57**, 1918–1929.

60 E. L. Ratkova, D. S. Palmer and M. V. Fedorov, *Chem. Rev.*, 2015, **115**, 6312–6356.

61 T. Hahn, *International Tables for Crystallography: Volume A*, Springer, 5th edn., 2005.

62 H. Arnold, in *International Tables for Crystallography Vol.A*, IUCr, 2006, pp. 86–89.

63 H. Wondratschek, in *International Tables for Crystallography Vol.A*, IUCr, 2006, pp. 720–725.

64 W. Fischer, E. Koch and H. Arnold, in *International Tables for Crystallography Vol.A*, IUCr, 2006, pp. 812–816.

65 M. F. Costa Gomes and A. A. H. Pádua, in *Developments and Applications in Solubility*, Royal Society of Chemistry, Cambridge, 2007, pp. 153–170.

66 C. G. Gray and K. E. Gubbins, in *Theory of Molecular Fluids*, 1984, pp. 398–400.

67 S. J. Singer and D. Chandler, *Mol. Phys.*, 1985, **55**, 621–625.

68 M. S. Wertheim, *Phys. Rev. Lett.*, 1963, **10**, 321–323.

69 J. K. Percus and G. J. Yevick, *Phys. Rev.*, 1958, **110**, 1–13.

70 D. Chandler, *Mol. Phys.*, 1976, **31**, 1213–1223.

71 J. M. J. van Leeuwen, J. Groeneveld and J. de Boer, *Physica*, 1959, **25**, 792–808.

72 T. Morita, *Prog. Theor. Phys.*, 1960, **23**, 829–845.

73 G. S. Rushbrooke, *Physica*, 1960, **26**, 259–265.

74 E. Meeron, *J. Math. Phys.*, 1960, **1**, 192.

75 T. Morita and K. Hiroike, *Prog. Theor. Phys.*, 1960, **23**, 1003–1027.

76 I. M. Mladenov, *Europhys. Lett.*, 1996, **33**, 577–581.

77 E. Anderson, G. D. Veith and D. Weininger, *SMILES, a line notation and computerized interpreter for chemical structures*, US Environmental Protection Agency, Environmental Research Laboratory, 1987.

78 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1990, **30**, 237–243.

79 D. Weininger, A. Weininger and J. L. Weininger, *J. Chem. Inf. Comput. Sci.*, 1989, **29**, 97–101.

80 H. L. Morgan, *J. Chem. Doc.*, 1965, **5**, 107–113.

81 R. Mannhold and H. van de Waterbeemd, *J. Comput. Aided. Mol. Des.*, 2001, **15**, 337–354.

82 T. Fujita, J. Iwasa and C. Hansch, *J. Am. Chem. Soc.*, 1964, **86**, 5175–5180.

83 R. F. Rekker, *The hydrophobic fragmental constant*, Elsevier, Amsterdam, 1977, vol. 1.

84 R. F. Rekker and R. Mannhold, *Calculation of drug lipophilicity: the hydrophobic fragmental constant approach*, Wiley-VCH, 1992.

85 A. J. Leo, *Chem. Rev.*, 1993, **93**, 1281–1306.

86 A. K. Ghose and G. M. Crippen, *J. Chem. Inf. Model.*, 1987, **27**, 21–35.

87 R. Wang, Y. Fu and L. Lai, *J. Chem. Inf. Comput. Sci.*, 1997, **37**, 615–621.

88 A. K. Ghose, V. N. Viswanadhan and J. J. Wendoloski, *J. Phys. Chem. A*, 1998, **102**, 3762–3772.

89 S. A. Wildman and G. M. Crippen, *J. Chem. Inf. Comput. Sci.*, 1999, **39**, 868–873.

90 H. Wiener, *J. Am. Chem. Soc.*, 1947, **69**, 17–20.

91 M. Randic, *J. Am. Chem. Soc.*, 1975, **97**, 6609–6615.

92 L. B. Kier and L. H. Hall, *Molecular connectivity in structure-activity analysis*, Research Studies, 1986.

93 L. H. Hall, B. Mohney and L. B. Kier, *Quant. Struct. - Act. Relationships*, 1991, **10**, 43–51.

94 T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Stanford, 2nd edn., 2008.

95 R. Tibshirani, *J. R. Stat. Soc. B*, 1996, **58**, 267–288.

96 R. Tibshirani, *J. R. Stat. Soc. B*, 2011, **73**, 273–282.

97 T. T. Wu and K. Lange, *Ann. Appl. Stat.*, 2008, **2**, 224–244.

98 B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, *Ann. Stat.*, 2004, **32**, 407–499.

99 H. Zou and T. Hastie, *J. R. Stat. Soc.*, 2005, **67**, 301–320.

100 K. P. Burnham and D. A. Anderson, *Model Selection and Multimodel Inference*, Springer New York, New York, NY, 2nd edn., 2004.

101 A. Lusci, G. Pollastri and P. Baldi, *J. Chem. Inf. Model.*, 2013, **53**, 1563–1575.

102 C. M. Wassvik, A. G. Holmén, R. Draheim, P. Artursson and C. A. S. Bergström, *J. Med. Chem.*, 2008, **51**, 3035–3039.

103 O. Wolk, R. Agbaria and A. Dahan, *Drug Des Devel Ther*, 2014, **8**, 1563–1575.

104 C. Ouvrard and J. B. O. Mitchell, *Acta Crystallogr. Sect. B*, 2003, **59**, 676–685.

105 F. Lovering, J. Bikker and C. Humblet, *J. Med. Chem.*, 2009, **52**, 6752–6.

106 M. Salahinejad, T. C. Le and D. A. Winkler, *Mol. Pharm.*, 2013, **10**, 2757–2766.

107 F. Pedregosa, R. Weiss and M. Brucher, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.

108 T. J. Hou, K. Xia, W. Zhang and X. J. Xu, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 266–75.

109 J. Huuskonen, *J. Chem. Inf. Model.*, 2000, **40**, 773–777.

110 J. S. Delaney, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 1000–5.

111 U. S. Estimation Programs Interface (EPA) 4.11, 2012.

112 G. Landrum, 2015, http://www.rdkit.org/.

113 C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann and E. Willighagen, *J. Chem. Inf. Comput. Sci.*, 2003, **43**, 493–500.

114 J. L. Finney, in *Journal of Physics: Conference Series*, 2007, vol. 57, pp. 40–52.

115 K. Morris, in *Polymorphism in Pharmaceutical Solids*, ed. H. G. Brittain, Marcel Dekker, New York, 1999, pp. 125–181.

116 G. R. Desiraju, *J. Chem. Soc. Chem. Commun.*, 1991, 426.

117 L. Infantes, L. Fabian and W. D. S. Motherwell, *CrystEngComm*, 2007, **9**, 65.

118 J. van de Streek and S. Motherwell, *CrystEngComm*, 2007, **9**, 55.

119 L. Infantes and S. Motherwell, *CrystEngComm*, 2002, **4**, 454.

120 I. J. Bruno, J. C. Cole, P. R. Edgington, M. Kessler, C. F. Macrae, P. McCabe, J. Pearson and R. Taylor, *Acta Crystallogr. Sect. B*, 2002, **58**, 389–397.

121 L. Infantes, J. Chisholm and S. Motherwell, *CrystEngComm*, 2003, **5**, 480.

122 G. A. Jeffrey and H. Maluszynska, *Acta Crystallogr. Sect. B Struct. Sci.*, 1990, **46**, 546–549.

123 M. Mascal, L. Infantes and J. Chisholm, *Angew. Chem. Int. Ed. Engl.*, 2005, **45**, 32–6.

124 P. T. A. Galek, J. A. Chisholm, E. Pidcock and P. A. Wood, *Acta Crystallogr. B. Struct. Sci. Cryst. Eng. Mater.*, 2014, **70**, 91–105.

125 A. K. Soper, *Chem. Phys.*, 2000, **258**, 121–137.

126 C. R. Groom, I. J. Bruno, M. P. Lightfoot and S. C. Ward, *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.*, 2016, **72**, 171–179.

127 J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman and D. A. Case, *J. Comput. Chem.*, 2004, **25**, 1157–1174.

128 J. Wang, W. Wang, P. A. Kollman and D. A. Case, *J. Mol. Graph. Model.*, 2006, **25**, 247–60.

129 J. Rodriguez-Carvajal and J. Gonzalez-Platas, 2013.

130 M. I. Aroyo, J. M. Perez-Mato, E. T. D. Orobengoa, G. de la Flor and A. Kirov, *Bulg. Chem. Commun.*, 2011, **43**, 183–197.

131 M. I. Aroyo, J. M. Perez-Mato, C. Capillas, E. Kroumova, S. Ivantchev, G. Madariaga, A. Kirov and H. Wondratschek, *Z. Krist.*, 2006, **221**, 15–27.

132 M. I. Aroyo, A. Kirov, C. Capillas, J. M. Perez-Mato and H. Wondratschek, *Acta Cryst.*, 2006, **A62**, 115–128.

133 S. Le Roux and V. Petkov, *J. Appl. Crystallogr.*, 2010, **43**, 181–185.

134 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, Oxford, 5th edn., 1991.

135 C. F. Macrae, I. J. Bruno, J. A. Chisholm, P. R. Edgington, P. McCabe, E. Pidcock, L. Rodriguez-Monge, R. Taylor, J. van de Streek and P. A. Wood, *J. Appl. Crystallogr.*, 2008, **41**, 466–470.

136 P. Atkins and J. de Paula, *Physical Chemistry for the Life Sciences*, Oxford University Press, illustrate., 2011.

137 J. G. Kirkwood, *J. Chem. Phys.*, 1935, **3**, 300.

138 A. Savitzky and M. J. E. Golay, *Anal. Chem.*, 1964, **36**, 1627–1639.

139 J. D. Bernal and R. H. Fowler, *J. Chem. Phys.*, 1933, **1**, 515.

140 K. Molcanov, B. Kojić-Prodić and N. Raos, *Acta Crystallogr. B.*, 2004, **60**, 424–32.

141 J. P. M. Lommerse, S. L. Price and R. Taylor, *J. Comput. Chem.*, 1997, **18**, 757–774.

142 E. L. Ratkova, G. N. Chuev, V. P. Sergiievskyi and M. V. Fedorov, *J. Phys. Chem. B*, 2010, **114**, 12068–12079.

143 G. N. Chuev, M. V. Fedorov and J. Crain, *Chem. Phys. Lett.*, 2007, **448**, 198–202.

144 D. S. Palmer, M. Misin, M. V Fedorov and A. Llinas, *Mol. Pharm.*, 2015, **12**, 3420–3432.

145 H. Freedman and T. N. Truong, *Chem. Phys. Lett.*, 2003, **381**, 362–367.

146 J. S. Perkyns and B. M. Pettitt, *J. Chem. Phys.*, 1992, **97**, 7656–7666.

147 A. Kovalenko and F. Hirata, *J. Chem. Phys.*, 2000, **113**, 2793–2805.

148 D. Chandler, Y. Singh and D. M. Richardson, *J. Chem. Phys.*, 1984, **81**, 1975–1982.

149 S. Ten-no, *J. Chem. Phys.*, 2001, **115**, 3724–3731.

150 D. L. Mobley and J. P. Guthrie, *J. Comput. Aided. Mol. Des.*, 2014, **28**, 711–720.

151 CCDC, *Python API*.

152 M. Clark, R. D. Cramer and N. Van Opdenbosch, *J. Comput. Chem.*, 1989, **10**, 982–1012.

153 B. Lee and F. M. Richards, *J. Mol. Biol.*, 1971, **55**, 379–IN4.

154 S. Mitternacht, *F1000Research*, 2016, **5**, 1–12.

155 V. P. Sergiievskyi, W. Hackbusch and M. V. Fedorov, *J. Comput. Chem.*, 2011, **32**, 1982–1992.

156 S. Chiodo, G. N. Chuev, S. E. Erofeeva, M. V. Fedorov, N. Russo and E. Sicilia, *Int. J. Quantum Chem.*, 2007, **107**, 265–274.

157 I. V Tetko, Y. Sushko, S. Novotarskyi, L. Patiny, I. Kondratov, A. E. Petrenko, L. Charochkina and A. M. Asiri, *J. Chem. Inf. Model.*, 2014, **54**, 3320–3329.

158 A. I. Frolov, E. L. Ratkova, D. S. Palmer and M. V. Fedorov, *J. Phys. Chem. B*, 2011, **115**, 6011–6022.

159 I. J. Bruno, J. C. Cole, J. P. Lommerse, R. S. Rowland, R. Taylor and M. L. Verdonk, *J. Comput. Aided. Mol. Des.*, 1997, **11**, 525–37.