Crystallization Properties of Molecular Materials: Prediction and Rule Extraction by Machine Learning

Jerome G. P. Wicker

Worcester College

A thesis submitted for the degree of

Doctor of Philosophy

Department of Chemistry

University of Oxford

Trinity Term 2017

THE UNIVERSITY OF OXFORD

DEPARTMENT OF CHEMISTRY

The work reported in this thesis was carried out in the Chemistry Research Laboratory in Oxford between October 2013 and April 2017 under the supervision of Professor Richard Cooper and Professor Bill David. All the work is my own, unless stated to the contrary, and has not been submitted, either in full or part, for any degree in this or any other university.

Jerome George Pierre Wicker

For my parents

Abstract

Crystallization Properties of Molecular Materials: Prediction and Rule Extraction by Machine Learning
Jerome Wicker
Worcester College
Doctor of Philosophy
Trinity Term 2017

Crystallization is an increasingly important process in a variety of applications from drug development to single crystal X-ray diffraction structure determination. However, while there is a good deal of research into prediction of molecular crystal structure, the factors that cause a molecule to be crystallizable have so far remained poorly understood.

The aim of this project was to answer the seemingly straightforward question: can we predict how easily a molecule will crystallize? The Cambridge Structural Database contains almost a million examples of materials from the scientific literature that have crystallized. Models for the prediction of crystallization propensity of organic molecular materials were developed by training machine learning algorithms on carefully curated sets of molecules which are either observed or not observed to crystallize, extracted from a database of commercially available molecules. The models were validated computationally and experimentally, while feature extraction methods and high resolution powder diffraction studies were used to understand the molecular and structural features that determine the ease of crystallization. This led to the development of a new molecular descriptor which encodes information about the conformational flexibility of a molecule.

The best models gave error rates of less than 5% for both cross-validation data and previously-unseen test data, demonstrating that crystallization propensity can be predicted with a high degree of accuracy. Molecular size, flexibility and nitrogen atom environments were found to be the most influential factors in determining the ease of crystallization, while microstructural features determined by powder diffraction showed almost no correlation with the model predictions. Further predictions on co-crystals show scope for extending the methodology to other relevant applications.

Table of Contents

List of Tables

List of Figures

Acknowledgements

There are a number of people who I would like to thank for helping me through the past three and a half years of work that this thesis represents. First of all I would like to thank my supervisor, Professor Richard Cooper, who introduced me to machine learning in the first place. His guidance, support and calming influence have been invaluable to me, always showing confidence in my ability while reminding me that it is important to enjoy myself too. Thanks must also go to my co-supervisor, Professor Bill David, for his advice on all things related to powder diffraction, and for his excellent knowledge of Grenoble's restaurants.

The members of ChemCryst, past and present, have made my experience in the 118 office a pleasure. The pub quizzes and trips to Atomic Burger made me look forward to going into the CRL every day. Dr Pascal Parois, an ever-present in the office during my DPhil., could always be relied upon for a Gallic one-liner, while Amber and Kirsten provided much moral support. I'd also like to thank everyone who has been directly involved with the project: collaborators Max and Trixie at Novartis, Lorraine and Simon at University College Cork, and beamline scientists at the ESRF.

Special thanks must go to Dr. Karim Sutton, who was a trailblazer in the group and a shining example to me. The West Wing binges, long hours in the bar and post-conference road trips will forever remain as some of my fondest memories. Peter Thygesen, with whom I have shared some tough recent nights as well as some fun ones, kept me sane through the thesis writing process. I'll miss our football/burger nights at the Uni Club.

Last, but by no means least, I would like to thank Laura, who has made the last few months so much more enjoyable, as well as my parents, friends and family for their support and encouragement.

Abbreviations

Å	Ångström(s), 10^{-10} m
API	active pharmaceutical ingredient
AUC	area under curve
°C	degree(s) Celsius
CSD	Cambridge Structural Database
DCM	dichloromethane
DMF	dimethyl formamide
DFT	density functional theory
DUD	Directory of Useful Decoys
EtOH	ethanol
FN	false negative
FP	false positive
FWHM	full width half maximum
g	gram(s)
HBP	hydrogen bond propensity
InChI	IUPAC International Chemical Identifier
IR	infrared
IUPAC	International Union of Pure and Applied Chemistry
logP	octanol-water partition coefficient
K	Kelvin
m	metre(s)
μ	micro
MMFF94	Merck molecular force field
MQN	molecular quantum number

MR	molar refractivity
NMR	nuclear magnetic resonance spectroscopy
QSPR	quantitative structure-property relationship
RBC	rotatable bond count
RBF	radial basis function
RF	random forest
RMS	root mean square
ROC	receiver operating characteristic
s	second(s)
SDF	structure data file
SMARTS	SMILES Arbitrary Target Specification
SMILES	Simplified Molecular-Input Line-Entry System
SSIP	surface site interaction point
SVM	support vector machine
SXRD	single crystal X-ray diffraction
TN	true negative
TP	true positive
UFF	Universal Force Field
VSA	Van der Waals surface area
λ	wavelength
ZINC	A Free Database of Commercially Available Compounds for Virtual Screening

Chapter 1

Introduction

Crystallization underpins many scientific and industrial applications from pharmaceutical development to polymer chemistry, yet it is little understood, and the ability to predict if, how and why a given material is likely to crystallize is one of the biggest challenges currently facing chemistry. In recent years, the size and availability of databases with information on this topic has increased drastically, and simultaneously machine learning algorithms have been developed which allow us to mine and extract useful information from such "big data" sources faster and in greater depth than a human being.

In this thesis, methods for predicting the ease of crystallization of small organic molecules are proposed, validated and used to rationalise the molecular and structural factors aiding or hindering crystallization.

Contents

1.1 Crystallization

The crystallization properties of a particular material are of great importance in a wide number of applications, ranging from industrial processes to single crystal X-ray diffraction (SXRD) structure determination and racemate separation.[1] However, crystallization is still a poorly understood process.[2]

1.1.1 Drug development process

In recent years there has been a decrease in productivity in the pharmaceutical industry.[3, 4] This has been caused by greater costs in the development of drugs, and high attrition rates for both efficacy and safety reasons.[5] As a result, there has been an increase in the use of computational drug discovery methods in order to identify more suitable candidates which are less likely to fail at some stage of the drug discovery pipeline.

One of the greatest challenges currently facing pharmaceutical chemists is predicting the crystallization propensity of targets.[6] Despite this, relatively little work has been published on this subject. It would be useful to be able to identify potential targets with the desirable crystallization properties computationally before synthesis, to save time and resources.

There are two main situations where knowledge of the crystallization propensity of a molecule is useful. One is the scenario where an amorphous material is desired to increase bioavailability by improving oral absorption.[7] Often the crystalline state has low water solubility and therefore requires compounding in a special dosage form to increase the bioavailability of the drug,[8] and finding amorphous counterparts is one of the most promising ways of overcoming this problem.[9] However, unexpected crystallization of a material which has been engineered to be amorphous is a key cause of failure as it inhibits the performance of the drug,[6] so it would be useful to be able to identify materials which have this tendency in advance.

The second case is due to crystallization often being used as a separation and purification step. Early on in the drug discovery process, a medicinal chemist will attempt to obtain a pure crystalline material which is stable enough to be used in early clinical formulations.[6] Furthermore, crystallization or recrystallization is a convenient way to isolate and purify a material,[10] which means that crystallization is widely used during the final stages of purification and separation of active pharmaceutical ingredients[11] and consequently over 90% of drugs are delivered in crystalline form.[12] The crystalline form has desirable properties, including enhanced

thermodynamic stability relative to amorphous forms.[13] New drug applications also require proof of the structure of the candidate, in which case it is preferable to be able to grow a single crystal of the material,[14] since single-crystal X-ray diffraction is the most reliable technique for structure determination.[15] In these cases, the ability to identify and select targets which are likely to crystallize would prevent failures further along the drug discovery pipeline.

1.1.2 Crystallization process

Crystallization can be thought of as the ideal case of molecular self-assembly,[1] and is generally thought to occur in two key phases; nucleation, and the subsequent growth of these nuclei into crystals.[16]

Nucleation

Nucleation is the "process of fluctuational appearance of nanoscopically small molecular clusters of the new crystalline phase".[17] The process is important since it sets the initial size distribution of the crystals.[18] The general process involves the dissolution of material to form a saturated solution, which is then either heated or left to evaporate to create a supersaturated solution.[19] The supersaturated solution is higher in energy than the saturated solution containing nuclei, so nucleation is favourable and returns the system back to the equilibrium saturated phase.

Homogeneous nucleation is the case where nucleation occurs spontaneously from this non-equilibrium metastable phase, while heterogeneous nucleation involves the introduction of an additive to the solution to promote nucleation, a pathway which has a lower activation energy.[20] Nucleation may not occur immediately if the solution can remain in the metastable supersaturated state, with the time between supersaturation and the formation of a detectable amount of crystalline solid being termed the induction time.[17]

Classical nucleation theory has been widely used to describe the homogeneous

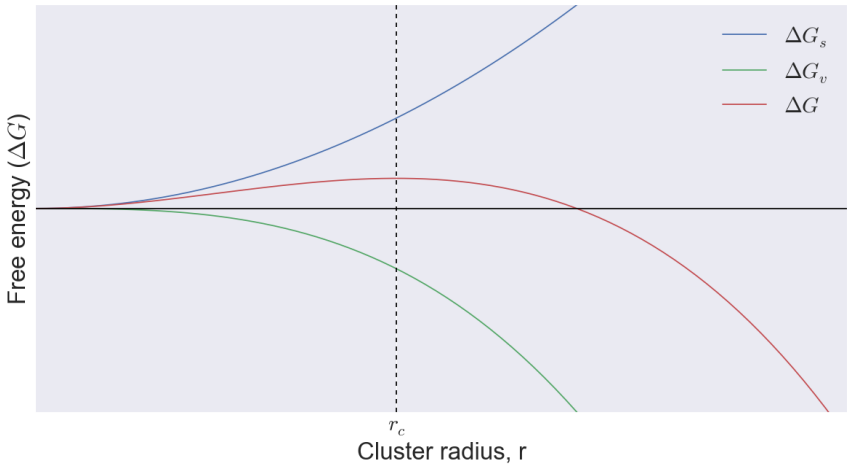


Figure 1.1 Free energy diagram for nucleation.

nucleation process due to its simplicity. This theory is based on a number of assumptions: initial groups of molecules form a cluster which is modelled as a sphere, and the growth of the cluster occurs by sequential addition of single monomers. Monomers are assumed to attach and detach in the correct orientation, so the cluster has the same thermodynamic properties as the bulk material.[21] The free energy change for the process, ΔG, is a trade off between the free energy change of the phase change (ΔG_v), and the free energy change for surface formation (ΔG_s). The first term, related to the cluster volume, is negative due to the greater stability of the solid compared to the solute, which promotes cluster growth. The second term is positive and causes an increase in the free energy proportional to the surface area of the cluster, favouring dissolution. At small cluster radii, the second term is dominant, so these small clusters redissolve quickly. As the cluster size increases, the total free energy change passes through a maximum, above which growth is energetically favourable, as shown in Figure 1.1. Once a cluster has grown sufficiently to reach this critical size, this is termed a nucleus.

The nucleation rate, J can be defined according to this theory as follows:

$$J = A\exp\left(-\frac{B}{\ln^2 S}\right) \tag{1.1}$$

where A and B are constants and S is the supersaturation ratio, $\frac{C}{C_s}$, with C and C_s

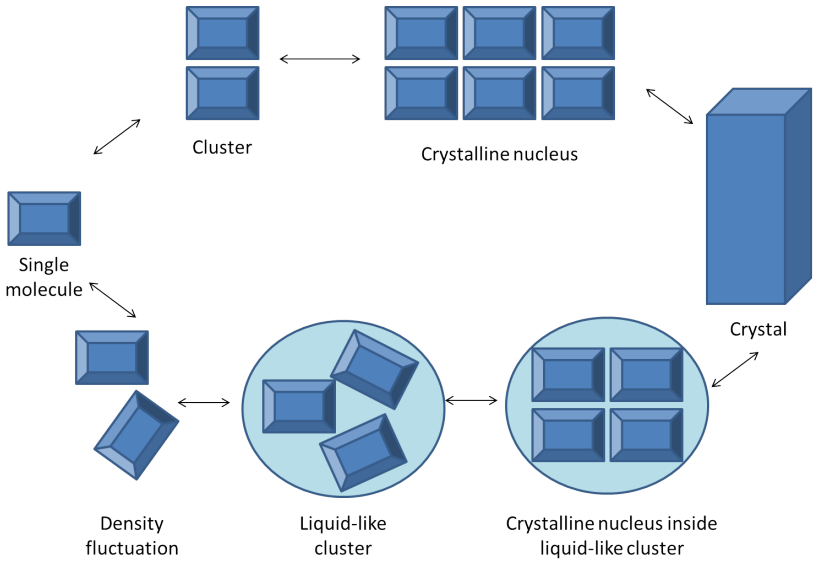


Figure 1.2 Single– and two–step nucleation, from Davey *et al.*[25]

being the actual and equilibrium concentrations of the solute. Since J in this equation has a non-linear relationship with S, small changes in supersaturation can cause large changes in the nucleation rate.

An alternative non–classical nucleation theory has been proposed whereby nucleation is considered to be a two-step process.[22] This involves a liquid-liquid separation in which a dense, disordered, solute–rich liquid forms. The second step involves reorganisation of this to the ordered phase which allows for further crystal growth (Figure 1.2). This has particularly been shown to be consistent with inorganic materials,[23] although it has also been observed in small molecule crystallization.[24]

However, these theories take no consideration of the molecular effects on the nucleation, although molecular information is hidden within the constants.[25] It has been established that conformational flexibility has an important role to play in crystallization because a molecule may exist as a mixture of many different conformers in solution, which could all be of similar energies.[26] For crystallization to occur, the molecule must achieve the "correct" conformation so that it can nucleate and then grow into a crystal. If there are several crystallizing conformations, then this will lead to polymorphism, with differing crystal structures, but there may also be only a single crystallizing conformer. The more conformationally flexible that molecule is,

the larger the number of potential conformers it will have in equilibrium in solution. This effectively dilutes the concentration of the desired conformer, decreasing the degree of supersaturation and therefore the crystallization tendency.

This effect is further increased when the crystallizing conformer is of a relatively high energy, as the beginning of the crystallization process will deplete this conformer and it will need to be replaced for crystallization to continue, which occurs at a rate dependent on the energy barrier.[26] Studies modelling the crystallization from a solution containing multiple conformers have also found that, particularly at low temperatures, the crystal growth is slowed by poisoning of the crystal surface by conformers other than the crystallizing conformer.[27, 28]

The ability to nucleate is therefore a key factor to consider when attempting to assess the ease of crystallization of a material.

Crystal growth

Once nuclei have been created, further growth must occur for the solid to form a crystal of sufficient size for SXRD characterisation. The solubility curve diagram in Figure 1.3 shows the three key regions related to crystallization: the undersaturated zone, the metastable zone and the labile zone. For nucleation to occur, the material must move from the undersaturated region of the solubility diagram to the supersaturated regions of either the labile zone or the metastable zone, and this can either be achieved by cooling (to reduce the temperature), or by evaporation (to increase the concentration).

For the single crystal to grow to a sufficient size, the material must be in the metastable zone of the solubility diagram. For a given temperature, the metastable zone is the region between the equilibrium saturation temperature (the point at which all the solute dissolves) and the critical nucleation temperature (the point at which the solute spontaneously precipitates out of solution). Below the critical nucleation temperature, in the labile zone, spontaneous nucleation occurs, meaning that the

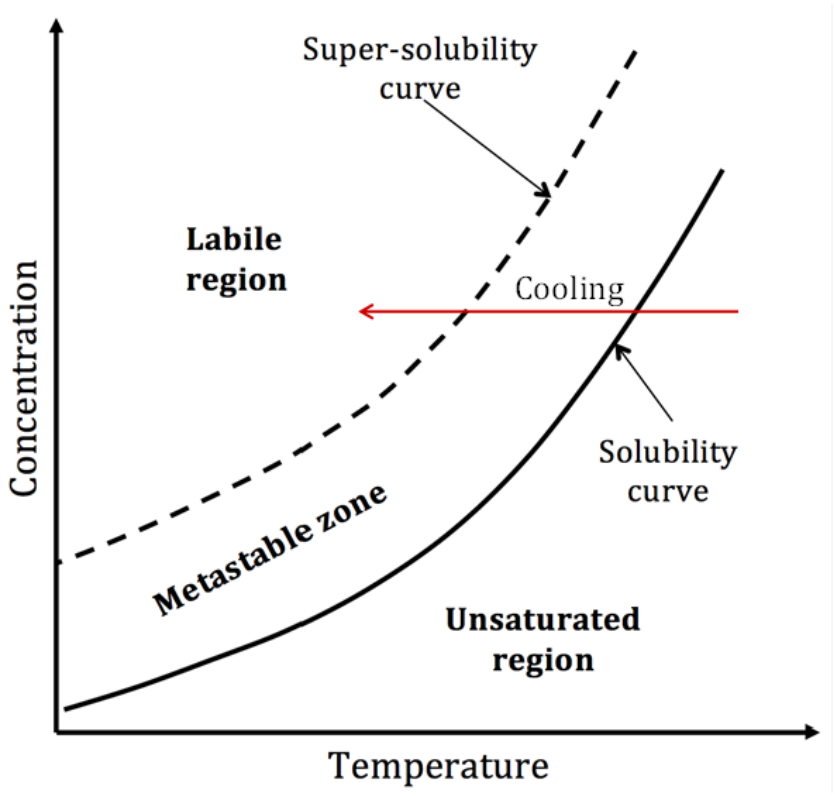


Figure 1.3 Schematic solubility curve diagram.

material crystallizes too quickly and therefore forms a powder rather than growing into a single crystal.[29]

Therefore, the material must remain within the metastable zone for as long a period as possible on cooling to obtain high quality single crystals. This is easy for a material with a wide metastable zone, but for a material with a narrower metastable zone, the cooling rate of the solution must be carefully adjusted to promote crystal growth over precipitation. Indeed, the rule of thumb for industrial processes is that crystallizers should be operated in the middle of the metastable zone,[30] a region that is not easy to attain if the metastable zone is narrow. Although the metastable zone width has been found to be dependent on the volume of the solution,[31] the intrinsic metastable zone width of a material must still have an important effect on the ease of crystallization of a material, with a large metastable zone width being associated with a greater ease of crystallization.

A number of techniques for obtaining single crystals have been documented.[32]

The most common methods involve increasing the supersaturation of the solution by evaporation from a single solvent, or cooling of a hot solution. Vapour diffusion is another commonly–used technique which provides a way of increasing the supersaturation of the solution by equilibration of a saturated solution with an antisolvent in which the solute is less soluble than in the original solvent. In some cases even these methods are unsuccessful in forming crystals, with materials either remaining as powders or forming oils or pastes, and considerable time and resources can be wasted attempting to crystallize materials which may be impossible to crystallize. It would be useful to identify these cases in advance, to either discard them in favour of alternatives which are easier to crystallize, or to guide the choice of conditions under which recrystallization should be attempted.

1.1.3 Co-crystallization

Co-crystals are multi-component crystalline materials that can be assembled via intermolecular interactions, including hydrogen bonds, halogen bonds and/or π–π stacking. The four most common hydrogen bond supramolecular synthons used in the design-phase of co-crystallization studies are shown in Figure 1.4. For a hydrogen-bonded co-crystal to form, there must be a degree of complementarity between the two components (coformers), thus, careful coformer selection is crucial.[33, 34] The hierarchical nature of supramolecular synthons is considered a key factor in accessing heteromeric interactions in the solid state.

Methods for predicting co-crystal formation have focussed on comparison of lattice energies of the co-crystal and the pure components,[36] or co-crystal structure prediction.[37] However, such methods are based on the generation of trial structures, which is a computationally expensive approach and requires significant calculation for each new set of potential co-crystal components.

The strength of potential interactions between donors and acceptors within co-crystals relative to the pure components can be estimated for solution co-crystallization

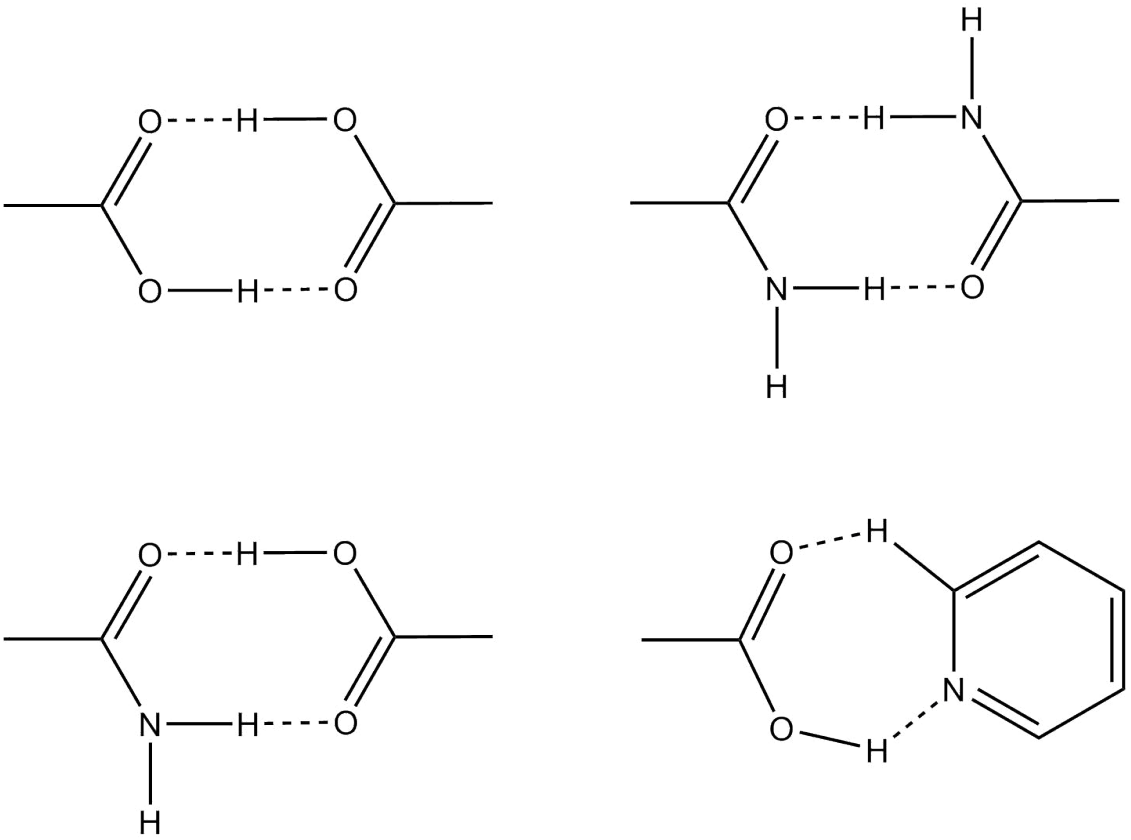


Figure 1.4 The four most common supramolecular synthons.[35]

in the form of pair interactions characterised by pulsed gradient spin-echo nuclear magnetic resonance.[38]

It has also been reported that effective co-crystal screening can be achieved using a fluid-phase thermodynamics model to calculate the excess enthalpy of the interactions between a mixture of active pharmaceutical ingredient (API) and co-former relative to the pure components in a virtually supercooled liquid mixture (which can be approximated to the mixed solid phase crystal).[39]

Hunter proposed a set of rules for quantifying the molecular interactions within co-crystals in the solid state. Calculations of the molecular electrostatic potential surface are used to assign hydrogen bond strength parameters to the donors and acceptors within a molecule, which are collectively known as surface site interaction points (SSIPs).[40] Since the strongest donors and acceptors are most likely to hydrogen bond with each other,[41] a hierarchical list of these can be used to calculate the

interaction energy for each pairing until all possible contacts have been made, with excess donors or acceptors being ignored. The sum of these interaction energies gives a measure of the stability of a co-crystal relative to the pure components without any knowledge of three-dimensional structure.[35] This approach to giving an estimate of the probability of a co-crystal forming can be used to rank a set of potential crystal coformers, which has been shown to be able to identify new co-crystals.[42]

A similar method based on hydrogen bond propensity (HBP) analysis of the Cambridge Structural Database (CSD) to determine the likelihood of co-crystal formation by assessing the probability of the homo- and hetero-interactions has been shown to correctly identify co-crystals of paracetamol.[43] Further statistical analysis of the descriptors of components of co-crystal structures extracted from the CSD has uncovered correlations between the shape and polarity of co-crystal components,[44] but the lack of data on failed co-crystallizations means that no predictive model has been derived from this to date.

1.1.4 Crystal structure/polymorph prediction

The question of whether crystal structure itself is predictable has been asked since the mid-1990s.[45] In practice, this requires generation of many trial structures from the chemical diagram, which are ranked according to their relative energy, as calculated by some approximate force-field method,[46] followed by density functional theory (DFT) to find the global energy minimum, which should correspond to the experimental crystal structure.[47]

In many cases, several crystal structures may be observed in which only the same molecules pack in different ways, with no arrangement being significantly more favourable than any other, which is termed polymorphism.[48] The low-energy minima within a certain energy of the global minimum correspond to thermodynamically feasible crystal structures, and this is known as the crystal energy landscape.[49] However, there are nearly always more structures in this landscape than experimentally ob-

served structures, and so an understanding of the crystallization kinetics is required to determine which polymorphs identified by the thermodynamic models are actually kinetically feasible too.[50] This involves using an attachment energy model to identify those structures which have a kinetic advantage in crystal growth and are therefore more likely to be observed,[51] but progress still needs to be made in understanding the kinetic factors affecting crystallization.[52] If none of the polymorphs have crystal faces with advantageous crystal growth, then the molecule may not crystallize at all, while if there are many low energy polymorphs of similar energy, macroscopic crystal growth may be severely inhibited, with significant disorder in the solid phase which forms.[53]

A number of blind tests have been carried out to attempt to predict the structures of molecules, with varying but improving results.[54–59] However, while these methods are valuable in understanding pharmaceutical solids,[52] these calculations are computationally expensive and routinely take between 3000 and 200,000 CPU hours to complete.[58]

1.1.5 Protein crystallizability

One area in which the prediction of crystallization has already been attempted is in the field of proteins. It is accepted that there is a close relationship between the 3-dimensional structure of a protein and its function. Nuclear magnetic resonance (NMR) is one method of determining the structure, but is time-consuming and costly, and the preferred method is to use SXRD. However, the crystallization of proteins is a non-trivial task due to the complex physico-chemical nature of protein crystallization, which depends on many kinetic factors such as equilibration rates, molecular association, nucleation and crystal growth, about which little is known of the influence.[60] The result is that much time and resources are wasted on non-crystallizable proteins.

Several different methods have been implemented which attempt to solve this

problem. These knowledge-based approaches take advantage of the availability of databases which catalogue both the successful and failed crystallizations of proteins. The general method involves representing each individual protein in the database by a set of descriptors, usually derived from the amino acid composition and sequence. This is used as the input for a machine learning algorithm which, once trained on the input data, can make predictions on unseen data. Such methods have been developed using support vector machines,[61–63] random forests,[64] and neural nets,[65] and the key factors involved in protein crystallization have been found to include iso-electric point, amino acid frequencies (particularly cysteine and histidine) and hydrophobicity. There may be parallels between protein and small-molecule crystallization propensity, particularly in the kinetic reasons for the inhibition of crystal growth.

1.1.6 Small-molecule crystallization propensity

Structure prediction and energy calculations do not answer two important questions: (i) will a material crystallize at all? (ii) what changes can be made to a molecule to modify the crystallization propensity?

Attempts have been made to engineer small molecules which do not crystallize. These are called molecular glasses[66] and have applications in drug formulations, foods and photo-voltaic cells.[67] Although empirical relationships have been made between glass formation and conformational flexibility, prevention of directional interactions and use of bulky groups,[68–70] the factors which affect glass formation are still little understood. Principal component analysis[71] and support vector machines[72] have been used to predict the crystallization tendency of these molecular glasses, and these studies have suggested that reduced number of rotatable bonds and a lower molecular weight are important factors for increasing crystallization tendency, while another model used parameters including number of hydrogen bond donors, lipophilicity, and the ratio of carbon to heteroatoms.[73] These observations are specific to these

materials and make no attempt to assess crystallization tendency from solution.

Observations of the crystallization behaviour of a group of acylanilides found that increased conformational flexibility resulted in reduced crystallization tendency and slower crystal growth, a finding that was attributed to the increased number of orientations in which a molecule could dock at a crystal surface.[74] A small study using random forest algorithms on the crystallization from solution of the same set of acylanilides found that number of rotatable bonds and the length of alkyl side chains were the most important factors affecting the crystallizability.[75] The impact of conformational flexibility on the crystallization tendency of alditols has been documented,[26] a study which indicated that the presence of multiple conformers in solution was an underlying cause of reduced crystallization tendency. A crude set of rules for the qualitative assessment of the crystallization of drug substances has subsequently been proposed, based on the analysis of data compiled from several sources.[6] These rules are based on the observation that molecules in these data sets with a molecular weight below 350 Daltons and fewer than 4 rotatable bonds will readily crystallize, and attains a success rate of 70-80% on data from the studies of molecular glasses.[71, 72] However, there has been no more complex classification model proposed which can be applied to the crystallization propensity from solution of a broader range of compounds.

1.2 Molecular representation

1.2.1 Molecular file formats

There are a number of molecular file formats which can read by cheminformatics packages in order to generate a computer representation of a molecule for further calculation, visualisation or description in terms of chemical features.

Chemical table file

Several formats for storing and transferring chemical structure information have been developed at Molecular Design Limited which have become commonly used by a wide variety of software.[76] For a single molecule, the Mol format is used, which consists of a connection table describing the structural relationships and properties of a collection of atoms. The file consists of a header followed by several blocks of information:

1. Counts line – contains information such as atom and bond counts, as well as the chiral flag setting and the connection table version.

2. Atom block – contains a line for each atom in the molecule, consisting of the atomic symbol followed by specifications of the coordinates, charge, stereochemistry and associated hydrogens.

3. Bond block – contains a line for each bond, which specifies the two atoms which are connected by it, the bond order, and any stereochemistry or topology associated with it.

4. Properties block – lines used to encode molecular properties such as charge or isotopes.

The Structure Data File (SDF) is an extension of the Mol file that contains this information for multiple molecules, with the addition of a stream-like section after the connection table. This allows the storage of additional data fields such as molecular properties or identifiers, such as in the example shown in Figure 1.5, where two additional types of identifier are given after the connection table.

These formats are useful as they can encode a large amount of complementary information in the data fields after the connection table. However, as a storage method these formats are quite inefficient due to the amount of redundant information that is kept (for example, each atom always requires a line of 69 ASCII characters to encode

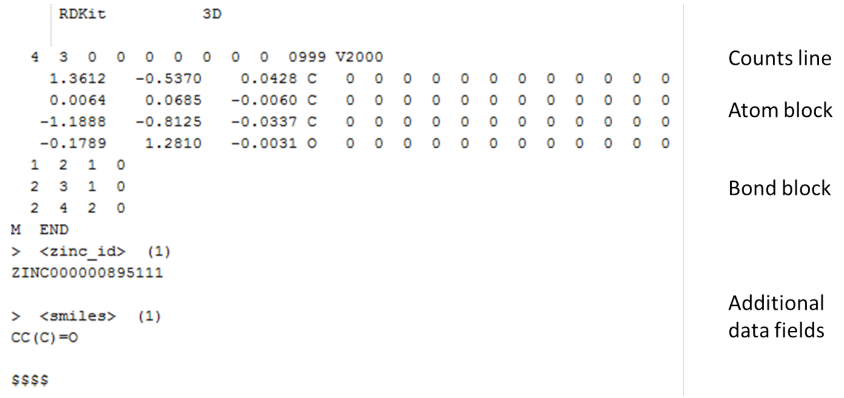


Figure 1.5 Example SDF file for acetone.

it). Additionally, it is not easy to match molecules using this cumbersome table-based format. These formats have been in existence for a long time, and so most software has the capability to read and manipulate them.

Line notation

Line notations are linear representations of a molecule that encode information about the connectivity and stereochemistry of a particular molecule, and can therefore be considered as a sort of chemical structural identifier. They are usually derived from a chemical table file such as the ones described above, and have the advantage of being more compact, since they contain the same connectivity information on a single line, although this does come with the loss of information such as 3-dimensional coordinates. For example, the acetone molecule represented by the SDF file in Figure 1.5 can be encoded in Simplified Molecular-Input Line-Entry System (SMILES) notation by the string 'CC(C)=O', a significant reduction in characters. Consequently, these line notations are widely used for database purposes, with the two most established of these being SMILES and the IUPAC International Chemical Identifier (InChI).[77]

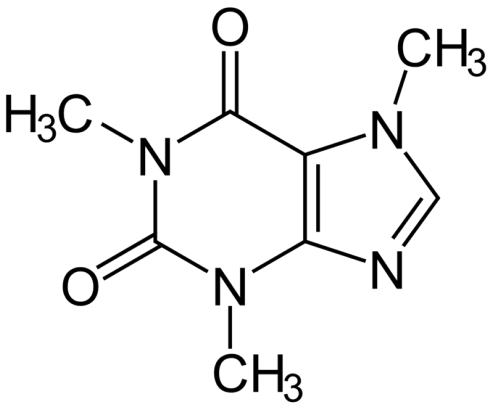
SMILES was developed in the 1980s with the aim of creating a notation which uniquely described the graph structure of the molecule using a specification which was user–friendly and easy to learn, but also machine–friendly.[78] However, there are many different ways of representing a molecule as a SMILES string depending on the starting atom chosen, so although the algorithm for producing a unique, canonical

SMILES for a particular molecule was published,[79, 80] the original implementation and software remain proprietary.[81] Consequently, other software developers independently developed differing unpublished algorithms, which causes problems when attempting to compare SMILES from different sources. A related notation, SMILES Arbitrary Target Specification (SMARTS), is used to specify substructural patterns within molecules.

More recently, an open-source standard has been developed to attempt to resolve this issue.[77] Since cheminformatics packages such as RDKit[82] contain their own algorithm for ensuring the same SMILES is obtained for a molecule regardless of the input method, the SMILES generated and used within a particular program are canonical with regards to that program, so SMILES from other sources could be converted into a canonical SMILES using the program of choice. The main advantage of SMILES is that the representation is easily human-readable and modifiable.

The InChI format was developed as a non-proprietary identifier which uses a well-defined atom numbering system to ensure that the same identifier would always be generated for the same molecule, which was unique to that molecule.[81] Aside from encoding the core parent structure of the molecule (the connectivity), InChI strings contain several layers which allow expression of the particular stereoisomeric, tautomeric, isotopically-substituted, charged form of that parent structure.

The various layers allow molecules to be expressed in varying levels of detail depending on the purpose. This can lead to very long InChI strings which are impossible to use with a traditional search engine, so the InChIKey was developed as a more compact, 27-character registry-lookup identifier, which can be used for searches and indexing databases. However, the structure cannot be directly deduced from the InChIKey; the source InChI string must be retrieved first, requiring a lookup table. Human-readability of the InChI string itself was considered a low priority when developing this notation, and as a result an InChI string is less easy to read and modify than the equivalent SMILES string, particularly for extremely long strings.



SMILES: CN1C=NC2=C1C(=O)N(C(=O)N2C)C

InChi:InChI=1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H,1-3H3

Figure 1.6 SMILES and InChI strings for caffeine.

An example of a molecule expressed as both a SMILES string and an InChI string is shown in Figure 1.6. SMILES strings were chosen for use in this project due to their compactness, human-readability and widespread use.

1.2.2 Descriptors

One of the most important factors affecting the performance of a learning method when applied to a cheminformatics situation is the chemical space that is used to define the problem.[83] The descriptors of each molecule define the basis vectors of this chemical space, with a feature vector of n linearly independent variables giving rise to an n-dimensional feature space. The values of the descriptors for a molecule define where it resides in this chemical space.

The type of molecular representation used to calculate a descriptor defines the dimensionality of the descriptor.[84] 0-dimensional descriptors are calculated from the composition of the molecule, such as integer counts of atom types. 1-dimensional descriptors are bulk properties of the material and include the calculated solubility,[85] molecular weight and functional group counts. 2-dimensional descriptors require the connectivity of the molecule to be known *e.g.* the number of rotatable bonds. Properties calculated from a known conformation of a molecule are 3-dimensional

descriptors, such as pharmacophore features, distribution of charges and radius of gyration.[86]

As a result, descriptor sets can be a combination of simple integer counts such as functional group counts, or more complex shape and connectivity based indices. Cheminformatics toolkits contain standard descriptor sets which are a combination of the above descriptors. Generally, the most common descriptors tend to be calculated from the hydrogen-suppressed 2-dimensional connectivity of the molecule. This means that no 3-dimensional information is incorporated into a basic model calculated from the standard descriptors.

1.2.3 VSA descriptors

Some important 1-dimensional bulk properties, such as molar refractivity (MR) and octanol-water partition coefficient (logP), are calculated by summing atomic contributions, which have been determined by fitting on an extensive training set.[87] These properties can be converted into a set of 2-dimensional descriptors by taking the atomic contributions instead.[88]

Each atom is assigned a van der Waals surface area (VSA) which is the spherical surface area not contained in any other atom. By summing the surface area with a physicochemical property contribution between particular limits of that property, a set of VSA descriptors can be established. These limits are ($-\infty$, -0.4, -0.2, 0, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, ∞) for logP and (0, 0.11, 0.26, 0.35, 0.39, 0.44, 0.485, 0.56, ∞) for MR. A similar approach can be applied to the atomic partial charges, as calculated by the Gasteiger (PEOE) method,[89] with limits of ($-\infty$, -0.3, -0.25, -0.20, -0.15, -0.10, -0.05, 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, ∞). Overall, this gives a 32-dimensional chemical space with descriptors SlogP VSA_{1-10}, SMR VSA_{1-8} and PEOE VSA_{1-14}, where SlogP VSA_1 is the van der Waals surface area of the molecule with logP in the range -∞ to -0.4, and so on. These descriptors have been successfully used to create a number of good quantitative structure-property relationships (QSPR) models

for properties including boiling point, water solubility and receptor class.[88]

1.2.4 Molecular quantum number descriptors

Molecular quantum numbers (MQNs) are a simple descriptor set which was developed to classify organic molecules in a manner analogous to the way the Periodic Table classifies elements.[90] The system consists of 42 integers belonging to one of 4 categories: atom counts, bond counts, polarity counts and topology counts. Since all of the descriptors are simple counts of features of a molecule, they can be calculated easily even with relatively little chemical knowledge. They have been used to provide searchable maps of databases including ZINC,[90] PubChem,[91] Drugbank[92] and GDB-13.[93] Principal component analysis of this descriptor set has shown that compounds with similar bioactivity in the Directory of Useful Decoys (DUD) dataset cluster together in MQN-space,[90] and the capability of this system to provide a simple distance measure which does not select for substructure similarity[94] means that this chemical space description could be useful for virtual screening since it can reveal non-trivial lead-hopping relationships.[95] However, the main issue with the MQN system is that there are many cases where several molecules occupy the same bin (*i.e.* they have the same value for every descriptor). These are termed MQN-isomers, and this can cause problems if there are two molecules of the same class which share the same descriptors, since there is no way to distinguish between the two.

1.2.5 Topological descriptors

The topological features encoded by the MQN system are quite simple and consist only of counts of the valency of the nodes (atoms) and the types of ring present (Table 1.1). Topological features can be represented in a more complex manner using the set of χ connectivity indices[96] and κ shape indices[97, 98] proposed by Kier and Hall, which have been used extensively in structure-property modelling.[99] These are implicit 3D descriptors, since they describe 3-dimensional aspects of the molecule

Table 1.1 Molecular quantum numbers.

	Atom counts		Polarity counts
1	carbon	20	H-bond acceptor sites
2	fluorine	21	H-bond acceptor atoms
3	chlorine	22	H-bond donor sites
4	bromine	23	H-bond donor atoms
5	iodine	24	negative charges
6	sulphur	25	positive charges
7	phosphorus		Topology counts
8	acyclic nitrogen	26	acyclic single valent nodes
9	cyclic nitrogen	27	acyclic divalent nodes
10	acyclic oxygen	28	acyclic trivalent nodes
11	cyclic oxygen	29	acyclic tetravalent nodes
12	heavy atoms	30	cyclic divalent nodes
	Bond counts	31	cyclic trivalent nodes
13	acyclic single bonds	32	cyclic tetravalent nodes
14	acyclic double bonds	33	3-membered rings
15	acyclic triple bonds	34	4-membered rings
16	cyclic single bonds	35	5-membered rings
17	cyclic double bonds	36	6-membered rings
18	cyclic triple bonds	37	7-membered rings
19	rotatable bonds	38	8-membered rings
		39	9-membered rings
		40	≥ 10-membered rings
		41	nodes shared by ≥2 rings
		42	edges shared by ≥2 rings

from a 2D representation.

The connectivity indices are calculated by incorporating weighted counts of substructure fragments into numerical indices, and encode information about size, atom identity, unsaturation, branching and cyclicity. The simplest of these is calculated by assigning a δ value to each atom, which is simply the "degree" of the atom (the number of bonded heavy atoms), which can be calculated by subtracting the number of bonded hydrogen atoms from the number of valence electrons involved in σ bonding.[100] The skeleton of the molecule is decomposed into subgraphs of length m, with a subgraph of order 0 being equivalent to the atoms, a first order subgraph

corresponding to one bond paths and so on. For each subgraph c of length m,

$$^{m}c_i = \prod_{k}^{m+1} (\delta_k)^{-0.5} \tag{1.2}$$

The corresponding connectivity index, $^{m}\chi$, is then the sum of these subgraph contributions, $\sum_i {}^{m}c_i$.

These connectivity indices do not differentiate between heteroatoms, so a modification to δ is required which takes into account the valency of the heteroatom.[101] For elements in the first row of the Periodic Table, the number of σ bonding electrons is replaced by Z^v, the total number of valence electrons, to give $\delta^v = Z^v - h$. This can be extended for elements in subsequent rows of the Periodic Table by including the total electron count, Z, which accounts for the shielding of the valence electrons by the inner electrons, to give $\delta_v = (Z^v - h)/(Z - Z^v - h)$. The valence connectivity indices are then calculated by $^{m}\chi^v = \sum_i {}^{m}c_i^v$, where $^{m}c_i^v = \prod_k^{m+1}(\delta_k^v)^{-0.5}$. For the zero order connectivity index, $^{0}\chi^v$, this simply becomes the summation

$$^{0}\chi^v = \sum_{i=1}^{n} (\delta^v)^{-0.5} \tag{1.3}$$

and this index has been found to correlate closely with molecular volume.[102]

The shape indices ($^{1}\kappa$, $^{2}\kappa$ and $^{3}\kappa$) relate the number of 1, 2 and 3 bond paths in a molecule to the numbers of these paths in reference molecules with minimum and maximum values. $^{1}\kappa$ encodes information about the atom count and the relative cyclicity,[98] while $^{2}\kappa$ captures information about the degree of branching and the relative spatial density[97] and $^{3}\kappa$ provides information about the centrality of the branching.[98] For example, $^{2}\kappa$ is calculated by multiplying the ratio of 2 bond paths in the molecule, $^{2}P_i$, with each of the minimum and maximum 2 bond paths ($^{2}P_{\min}$ and $^{2}P_{\max}$). This gives the formula $\frac{^{2}P_{\min}.{}^{2}P_{\max}}{(^{2}P_i)^2}$. Since $^{2}P_{\min}$ is equivalent to $(A-2)$ for an acyclic molecule with A atoms, and $^{2}P_{max}$ can be shown to be $\frac{(A-1)(A-2)}{2}$, analysing the limits of the equation shows that a linear molecule (where $^{2}P_i = {}^{2}P_{\min}$) provides a value equal to half the number of bonds. Consequently a scaling of two is applied to

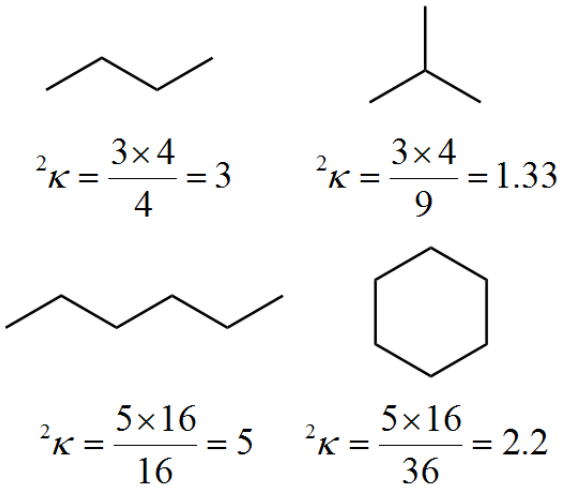


Figure 1.7 Example ${}^2\kappa$ values for a series of small hydrocarbons.

give the final equation:

$$ {}^2\kappa = \frac{(A-1)(A-2)^2}{({}^2P)^2} \tag{1.4} $$

As can be seen from Figure 1.7, increased size, decreased branching and less cyclicity give higher ${}^2\kappa$ values. Since heteroatoms and C atoms with hybridisation other than sp^3 contribute differently to the shape of the molecule because of their differing covalent radii, the indices should be adjusted to account for this.[103] This adjustment is achieved by modifying the atom count, A; for each atom the new count becomes $1+\alpha_x$, where $\alpha_x = \frac{r_x}{r_{\mathrm{Csp^3}}} - 1$ and r_x and $r_{\mathrm{Csp^3}}$ are the covalent radii of atom x and a $\mathrm{sp^3}$-hybridised carbon atom respectively. In the equations for the κ indices, the value of A can be substituted by $A+\alpha$, where $\alpha = \sum \alpha_x$, to create a heteroatom-weighted index.

1.2.6 Flexibility descriptors

Molecular flexibility is an important concept in cheminformatics, which can be divided into two distinct types: thermodynamic conformational flexibility, in which the molecule exists in a number of conformations out of all the possible conformations that might be generated for that structure, and kinetic conformational flexibility,

which is determined by the rate of interconversion between these conformers.[104] This information is difficult to capture from the 2-dimensional representation of the molecule, and attempts to define a single descriptor for a molecule based on conformational flexibility have been restricted to specific subsets of molecules such as alkanes[104] and endomorphins.[105]

The simplest way of establishing the flexibility of a molecule is the rotatable bond count.[106] Definitions can vary depending on the strictness, but methyl groups and amide bonds are usually considered non-rotatable. Other descriptors have been developed which rely solely on the connectivity of the molecule. One of the earliest and simplest of these, the path length flexibility index F_k, accounts only for branching.[106] This relates the number of 3-bond paths, ${}^3P$, to the length of the longest path in the molecule, L, using the equation $F_k = \frac{L}{1-1/{}^3P}$. A longer chain gives a higher F_k, while increased branching gives a lower value of F_k. If there are no 3-bond paths in the molecule then F_k is set to be zero. However, this index is very simplistic and fails to take account of other important factors affecting the flexibility.

A more comprehensive descriptor, the Kier flexibility index, was designed based on the assumption that a perfectly flexible molecule is an infinite chain of sp^3-hybridised carbon atoms. This is mitigated in other molecules by several factors: number of atoms, cyclicity, branching and atoms with smaller covalent radii. Since ${}^1\kappa$ contains information about number of atoms and cyclicity and ${}^2\kappa$ contains the branching information, the versions of these descriptors which are modified to account for heteroatom covalent radius can be combined to create a flexibility index.[107] These modified κ_α values can be combined and normalised by the atom count, A, to give a single flexibility index with the following equation:

$$\Phi = \frac{{}^1\kappa_\alpha \cdot {}^2\kappa_\alpha}{A} \tag{1.5}$$

However, there are some consequences of this description of molecular flexibility, which include the overestimation of flexibility of conjugated systems and substituted

rings.[108]

1.3 Machine learning/classification methods

Machine learning algorithms are a type of artificial intelligence that can learn without being explicitly programmed,[109] and are being used increasingly extensively to tackle problems which a human being cannot process. This is particularly true in circumstances where there are multiple relationships between a large number of features in a large dataset.[110] In these cases, it is preferable to use computational methods to extract this information in a quicker and more reliable way.

In terms of their use in a computational chemistry context, they have already been used to address a wide array of problems, modelling QSPRs of small molecules to predict melting points,[111] solubility[112], heat capacity[113] and lower flammability limit.[114]

Algorithms can be split into two distinct types:

1. Supervised, where the input data has an attribute associated with it which the algorithm attempts to reproduce. There are two further subsets of supervised learning.

   (a) Classification — if the target attribute is a class, meaning that each input vector belongs to a discrete category, the problem is a classification task.

   (b) Regression — if the attribute is a continuous variable, a regression algorithm is required.

2. Unsupervised, in which case the input data has no associated target values. These are useful for attempting to extract previously unknown patterns from the data, either by clustering the input points, in the hope of discovering similar examples,[115] or by estimating the distribution of the data within the input space.

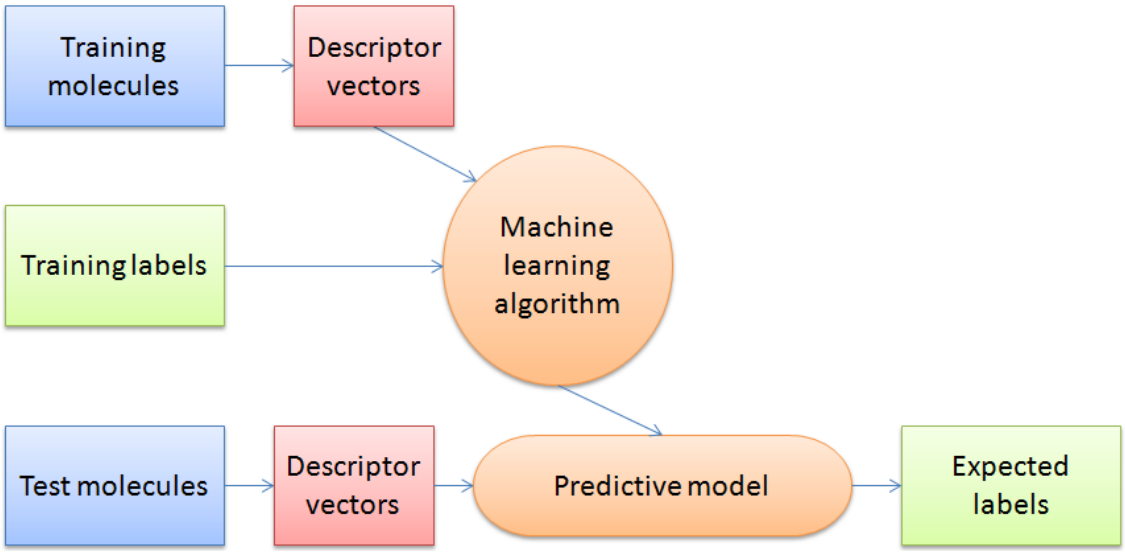


Figure 1.8 Flowchart of the general method for creating and testing a predictive model using a machine learning algorithm.

In the case of the crystallization prediction problem, the goal is to attempt to distinguish between crystalline and non-crystalline classes of molecules based on the known labels of these molecules in the input data. Therefore a supervised classification algorithm is required.

There are several key steps involved in creating a predictive model using a supervised machine learning algorithm, as demonstrated in Figure 1.8. The algorithm requires a set of input molecules, each of which has a label associated with it, which in this case would be “crystalline” or “non-crystalline”. Each input data point is represented as a feature vector, which is a list of calculated numerical attributes or descriptors for that particular molecule. Each feature vector contains the same attributes. The algorithm uses these feature vectors and their associated labels to infer a set of rules for distinguishing between the two classes, and these rules are the predictive model. The model is then used to assign labels to a new, unseen set of molecules, based on their feature vectors. These labels can be compared to their actual labels to obtain a measure of the predictive accuracy of the model.

There are several important considerations to make when developing a model.

1. Selection of input data — the input data needs to be extensive enough to be

able to infer a model which generalises well, and the classes must be as reliable as possible

2. Definition of training and test set — the external test validation set is used to determine the performance of the model, so it is important that this is completely separate to the training set used to develop the model, to avoid artificially inflating the predictive accuracy.

3. Choice of feature vector — the input data points must be represented in an appropriate manner, with enough descriptors relevant to the problem while reducing the number of irrelevant or confusing descriptors.

4. Choice of algorithm — there are a variety of algorithms that could be used, with various parameters which may need to be optimised for each one. Often, the determination of the best algorithm and parameters is done empirically.

The first two points will be discussed in more detail in Chapter 2, while examples of typical elements of the feature vector were discussed in Section 1.2. Here we will focus on the background to the algorithms.

1.3.1 Decision Trees

Decision trees are a method of representing the rules that underly a dataset using a hierarchical, sequential set of structures that use the feature values to recursively separate the data into their classes.[116] They essentially split the feature space into regions, and assign a class label to each region based on the samples contained within, which are used to predict on unseen molecules within those regions. The tree consists of internal nodes, leaf nodes, and splits. The general procedure for generating a tree is as follows:

1. At internal node *t*, if the node is pure (that is, all training data points belong to the same class), designate this to be a leaf node.

2. If node t is impure, compute all of the possible splits of each attribute and score them according to some goodness function.

3. Use the best possible split to partition the training data into child nodes.

4. Repeat the process (with child nodes becoming internal nodes) until all child nodes are pure, or no possible improvement is obtained.

Common scoring functions for evaluating the best split at an internal node include entropy and the gini impurity.[117] If the proportion of class k observations in node m is $p_{m,k}$, then for that node the entropy can be calculated by

$$-\sum_k p_{m,k} \log(p_{m,k}) \tag{1.6}$$

and the gini impurity by

$$\sum_k p_{m,k}(1 - p_{m,k}) \tag{1.7}$$

The higher the value of these coefficients, the more impure the node is, with the maximum occurring when all classes have equal probability. An example of a simple decision tree is given in Figure 1.9.

There are a number of advantages to using decision trees, mainly that the rules inferred from the training data are simple, human-readable and can be visualised easily. In addition, no normalisation of the input features is required, both numerical and categorical features can be used, and they can handle multi-output problems.

The biggest disadvantage of decision trees is that they are prone to over-fitting, since by default they try to correctly classify every input training data point. This leads to a model which generalises poorly, particularly in cases where there are many attributes in the feature vector and some leaf nodes only contain a small number of samples. These problems can be reduced by artificially specifying a minimum number of samples to be present at a leaf node, or specifying a maximum depth of tree, but neither of these solutions removes this problem entirely. Another alternative

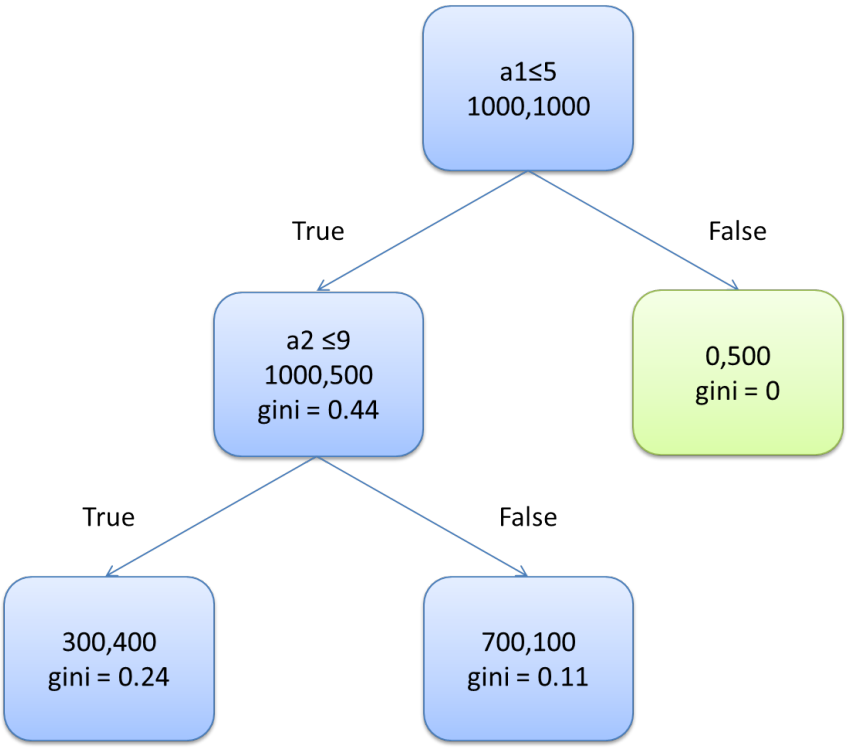


Figure 1.9 Example decision tree. With 1000 samples in each class, the best split uses attribute 1 to separate 500 of class 1 into a child leaf node for values greater than 5, with the remainder forming an impure child internal node, which is split further by attribute 2. These child nodes are also internal nodes, which can be split further by other attributes, or other splits of attributes 1 and 2. The gini impurity measures are also displayed for each node.

is to "prune" the tree to remove branches that do not contribute significantly towards the generalisation accuracy, but this requires construction of the complete tree first.

Another disadvantage is that trees can be biased towards a class with a greater number of samples, and the only way to remove this bias is to balance the classes beforehand.

Furthermore, the algorithms used to generate decision trees involve making locally optimal decisions, which is not guaranteed to provide the globally optimal tree. Even if the tree is globally optimal, it can still be an overcomplicated decision surface if diagonal partitioning is required, due to the reliance of the decision surface on hyper-rectangular regions, which is a result of the decision boundaries being orthog-

onal to the axes.[118]

1.3.2 Random Forest

The random forest algorithm is an extension of the decision tree method which was developed by Breiman.[119] This is an ensemble classifier, which means that the weighted predictions of many classifiers are combined to create the overall predictive model. Such methods have been shown to often outperform any single predictor.[120] The reason for this is threefold:

1. Statistical — A learning algorithm is a search of hypothesis space for a hypothesis which performs best. If the amount of training data is small relative to the size of the hypothesis space, many single classifiers can give very similar accuracy. By averaging the predictions of all of these, the chance of choosing the wrong classifier is reduced.

2. Computational — Many algorithms employ an approach which can lead to finding local optima rather than the global optimum. Combining classifiers which have begun from different starting points may find a better approximation to the true function.

3. Representational — It may not be possible to adequately represent the true function by any single classifier. Averaging a set of weighted classifiers may expand the set of representable functions.

In this case, the overall classifier is a collection, or "forest", of decision tree classifiers. Each tree is trained using a subset of the training dataset which is randomly drawn with replacement (bagging). This generates n new training subsets which are used to build n decision trees. However, each tree is grown by using the best split among a randomly selected subset of the attributes in the feature vector, rather than using the best split of all features (as is the case for a single decision tree), an idea that was first proposed by Amit and Geman.[121]

Each individual tree classifier is fully grown without pruning. The individual classifiers are combined, either by allowing each classifier to "vote" for a class for each data point and assigning the class based on the one with the most votes, or by averaging the probabilistic predictions that each classifier produces.

A result of the randomness of this method is that the bias (the true error of the best classifier) increases. However, the averaging causes a decrease in variance (the error of the trained classifier with respect to the best one in that concept class), which more than compensates for the increased bias and therefore yields a better model.

Breiman shows that due to the Strong Law of Large Numbers, the forests always converge, so overfitting is not a problem.[119] This means that adding further trees only produces a limiting value of the generalisation error.

Another advantage of random forests is that they contain a large amount of information about the relationship between the variables and the class labels. Although there is some loss of readability with respect to decision trees, information can be extracted about the contribution of each feature to the classification. This involves using the remaining training samples that are not used to generate a particular tree (the out-of-bag data). After each tree has been built, the values of the descriptor of interest for points in the out-of-bag sample are randomly permuted. The new class labels assigned by the tree as a result of this are compared with the class labels determined using the unchanged descriptors to gain a measure of the increase in classification error. When averaged across all trees, a feature which causes a larger increase in misclassification is more important for making the classification.[119]

1.3.3 Support vector machines

Support vector machine (SVM) algorithms were originally developed in the mid 1990s,[122] and have been gaining popularity ever since due to having several attractive features and promising empirical performance.[123] The SVM method revolves around the use of a hyperplane which separates the classes in the descriptor space. There will exist

many different hyperplanes, but the goal is to produce the separating hyperplane which generalises well — meaning it is successful on unseen examples. By maximising the margin between the hyperplane and the nearest point either side of it, the upper bound on the expected generalisation error can be reduced.[110]

Linear SVM

Considering a two class problem, for training vectors $\boldsymbol{x_i}$ with labels y_i where y denotes the two classes and is either 1 or -1, there exists a hyperplane $\boldsymbol{w_0} \cdot \boldsymbol{x} + b_0 = 0$ which separates the classes, where $\boldsymbol{w_0}$ is the optimal weight vector and b_0 is the bias scalar. The constraints on the minimisation are that all points belonging to class 1 satisfy the equation $\boldsymbol{w} \cdot \boldsymbol{x_i} + b \geq 1$ and all those belonging to class -1 satisfy the equation $\boldsymbol{w} \cdot \boldsymbol{x_i} + b \leq -1$. These constraints can also be expressed in a single equation as $y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1$. The separation is optimal if there are no classification errors, and the distance between the nearest training point in each class and the hyperplane is maximised.

When the data is perfectly linearly separable, as in the example shown in Figure 1.10, this optimal hyperplane can be found by minimising the squared norm of the hyperplane, $\Phi(\omega) = \frac{1}{2}\|\boldsymbol{w}\|^2$ with the constraint above. This is a convex quadratic programming problem which can be solved by assigning each training point a Lagrange multiplier, α_i, to give the Lagrangian

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_i \alpha_i y_i(\boldsymbol{x_i} \cdot \boldsymbol{w} + b) + \sum_i \alpha_i \tag{1.8}$$

which must be minimised with respect to $\boldsymbol{w}$ and b. Classification of an unknown test vector $\boldsymbol{x}$ is achieved by taking the sign of $f(x) = \boldsymbol{w} \cdot \boldsymbol{x} + b$.

The data points lying on the margin of the optimal hyperplane (where $y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) = 1$) are called the support vectors, and all other data points can be ignored, meaning that the SVM can be used to summarise the information in the dataset. The optimal hyperplane $\boldsymbol{w_0}$ can be expressed as a linear combination of training vectors,

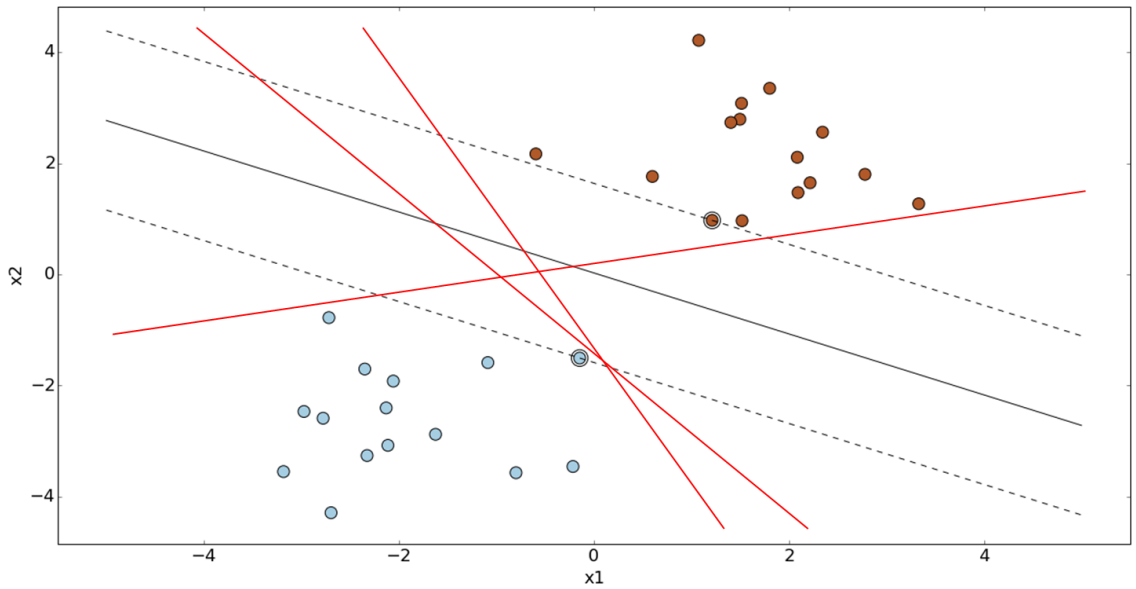


Figure 1.10 Example of linearly separable two class problem with optimal (black) hyperplane and margins (dashed lines) The support vectors lying on the margins are circled. The red lines denote other separating hyperplanes which are not the optimal one.

$\boldsymbol{w_0} = \sum_i y_i \alpha_i \boldsymbol{x_i}$, where $\alpha_i \geq 0$ (which is only true for support vectors). Since the number of support vectors is usually small, the model complexity is unaffected by the number of features in the training data, meaning that SVM works well when there is a large number of features relative to the number of training data points.

In practice, the training data are rarely linearly separable without errors, as highlighted in Figure 1.11, and so a "hard" margin such as the one described above is unsuitable. However, by introducing a cost function associated with the misclassification of training points, a "soft" margin can be used instead.[124] This function, which must also be minimised, takes the form $F(\xi) = \sum_i \xi_i$, where ξ_i are non-negative measures of the classification errors, and the constraint on the minimisation becomes $y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1 - \xi_i$. The function to be minimised becomes $\Phi(\omega) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i$, where C is a parameter which needs to be determined, but which introduces additional capacity control into the classifier.[123] This minimisation determines the hyperplane that minimises the number of errors in the training set and separates the rest of the elements with maximal margin. The magnitudes of the coefficients of this

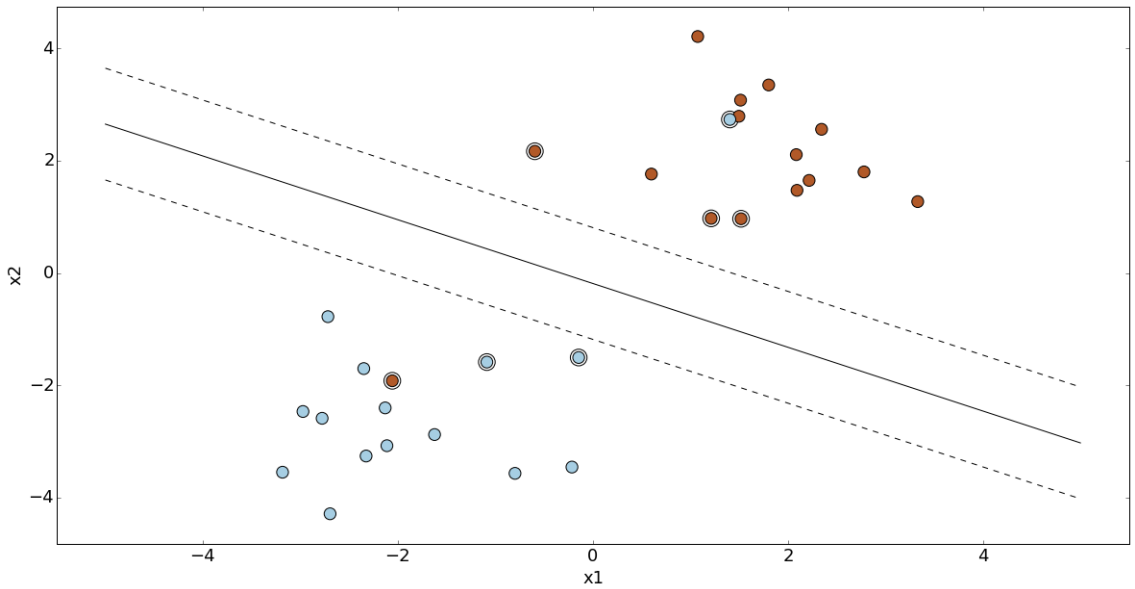


Figure 1.11 Example of linearly non-separable two class problem with optimal (black) hyperplane and margins (dashed lines) The support vectors lying near the margins are circled. Since the margin is now soft, the support vectors no longer lie on the margins.

plane determine the importance of each feature in making the classification, while the signs of the coefficients indicate the class with which that coefficient is correlated.

However, this still assumes that the data are linearly separable in the input space, after accounting for errors. This is often not the case, and so a more complex decision function is required.

Kernel SVM

One solution to combat linearly inseparable data is to transform the data from the n-dimensional input space into a higher, N-dimensional feature space using a vector function $\phi : \mathrm{Re}^n \rightarrow \mathrm{Re}^N$. If an appropriate non-linear transformation of sufficient dimensionality can be found, there should always be an optimal linear separation which can be achieved in this higher-dimensional space, which corresponds to a non-linear separation in the input space.

An N-dimensional linear separator is constructed in the feature space for transformed vectors $\phi(\boldsymbol{x_i})$. To classify a new vector x, the vector must be transformed into the feature space to obtain $\phi(\boldsymbol{x})$, before taking the sign of $f(x) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b$ in

an equivalent manner to before. Since w can be expressed as a linear combination of support vectors ($w = \sum_i y_i \alpha_i \phi(\boldsymbol{x_i})$), and dot products are linear, the classification function can be rearranged and $\boldsymbol{w}$ replaced to get

$$f(x) = \phi(\boldsymbol{x}) \cdot \boldsymbol{w} + b = \sum_i y_i \alpha_i \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x_i}) + b \tag{1.9}$$

which depends only on dot products in the feature space.

If there exist kernel functions, K, of the form $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i}) \cdot \phi(\boldsymbol{x_j})$, then K can be used in the algorithm without ever needing to determine ϕ, allowing the dot products to be calculated directly in the feature space. After the hyperplane has been determined, the decision surface is of the form

$$f(x) = \sum_i y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x_i}) \tag{1.10}$$

with $\boldsymbol{x_i}$ being the transformed support vector in the feature space, and α_i the weight of that support vector in the feature space. The kernel function is then used to map unknown vectors into the feature space for classification.

Some common kernel functions include:

1. Polynomial: $K(\boldsymbol{x}, \boldsymbol{x_i}) = (\boldsymbol{x} \cdot \boldsymbol{x_i} + 1)^p$
2. Exponential radial basis function: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \exp(-\gamma |\boldsymbol{x} - \boldsymbol{x_i}|)$
3. Gaussian radial basis function: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \exp(-\gamma |\boldsymbol{x} - \boldsymbol{x_i}|^2)$
4. Sigmoid: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \tanh(\kappa \boldsymbol{x} \cdot \boldsymbol{x_i} - \delta)^p$

Radial basis function (RBF) kernels have received the most attention, particularly Gaussian RBFs, which have been shown to outperform classical RBFs.[125] They are also desirable due to having fewer numerical difficulties than polynomial kernels (which may go to infinity or zero when the degree is high) and sigmoid kernels (which are not always valid for some parameter values).[126] As a result, the Gaussian RBF

kernel was chosen as the main kernel to use, and will be referred to simply as the RBF kernel from now on.

In the case of the RBF kernel, the C parameter trades off misclassification of training points against smoothness of the decision surface—a low C misclassifies more points but has a smoother decision surface, while a higher C tries to classify all training points correctly by including more support vectors, leading to a more complex decision surface.

In addition to the C penalty parameter, the RBF kernel requires the γ parameter to be specified, which denotes the radius of influence of each training point. A large value of γ denotes a small radius of influence, which can lead to overfitting since new instances would have to lie on the support vector for classification to be successful. A value of γ which is too low would lead to underfitting, since each point would have a large radius of influence, and would fail to capture the complexity of the decision surface effectively, instead performing more like a linear classifier. The effect of varying these two parameters is displayed in Figure 1.12.

Practical considerations

1. Categorical features — SVM cannot cope with these directly as each feature vector is a vector of real numbers, so categorical features must be converted into n numbers for an n-category feature.

2. Scaling — Attributes with a greater numeric range can dominate over those with a smaller numeric range, so the attributes must be scaled before use with the SVM.

3. Parameter choice — Parameters such as C and gamma have sensible default values, but the only way to determine the best parameters for a particular problem is to empirically determine them using the training set, as described in Section 2.3.2.

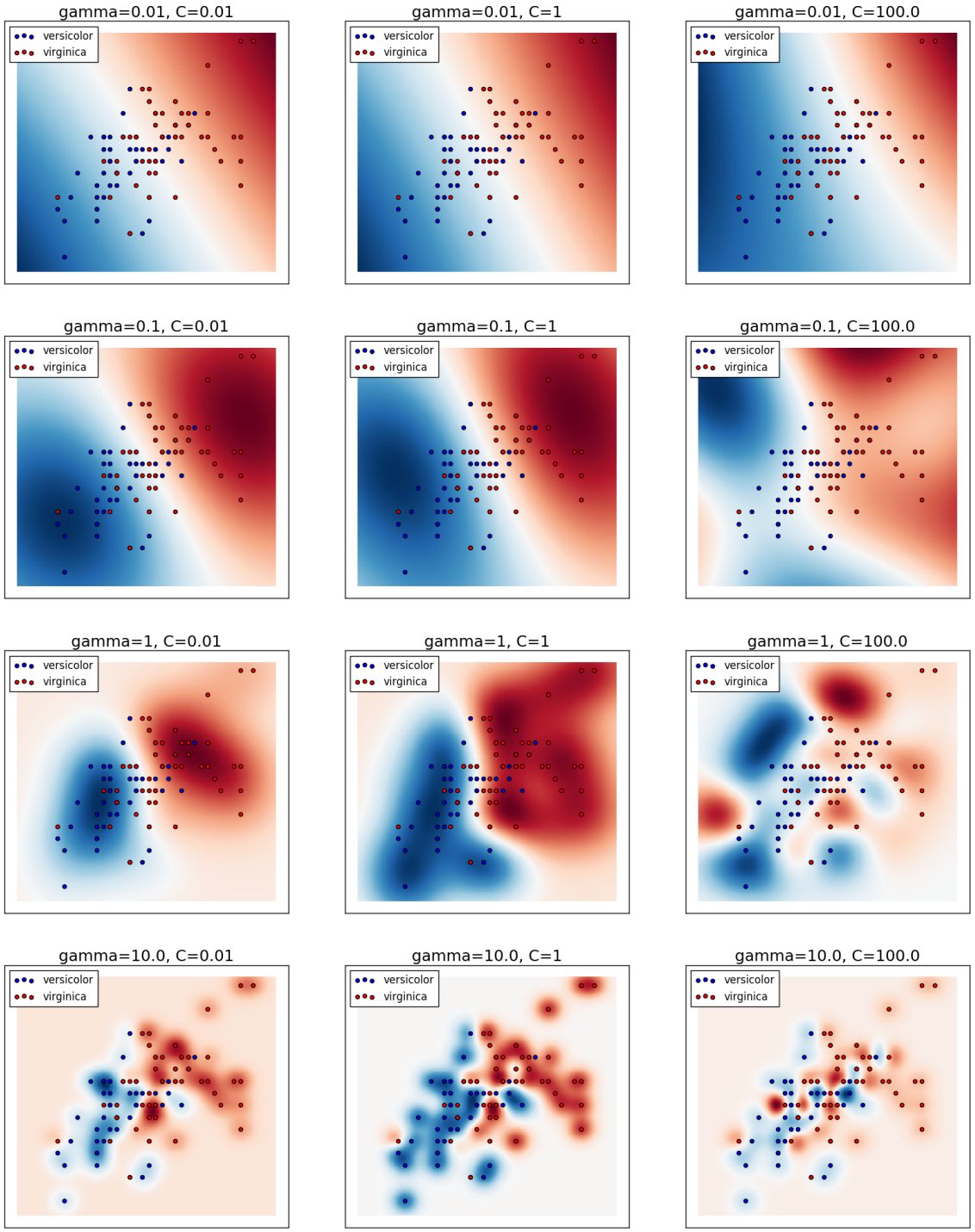


Figure 1.12 Visualisation of the effect of the parameters on the SVM decision surface for a RBF kernel, using a two-dimensional feature vector for two of the classes in the standard iris dataset, and the colour scale indicating the probability of a data point in that region belonging to the particular class. In this case, blue regions indicate the versicolor iris and red indicate the virginica iris.

4. Unbalanced classes — if there are more of one class than another, then either the sets should be balanced before use with the algorithm, or the C parameter can be set to different values for each class.

5. Interpretation — the implicit nature of the kernel mapping results in a "black-box model" - that is, it is not possible to directly extract information about the most important features involved in the classification. Other methods are required to explain the workings of the algorithm indirectly, as discussed in Section 2.3.4.

6. Feature selection — the SVM algorithm is generally robust to features that are irrelevant (provide no extra information to the algorithm) or redundant (provide information that is already provided by other features). However, selection of important features can in some cases improve the predictive accuracy,[127] but is most useful for improving the speed of the algorithm by reducing the dimensionality of the feature space.[118]

1.3.4 Performance metrics

There are a number of methods to evaluate the performance of a predictive model, with varying degrees of appropriateness. In practice, a combination of these is required to give a full picture of the effectiveness of the model.

Percentage accuracy

The most common way to evaluate the performance of a machine learning algorithm is to use the percentage accuracy of the predictions that the model makes on the test set, which is simply the ratio of correct predictions to total predictions made. However, the predictive accuracy can be misleading, particularly in cases where the class sizes are imbalanced. For example, a test set containing 900 of class 1 and 100 of class 2 could achieve a seemingly high accuracy of 90% simply by predicting that every test data point belongs to class 1.[128]

A similar problem can also occur in cases where the classes are balanced in terms of dataset size, but the model is much better at prediction on one class than another. The overall predictive accuracy will be an average of this, but the figure will hide the misclassification imbalance.

Confusion matrix

A confusion matrix provides an enhanced level of information compared to the overall predictive accuracy by showing the accuracy within each class.

Table 1.2 Example confusion matrix.

		Predicted label	
		Class 0	Class 1
Actual label	Class 0	True negative (TN)	False negative (FN)
	Class 1	False positive (FP)	True positive (TP)

Table 1.2 shows that the columns of the confusion matrix contain the instances within a predicted class while the rows show the instances within the actual class. The diagonal elements of the confusion matrix are the successful predictions, and the off-diagonal elements are the unsuccessful predictions. The advantage of a confusion matrix is that it provides a quick method of identifying if a model is confusing one class for another, by showing any imbalance in the classification errors.

The overall predictive accuracy can be calculated from this table, but there are also other metrics that can be extracted from the confusion matrix:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1.11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.13}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{1.14}$$

Precision is the ability of the model to ensure that predictions for a particular class are correct (*i.e.* to ensure that as few negative results are incorrectly classed as positive ones), and is also known as the positive predictive value. Recall is the ability of the classifier to correctly identify all positive values, and is also called the true positive rate or sensitivity. The specificity is the ability to correctly identify all negative values, and is also known as the true negative rate.

The precision and recall can be combined into a single score, the F1 score, which is the weighted harmonic mean of these two metrics and is given by

$$\text{F1 score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1.15}$$

When the two numbers are similar, the F1 score is approximately the average.

ROC curve

Receiver operating characteristic (ROC) curves are commonly used to provide a measure of the classification ability of a model.[129, 130] They take advantage of the ability of many classifiers to output a probability or score for each data point, representing the degree to which that point belongs to a class. The curve is generated by ranking the molecules in descending order of the probability of the molecule being a positive result (as calculated from the algorithm). A score threshold is defined above which any instance is expected to be a positive result. The first point is generated using a threshold score of $+\infty$, above which no points are positive instances, yielding the point (0,0). The threshold is then lowered to include a new instance each time. If the actual label of this instance is positive, then this is a true positive result, whereas if the actual label is negative this is a false positive. The true positive rate is then plotted against the false positive rate to produce a step function which tends to a curve as the number of data points approaches infinity.[131] The area under the curve (AUC)

provides a measure of the ability of the model to rank the positive results relative to the negative ones.

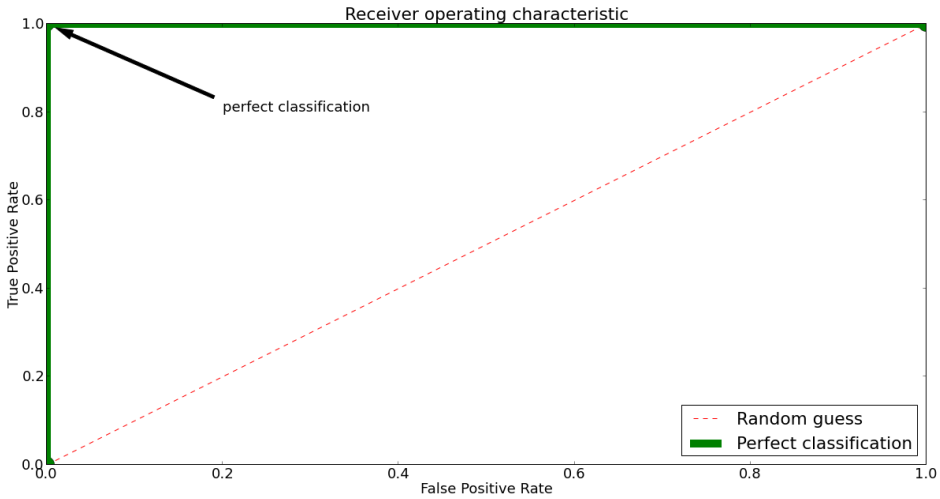


Figure 1.13 Example of a ROC curve showing perfect and random classification.

A perfect predictive model would generate a ROC curve which would follow the y axis from the origin to (0,1), then follow the line $y = 1$ to (1,1), denoting that all true positives are encountered at the top of the list, followed by all false positives, and giving an AUC of 1. A random model would follow the line $y = x$, that is that true positives and false positives are encountered at an equal rate on descending the list. This would give an AUC of 0.5. The random model is the worst model that can be obtained, since if a model performs worse than this, the predictions can simply be reversed to obtain a model that is better than random.

Learning curve

There are two types of error associated with machine learning algorithms: bias and variance. The bias of the algorithm is the true error of the best classifier in that class (*i.e.* the best performing linear separator or random forest). If the class of algorithms cannot model the data effectively, the bias will be high, and this leads to underfitting of the data. This problem cannot be remedied simply by inclusion of more training data.

The variance of the algorithm is the error of the classifier being used with respect to the best possible classifier in that class. The variance increases with increasing complexity of the classifier, and leads to overfitting. Variance can be decreased by increasing the amount of training data.

Both of these sources of error can be identified by plotting a learning curve, taking random subsets of the training set and using these subsets to build models for use on the test set. For each subset, the training accuracy (accuracy of the model on the training set—which is theoretically the best possible accuracy that can be obtained by the classifier) and test accuracy (accuracy on the test set) are plotted. A model with high bias will have a low training accuracy, while a model with high variance will have a low test accuracy relative to the training accuracy.

1.4 Powder diffraction

Although SXRD provides the most comprehensive information about a crystal structure, there are many real-life situations where the system under study is polycrystalline and often, for example in the case of multicomponent battery cathode materials, intrinsically heterogeneous and multiphase. In this thesis, powder diffraction is the technique of choice when the crystallite size is less than 5 µm and cannot be used for single crystal analysis. Indeed, there are as many orders of magnitude between 10 nm and 3 µm (the range of powder diffraction) as there are between 3 µm and 1 mm (the range of SXRD).

The underlying diffraction physics for powder diffraction is identical to SXRD. Each small crystallite in a polycrystalline sample diffracts in exactly the same way as the multi-micron-sized single crystal in a SXRD experiment. The diffraction pattern, therefore, for a small crystallite in a powder sample corresponds to an ordered array of spots arranged on a reciprocal lattice that conforms to the space group and crystal lattice of the material. The polycrystalline nature of the material, however, means that with each crystallite oriented in a different direction, the resulting diffraction pattern

in three dimensions of reciprocal space is a series of concentric spheres each with a reciprocal space radius that is determined by the length d* of the reciprocal lattice vector of each (h,k,l) reflection. Thus

$$d^{*2} = (\mathrm{h}a^* + \mathrm{k}b^* + \mathrm{l}c^*)^2 \tag{1.16}$$

The three dimensions of the reciprocal lattice are condensed onto the single dimension of a powder diffraction pattern. It is important, however, to note that it is the three dimensions of the reciprocal lattice and not reciprocal space that is compressed, so (h,k,l) directional information is retained, in contrast to liquid and amorphous diffraction and the use of the pair distribution function. The powder diffraction pattern consequently consists of Bragg peaks in which there are systematic overlaps, and also accidental overlap of neighbouring peaks from different crystallographic planes.

As with SXRD, the distribution of the electron density (and therefore the atomic positions) can be determined from the intensity of the peaks, while the peak positions provide information about the unit cell parameters. Peak overlap obscures individual diffraction intensities, which contributes to the "loss" of data and limits the size and complexity of the structures that can be solved.[132] The use of high resolution data, by using a shorter wavelength of radiation to collect the diffraction pattern, is one strategy to reduce the extent of peak overlap. Nevertheless, methods for the routine structure determination of molecular crystal structures have been developed,[133] ensuring that as of 2013 there were already over 2700 structures present in the CSD that were determined from powder diffraction studies.[134]

1.4.1 Indexing

The first step in the crystal structure solution from a powder diffraction pattern involves the determination of the unit cell parameters. These are the cell lengths a, b and c, and the angles α, β and γ. Indexing requires the positions of the Bragg

peaks to be accurately identified, which can be a challenge at shorter d-spacings because of the overlapping peaks. Best practice involves peak-fitting which, in high resolution synchrotron X-ray powder diffraction, involves the use of an asymmetry-corrected Voigt peak shape to fit the peaks.[135] The list of peaks is used as the input for an autoindexing program such as DICVOL91,[136] which performs an exhaustive search of all the possible unit cells within the given crystal system. In addition to the unit cell dimensions, the hkl index of each peak is determined. The observed hkl peaks can be compared to the expected ones for that particular system to identify systematic absences, which give information about the space group of the structure. For example, DASH determines the probability of each particular extinction symbol being represented by the peaks present, allowing the space group to be identified.

1.4.2 Pawley analysis

Refinement of the unit cell parameters against the diffraction pattern in the absence of a structural model is achieved using Pawley analysis.[137] The powder pattern is fitted by allowing parameters such as the lattice parameters, background, reflection intensities and peak shapes to vary in a least squares procedure. This allows the profile parameters to be obtained, as well as a list of correlated integrated reflection intensities, and is used to confirm the unit cell and crystal system obtained from indexing.

The next step, space group determination, involves a Pawley analysis assuming a space group with no systematic absences. The list of intensities which is obtained is then used to rank the extinction symbols (and therefore space groups) of the particular crystal system according to the probability that they explain the data. This provides a probabilistic space group determination, and such rankings can be achieved using programs like ExtSym.[138]

Additionally, the extracted intensities provide a set of structure factor magnitudes, $|F_{hkl}|$ which is used directly in the structure solution rather than the full diffrac-

tion pattern, providing a speed advantage. The weighted profile R factor, R^2_{wp}, is the sum of the squared differences between the observed and calculated intensities normalised by the weighted observed intensities, and is given by

$$R^2_{\mathrm{wp}} = \frac{\sum_i w_i \left(y_i^{\mathrm{obs}} - y_i^{\mathrm{calc}} \right)^2}{\sum_i w_i \left(y_i^{\mathrm{obs}} \right)^2} \tag{1.17}$$

where y_i^{obs} and y_i^{calc} are the observed and calculated intensities, and w_i is the statistical weight. The expected profile R factor, R^2_{exp} statistically represents the best possible R_{wp}, the case where every y_i^{obs} is accurately predicted, and is given by

$$R^2_{\mathrm{exp}} = \frac{N - P + C}{\sum_i w_i \left(y_i^{\mathrm{obs}} \right)^2} \tag{1.18}$$

where N is the number of data points, P is the number of parameters and C is the number of parameter constraints.

The χ^2 value, the measure of the quality of the fit, is given by $R^2_{\mathrm{wp}}/R^2_{\mathrm{exp}}$, and provides a benchmark against which the quality of the final structure determination is assessed.

1.4.3 Structure solution

Although single crystal methods such as direct methods or the Patterson method can be used to determine the structures of materials when the number of atoms in the asymmetric unit is small, this requires the number of overlapping reflections to be small. This is not usually possible as a result of the loss of three-dimensional information relative to a single crystal diffraction pattern, and is compounded by the sharp decrease in diffracted intensity with scattering angle for organic compounds, which means that peaks are not observed to high enough resolution. The structure factor quality can be improved using the differential thermal expansion method by varying the sample temperature during data collection.[139] This addresses the prob-

lem of reflection overlap by taking advantage of the anisotropic lattice expansion that most low-symmetry crystal structures exhibit with temperature change, however, a large amount of homogeneous sample is required along with specialised sample changing equipment, so such data collection strategies are impractical for routine organic crystal structure determination.

Usually direct space methods are employed, which requires *a priori* information provided by an initial molecular model. A global optimisation method is then used to find the crystal structure that best matches the data by varying the degrees of freedom of the initial molecular model, namely the position, orientation and flexible torsion angles. The most common global optimisation method employed is simulated annealing, which is attractive because of the relatively small number of algorithm control variables which can be set automatically without any requirement for user input, and has been found to be successful in many cases.[140]

In a simulated annealing run, the algorithm begins with random molecular position, orientation and torsion angles. The structure factors for this structure are calculated to create a simulated powder pattern, which is compared to the experimental pattern to give a profile χ^2 analogous to the χ^2 from the Pawley fit. The degrees of freedom are then randomly adjusted by a small amount and the profile χ^2 is recalculated. The move is accepted if it is a "downhill" move (it generates a better profile χ^2), while an "uphill" move, which generates a worse profile χ^2, is only accepted with a degree of probability based on how much worse the new solution is. To avoid getting trapped in a local minimum early on in the search, a "temperature" parameter is used to determine the likelihood of accepting an uphill move. This temperature is set to be high at the start of the run, meaning that more and larger uphill moves are accepted, and decreases as the run progresses, narrowing the search to an area of the search space in which the optimum solution is found.

To maximize the chance of success, a number of strategies are employed, including the use of torsion angle restraints based on the distribution of torsion angles

present in known crystal structures.[141, 142] This information can be readily obtained from the CSD via the Mogul program,[143] and reduces the size of the search space by only sampling torsion angles that are likely to be encountered.

Although the global minimum should be found using an infinitely long simulated annealing run, in practice it is quicker to employ multiple shorter simulated annealing runs. The stochastic nature of simulated annealing means that each run may find a different minimum, in which case the visual fit of the calculated pattern to the experimental pattern must be assessed, as well as the ratio of the profile χ^2 to the Pawley χ^2, to determine which solution corresponds to the global minimum.[135]

1.4.4 Rietveld analysis

The model obtained from structure solution can be further improved by performing a Rietveld analysis[144, 145] over a more extended range of the powder diffraction pattern. This is the final step in the structure solution process. Initially developed for neutron diffraction, the method uses the full profile intensities rather than just the integrated intensities of the peaks, and simultaneously fits all of the parameters of the diffraction pattern, including background, peak shape and peak width, as well as the lattice parameters, atomic positions and temperature factors. The procedure is similar to Pawley analysis, except that here the intensities of the peaks are constrained by the crystal structure model which is being refined.[137]

The difference between the calculated profile, y^{calc}, and the observed profile, y^{obs} is a function M where

$$M = \sum_i w_i \left(y_i^{\text{obs}} - \frac{1}{c} y_i^{\text{calc}} \right) \tag{1.19}$$

which is minimised with respect to the profile parameters by a least squares procedure, with w_i being the weight of that observation, typically given by $\left(y_i^{\text{obs}}\right)^2$, and c being a scale factor such that $y^{\text{calc}} = c y^{\text{obs}}$. The least squares refinement assumes that the model can fit the data accurately, which is not the case when impurities

cause intensity in the diffraction pattern, so robust refinement methods have been developed to overcome this problem.[146, 147]

Chemical sense must be retained, so restraints and rigid bodies are used to ensure that an improvement in the fit to the diffraction pattern is only achieved while keeping the bond lengths, angles and interatomic separations sensible.[142]

1.4.5 Peak width

The peak width in a powder diffraction pattern is dependent on two factors: the instrument and the sample. Both contribute to broadening of the peak, and the instrument broadening can be accounted for by the inclusion of instrument profile coefficients into the refinement,[148] although the instrument broadening is negligible in the case of synchrotron radiation.

The contribution of the sample to the line broadening is complex and can be due to either the crystallite size or the strain in the sample, so the broadening contains significant information about the microstructure of the material. The broadening can be measured either by extracting the full width at half maximum (FWHM) of a peak, which is the width of the peak at half the maximum intensity of the peak, or by the integral breadth, which is the ratio of the peak height to the peak area. The inverse relationship between crystallite size and line-broadening has long been known,[149] and an estimate of the cube root of the crystallite volume, b, can be obtained by

$$b = \frac{K\lambda}{\beta_l \cos\theta} \tag{1.20}$$

where λ is the wavelength of the radiation, β_l is the broadening in radians, θ is the Bragg angle, and K is the Scherrer constant. This and other more complex models of line broadening due to size have been implemented in software such as TOPAS.[150]

An estimate of the strain, ϵ, in the sample, assuming the strain is isotropic, is given by

$$\epsilon = \frac{\beta_\epsilon}{C \tan\theta} \tag{1.21}$$

where C is a constant approximately equal to 4. Often the strain is anisotropic and such situations require more complex treatment of the strain.[151]

TOPAS uses the Double-Voigt approach to calculate volume-weighted mean crystallite size and a mean strain value, based on peak-shape fitting of both Lorentzian and Gaussian type line-broadening, where the size and strain contributions to the line profile shapes are identified using the angular dependence shown in Equations 1.20 and 1.21.[152]

1.5 Lattice energy calculation

The structure and properties of molecular materials are dependent on the intermolecular forces, and so the lattice energy of a material could provide information about the crystallization tendency of a molecular material. Although fully *ab initio* calculations of interaction energies are feasible for dimers of medium-sized molecules,[153] and crystal structure generation/prediction can be achieved with hybrid *ab initio*-semiempirical methods,[154] quantum chemical methods are computer-intensive, and ideally a method which gives the energy of the system under study in fractions of a second is desirable.

An approximation to the interaction between nuclei and electron clouds can be obtained by dividing the interaction energy into Coulombic-polarisation, dispersion (or London) and repulsion (Pauli) terms, in an approach which is termed the CLP assumption that has been successfully used for a number of years.[155] PIXEL, a semi-empirical scheme for calculating these terms by integration over electron densities, gives reliable results with affordable computing times, but still requires use of a quantum chemical method such as GAUSSIAN to evaluate the charge density.[156, 157]

More recently, an atom-atom approach to such a calculation has been developed, the AA-CLP method, which does not require the initial evaluation of the charge density.[158]

Instead, a few standard atomic parameters (such as polarisability, ionisation potential and atomic number) are used to calculate atom-atom potential functions for each particular molecule. While less accurate than the PIXEL method, the heats of sublimation of 154 organic crystal structures were successfully reproduced by this computationally faster approach.

1.6 Thesis Overview

The crystallization properties of organic molecular materials are important not only for the pharmaceutical industry, but also for any materials chemist wishing to obtain single crystals for SXRD analysis. The aim of this project was to develop predictive models to assess ease of crystallization of organic molecular materials, and use these models to rationalise the molecular and structural factors governing the crystallizability.

To assist reproduction of results reported in this thesis, chapter 2 describes the experimental techniques and software tools used, including details of the methods relating to dataset creation, algorithm training and model testing, as well as the experimental validation techniques and powder diffraction studies.

In the first part of Chapter 3, a dataset of drug-like molecules is used to train and evaluate an initial predictive model. The accuracy is improved by careful curation and dataset extension, and rules are extracted to rationalise crystallization propensity in terms of molecular descriptors. An extension of the approach to co-crystallization is also presented.

Chapter 4 uses the information extracted from the models in Chapter 3 to guide the development of a new descriptor to better capture the molecular flexibility. Several methods of generating this descriptor are explored and evaluated, and the effect of incorporating it into the original model is tested.

Chapter 5 contains two experimental validation methods, the first of which is a blind recrystallization screen of a set of molecules previously not reported to crys-

tallize. The second is a set of controlled cooling crystallizations to test whether the model predictions are related to the cooling rate required for high quality crystals to grow—a direct measure of "crystallizability".

Chapter 6 describes the use of high resolution synchrotron X-ray powder diffraction to investigate whether there is a relationship between the microstructure and the crystallization propensity of a set of nearly 200 molecular materials. The details of nearly 70 previously unreported crystal structures determined as a result of this investigation are included.

The final chapter sets out the conclusions from the model testing, training and validation presented in the thesis. Ideas for the extension of the methods to other applications are also presented.

Chapter 2

Experimental

This section records the experimental techniques and computer software used through the course of this project. All computational techniques were implemented using Python 2.7.10 unless otherwise stated. All manipulation of molecules was performed using the RDKit cheminformatics toolkit as implemented in Python.

Contents

2.1 Database curation

The performance of a predictive model created using a machine learning algorithm depends largely on the quality of the input data which is used to train the algorithm. For a binary classification problem, training data is required for both classes, which necessitates either mining databases to extract molecules, or creating an in-house set of training data. The choice of training data is important as it sets the domain of applicability of the model, an important concept which states that model predictions are only reliable for molecules which occupy a similar area of chemical space to those in the training set.[159]

2.1.1 Crystallization predictions

A reliable set of input molecules which are known to crystallize can be compiled from the Cambridge Structural Database (CSD). If we assume any material which is not completely amorphous to be "crystalline" to some extent, with kinetic or thermodynamic factors being the cause of restricted crystal growth in less crystalline materials, then we will only encounter crystal sizes upwards of 0.1 mm in SXRD experiments (or perhaps as small as 10 µm for synchrotron radiation experiments). Such a crystal size will contain enough material to produce a diffraction pattern with sufficient signal for the structure to be determined. Consequently, a distinction can be made between those crystalline molecules which are able to have their structures determined by SXRD, and those where the domain size is too small for such an analysis. The CSD provides a starting resource for making this distinction, and we can say that these molecules are "crystallizable".

However, information on those molecules which cannot be crystallized is not readily available. Failed crystallization attempts are rarely recorded, and the rate at which new structures are added to the CSD is an indicator that a database of all crystalliz-

able structures is far from complete. There are a number of reasons why a material may not have a structure recorded in the CSD; the crystallization may never have been attempted, the crystal structure may not have been solved, and for proprietary reasons the crystal structure may not be publicly available.

Promisingly, there are databases much larger than the CSD which contain a significant number of molecules. Databases such as PubChem[160] contain millions of molecules with information on their biological activity, while attempts have been made to enumerate all possible molecules with a given maximum molecular size, resulting in the GDB databases.[161] In order to keep the selection relevant to molecules that might be easily obtained, the ZINC database[162, 163] of purchasable molecules was used.

The use of these two databases, and the limitations of the data available in them, led to the treatment of the problem as a binary classification problem, since the only potential outcomes we can learn from this data is whether a molecule can form a single crystal or not. Preliminary attempts to treat the problem as a multi-class classification, by including materials which form powders or solvates as separate classes, were unsuccessful due to the relatively small number of data points for these potential classes.

There are two main differences between the CSD and ZINC. The first, and the one of interest for this project, is that the CSD contains solely crystallizable compounds, while the ZINC database contains both molecules which will crystallize and those which will not. This necessitates the identification and removal of crystallizable molecules from ZINC to give a set of non-crystallizable molecules.

The second is that the ZINC database contains purchasable compounds, while the CSD contains a large number of research materials, which would not be found in the ZINC database, in addition to many purchasable compounds. It would be desirable to eliminate any extra differences between the two datasets so that the only difference between them is the crystallizability of the molecules. The easiest way to accomplish

this is to identify only those purchasable molecules within the CSD to be used as the crystallizable dataset, since these are likely to be relatively similar to the purchasable non-crystallizable molecules being extracted from the ZINC database. As a result, it was decided that only those CSD molecules which are also present in ZINC were to be included in the databases of test and training molecules. This means that every crystallizable and non-crystallizable molecule has come from the ZINC database, as shown in Figure 2.1, so there is no bias towards one database or the other.

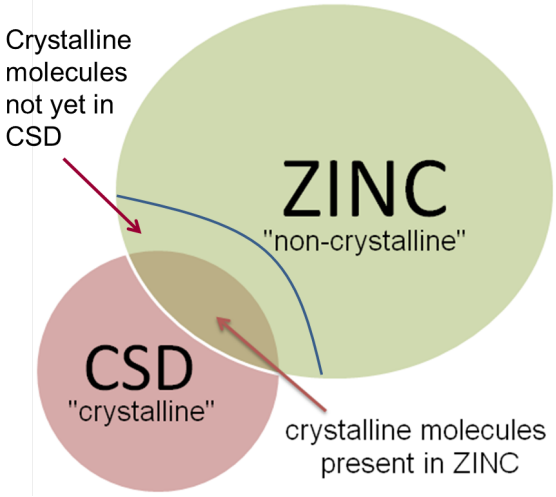


Figure 2.1 Schematic of the overlap of the CSD with the ZINC database.

The challenge lies in cross-referencing the CSD with ZINC. This step is important as it provides the "crystallizable" set of purchasable molecules, while also removing these molecules from the set used to create the "non-crystallizable" class. One of the best ways to compare molecules is using their SMILES string, since it is programmatically easy to check their presence in a set of reference SMILES. However, this requires molecules from both databases to be represented in a canonical form, so that comparisons can be made. RDKit converts identical molecules into their canonical SMILES automatically, but a pre-processing step is required to standardise the molecules. This involves ensuring each atom is in its neutral charge state, and accounting for potential tautomerisation by enumerating all possible tautomers and selecting the most probable one.

As already stated, the CSD is not a comprehensive database of all molecules that

can be crystallized. Consequently, there may be some crystallizable molecules which remain in the ZINC database after cross-referencing with the CSD, which would be incorrectly labelled as non-crystallizable. The existence of these false negatives in the training dataset may cause the algorithm to perform badly. However, we were confident that there was enough information already present within the CSD to outweigh the small contribution that these might make towards the classification. The random sampling of the non-crystallizable molecules to create a training set of molecules of similar size to the crystallizable set meant that it was unlikely that many false negatives would be selected. It was also hypothesised that any false negatives present within the test set might be identified by the model, which would allow identification of materials which are crystallizable but which have not been added to the CSD yet.

ZINC

The initial dataset was taken from the ZINC12 database,[162] from which the "All purchasable" set of molecules was downloaded in August 2012 as zipped SDF files. The subsequent updates and extensions to the data were made using the ZINC15 database,[163] from which the full "clean" set was downloaded as SMILES files. In both cases, the download was initiated using a batch script downloaded from the ZINC website, which retrieved the relevant files from the online Uniform Resource Locators (URLs) using the `wget` program.

CSD (using CONQUEST)

Initially, entries were extracted from the CSD using the CONQUEST interface to the November 2013 version of the database. Organic molecules were selected by searching for structures containing at least one carbon atom. Powder structures were excluded because the intention is to create a "crystallizable" dataset with molecules which can be crystallized, which is not necessarily the case for a structure solved by powder diffraction, and so their structures should have been solved by SXRD. Organo-

metallic compounds were also excluded, because the processes and interactions involved are significantly different to those for purely organic compounds, and in addition the RDKit toolkit is unable to handle these types of molecules. The 'Any' bond type was deselected and disordered atoms were excluded.

These molecules were exported as both SDF files and SMILES files. Both sets were curated to exclude molecules which were not recognised as molecules by the toolkit, had no atoms, or could not be converted into a SMILES string. For the remainder, the overall SMILES string, which contained each component separated by a period, was sorted by length of component SMILES string to obtain the main component of the molecule. Solvents of crystallization were deleted by comparing the components with a list of commonly occurring solvent molecules in the CSD (see Appendix A.3). Molecules which had crystallized as multicomponent crystals which were not solvates were removed from the set altogether. Salts were removed by ensuring that the remaining main component of the molecule was neutral. Both lists were then combined, excluding duplicates, which were identified by comparing their canonical smiles.

CSD (using Python API)

The CSD Python API[164] provides a direct interface to the CSD, which aids the curation of the dataset by removing the need to manually process the output from CONQUEST. Version 1.0.0 was used as the interface to CSD v5.37 (November 2015). For each entry in the CSD, entries were excluded according to the following rules, for which the code snippet is given in Figure 2.2:

1. The entry had no molecule associated with it - this indicated that only the unit cell had been determined, rather than the actual crystal structure itself.

2. They were not organic as defined by the CSD, where an organic material is one that contains no *p*-block metal, transition metal, lanthanide or actinide elements.

3. They had some atoms with no sites - again this indicated that only a partial crystal structure determination had been achieved, such as in the case where there is a disordered solvent present in the crystal structure.

4. The molecule was present as a salt - in other words the main component of the molecule is charged.

5. The structure was obtained by powder diffraction.

```
from ccdc import io

csd_reader = io.EntryReader('CSD')

for entry in csd_reader:
    try:
        mol = entry.molecule
    except:
        continue
    if entry.is_organic:
        main_mol = mol.heaviest_component
        if main_mol.all_atoms_have_sites:
            if main_mol.formal_charge == 0:
                if not entry.is_powder_study:
                    print "keep molecule"
```

Figure 2.2 Code snippet for the initial CSD dataset curation.

The "main component" as defined by the CSD Python API was then taken as the molecule, and these were converted to MolBlock format which can be read by the RDKit toolkit. Solvents were removed by checking against the self-curated list of common solvents, and multicomponent crystals which were not solvates were again removed. Duplicates were removed, but for each molecule a representation was retained which had a specific chirality, to provide a starting point for conformer generation in a later step.

Cross-referencing CSD with ZINC

It was necessary to standardise the SMILES strings from both sets of molecules before cross-referencing between the two databases to identify molecules present in both, and to remove duplicates. Molecules were converted back into their SMILES

string by the RDKit to ensure that the RDKit canonical SMILES for that particular molecule was indeed being used. Charges were neutralised according to a standard set of neutralisation reactions given in Figure 2.3. This ensured the molecules were not in zwitterionic form, as in the case of the molecule Figure 2.4, or in salt form in the case of molecules from ZINC, such as that shown in Figure 2.5.

```
def _InitialiseNeutralisationReactions():
    patts= (
        # Imidazoles
        ('[n+;H]','n'),
        # Amines
        ('[N+;!H0]','N'),
        # Carboxylic acids and alcohols
        ('[$([O-]);!$([O-][#7])]','O'),
        # Thiols
        ('[S-;X1]','S'),
        # Sulfonamides
        ('[$([N-;X2]S(=O)=O)]','N'),
        # Enamines
        ('[$([N-;X2][C,N]=C)]','N'),
        # Tetrazoles
        ('[n-]','[nH]'),
        # Sulfoxides
        ('[$([S-]=O)]','S'),
        # Amides
        ('[$([N-]C=O)]','N'),
        )
    return [(Chem.MolFromSmarts(x),Chem.MolFromSmiles(y,False)) for x,y in patts]
```

Figure 2.3 Neutralisation reactions in the form of SMARTS transformations.

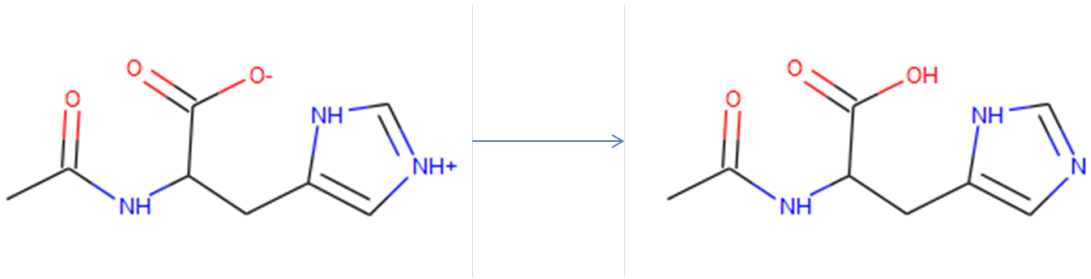


Figure 2.4 Neutralisation of the zwitterionic form of CSD refcode ACHIST20.

For the original dataset, no treatment of tautomers was included as it was beyond the scope of the original work. Subsequently, tautomers were accounted for using the tautomer canonicalisation function within the `molvs` package. This involves enumerating all possible tautomers (with a limit of 1000 to prevent combinatorial

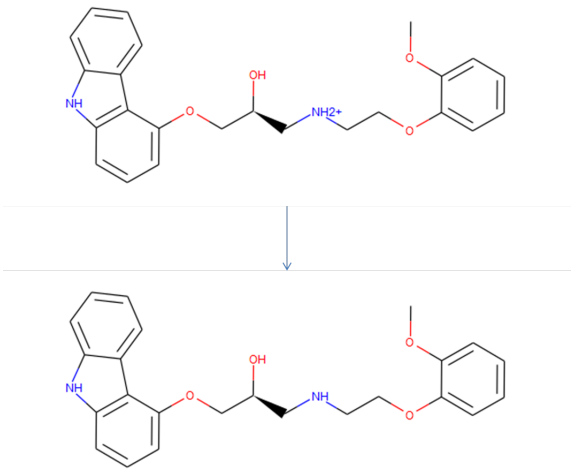


Figure 2.5 Neutralisation of the salt form of carvedilol from ZINC12 (ZINC1530579, CSD refcode GIVJUQ).

explosion) for a particular molecule based on a standard set of tautomer transforms, then scoring each tautomer based on the functional groups present, and converting the molecule into the best-scoring tautomer. An example of this process is given in Figure 2.6.

Molecules were compared using their non-isomeric SMILES, which matched CSD molecules with any stereoisomers present in the ZINC database. This allowed the identification of any CSD molecules which were "purchasable" regardless of their isomeric form. By doing so, the possibility of failing to match molecules for which an enantiomer had been crystallized while the racemate was present in ZINC was avoided. Those from ZINC which were successfully matched to ones in the CSD were added to the database of crystallizable molecules, while those which did not match were added to the database of non-crystallizable molecules.

Training and test set generation

The non-crystallizable set of molecules was randomly sampled to create a database of similar size to the crystallizable set of molecules, thereby avoiding bias between the two classes.[165] These sets of molecules were combined into one database. This was randomly sampled to split the data into a training set for parameter tuning and

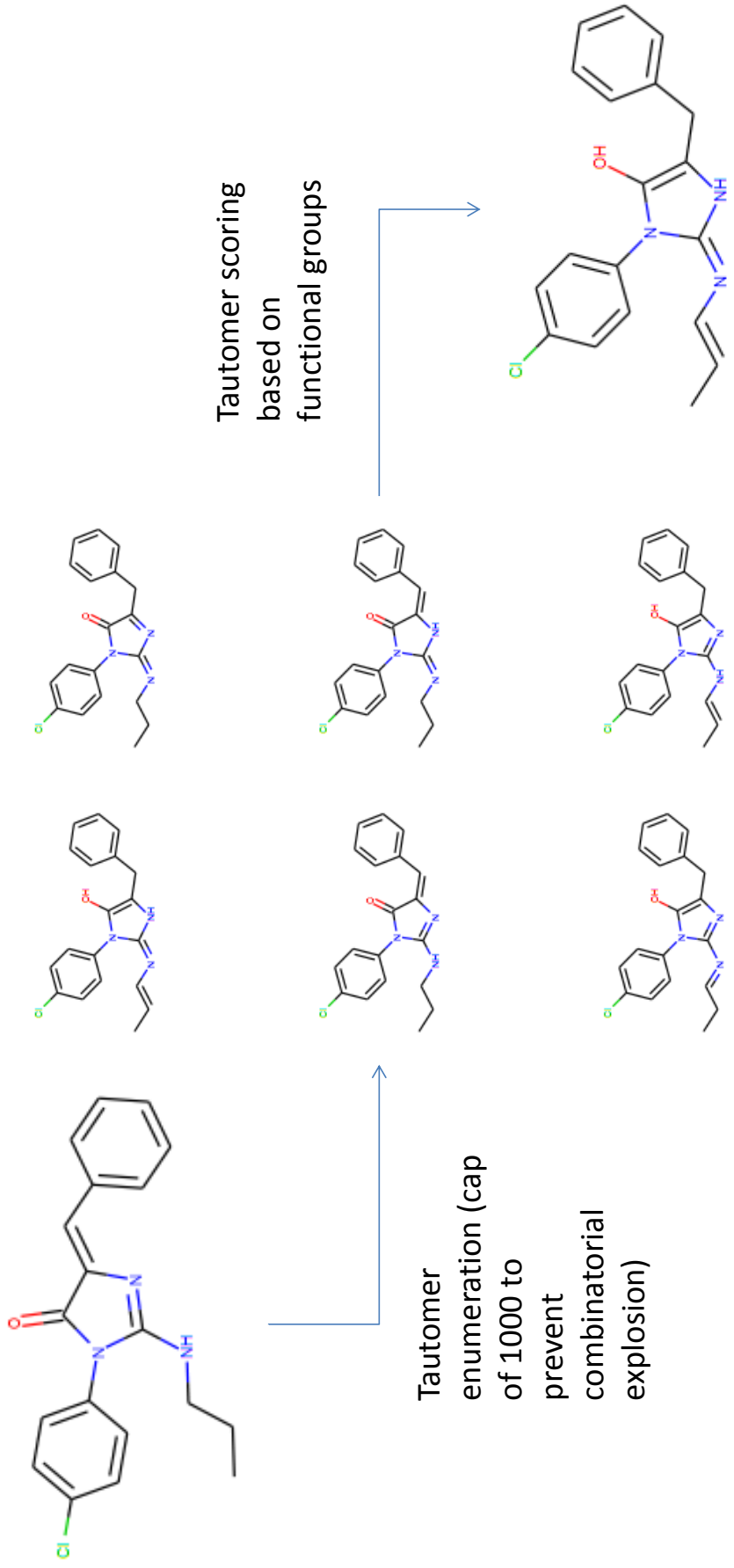


Figure 2.6 Example of tautomer canonicalisation for CSD refcode ADANAV.

algorithm training, and an external test validation set in a 3:1 ratio, with a roughly equal number of crystallizable and non-crystallizable molecules in each. Random sampling was chosen over rational sampling since this gives a more accurate estimate of the predictive ability, as a rational test set selection method has been found to yield an overly optimistic estimate of predictive ability.[166]

2.1.2 Co-crystallization predictions

Although the CSD contains crystal structures of co-crystals, negative results from co-crystallization studies are rarely reported in the literature, so it was necessary to generate a landscape of both positive and negative experimental data to support the training of the machine learning algorithm. The synthesis of these materials was carried out by Lorraine Crowley at University College Cork, in conjunction with Oliver Robshaw and Edmund Little, a Part II student and summer student respectively.

A set of 20 co-former molecules (Figure 2.7) was selected to be screened against 34 substituted aromatic acid and amide materials (Figure 2.8), which represent the potential APIs. Careful consideration was given in the selection process to incorporate the potential for all four main supramolecular synthons discussed in Chapter 1. Synthesis involved mixing 1 mmol of an API with 1 mmol of a co-former. These were combined in a steel reaction vessel containing two ball bearings, which was rapidly oscillated using a Retsch MM 400 ball mill at a frequency of 30 Hz for 20 minutes.

Assessment of co-crystal formation was based principally upon changes in the powder X-ray diffraction pattern when accompanied by a change in the infrared (IR) spectrum of characteristic peaks traditionally involved in hydrogen bonding. An unsuccessful co-crystallization results in a diffraction pattern which is simply the summation of the patterns of the pure reagents. The appearance of new peaks and disappearance of reagent peaks indicated that a co-crystal had been formed. This was confirmed by examination of the IR spectrum, where shifts in the peaks of functional groups involved in hydrogen-bonding indicate the formation of a new bonding net-

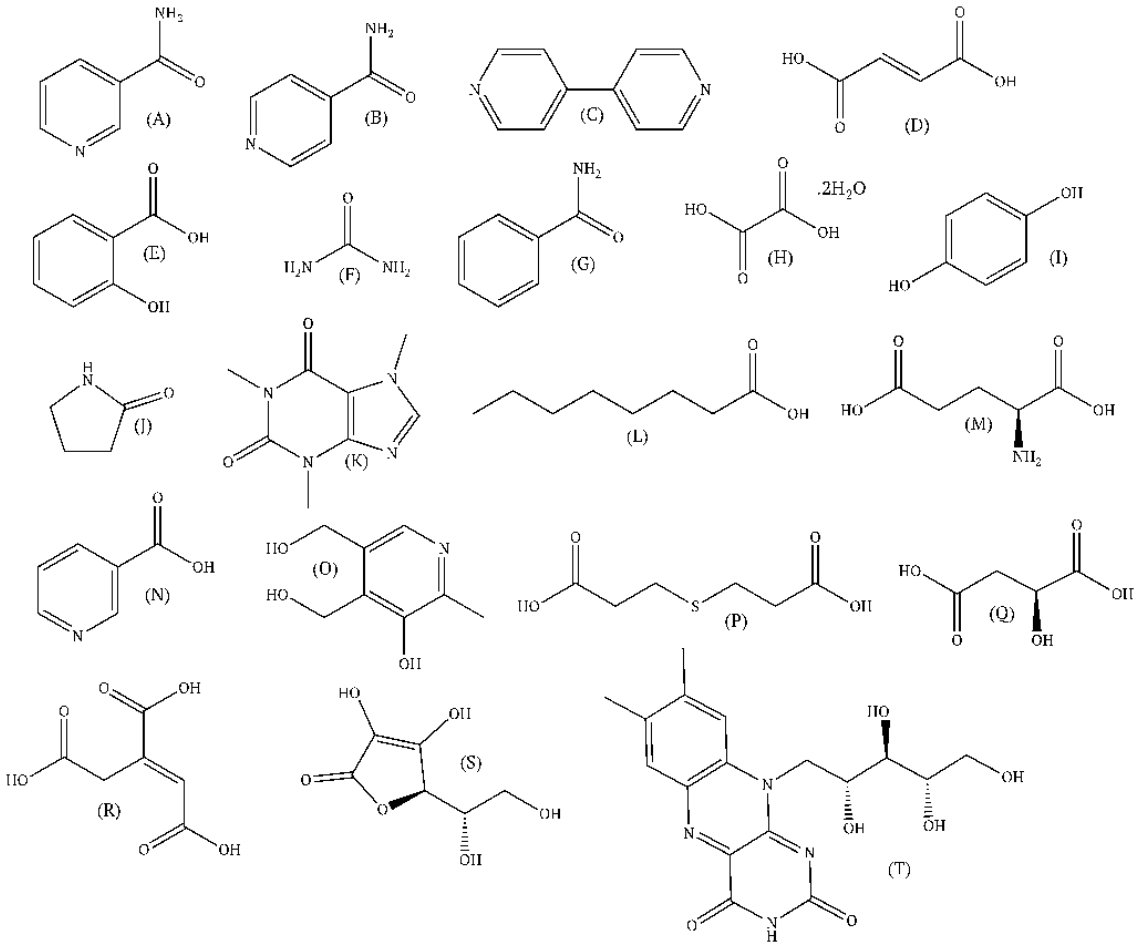


Figure 2.7 Co-former molecules used in this study.

work. Ambiguous situations could result in cases where the two components act as mutual impurities, inserting only small quantities of themselves into the lattice of the other.

A successful co-crystal was given the label 1, while an unsuccessful co-crystal was given a label of 0. For those few where the outcome was ambiguous, a label of 2 was given to allow the pair of components to remain in the grid while being ignored by the machine learning algorithm. This also applied to isonicotinamide, which could not be obtained for attempted co-crystallization with the amides, and to the salicylic acid–salicylic acid co-crystal (since it appears in both the co-former and API list).

This bespoke co-crystal screen provided a set of 657 data points to use as a training set of data. An external test validation set was obtained from a literature virtual co-crystal screen of paracetamol,[43], which provided data from several co-crystal

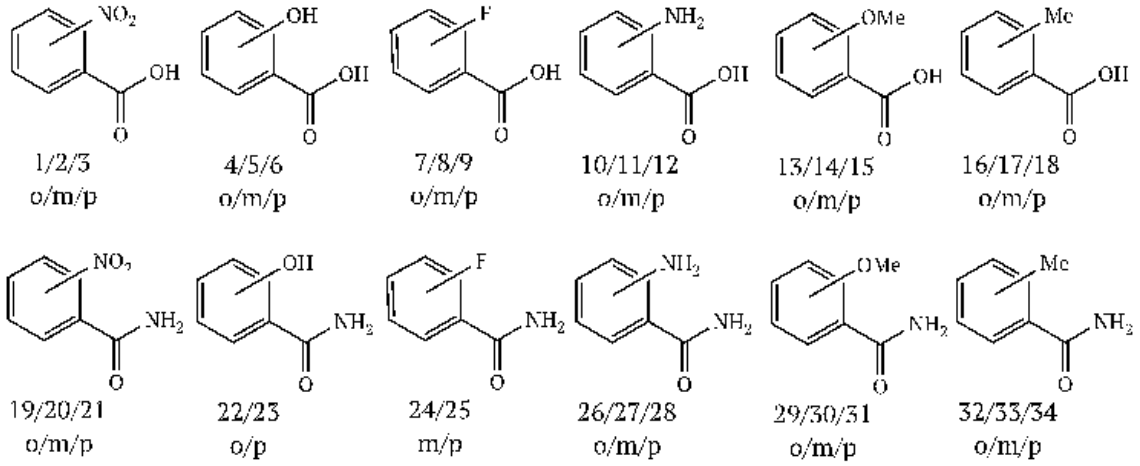


Figure 2.8 Acid and amide ("API") molecules used in this study.

screens,[167–171], covering 34 attempted co-crystallizations.

2.2 Descriptors

2.2.1 Standard descriptors

Descriptors were generated using the RDKit cheminformatics toolkit. The initial model used the set of descriptors from RDKit version Q4 2013, which calculated a standard set of 177 descriptors for each molecule. The descriptors in subsequent updates to the model were calculated using RDKit version Q1 2016, which contains an extended list of standard descriptors. For continuity, the crystallization predictions used the original RDKit set of descriptors.

MQNs were calculated using a script developed from the work of Nguyen *et al.*[90] which was self-coded and is available in the Appendix.

The standard set of RDKit descriptors contains no explicit flexibility descriptors other than RBC, meaning that no such descriptors were included in the original model. For comparison using single variable classifiers, two such descriptors were calculated according to the methods described in Section 1.2.6.

2.2.2 Co-crystal Descriptors

For the co-crystal predictions, the new set of RDKit descriptors was used, but certain ones were excluded from the feature vector as they gave undefined values (NaN) for some molecules (MinPartialCharge, MaxPartialCharge, MinAbsPartialCharge, MaxAbsPartialCharge, Ipc). This generated 191 standard descriptors for each molecule.

A new descriptor was developed using the electrostatic potential values for SSIPs calculated by Hunter.[40] This involved uniquely identifying functional groups which could participate in hydrogen bonding. This necessitated the modification of some functional group count descriptors, while new descriptors were added to account for aryl fluorides, aryl chlorides, other aryl halides and tertiary amides.

A look-up table was created containing the donor and acceptor electrostatic potential values, ϵ, for each hydrogen-bonding functional group. For a particular co-crystal, the functional groups were identified according to the counts generated by RDKit. For each functional group present, the value of the electrostatic potential from the lookup table was added to a list. This ranked list of values for acceptors and donors was used to identify the pairs of functional groups that were most likely to interact. For each interacting pair, the energy of the interaction is given by $\epsilon_i \epsilon_j$. The energy of a particular material can therefore be estimated by

$$E = \sum \epsilon_i \epsilon_j \tag{2.1}$$

with the overall energy of cocrystal formation being given by:

$$\Delta E = E_{\text{cocryst}} - E_{\text{API}} - E_{\text{co-former}} \tag{2.2}$$

Since a co-crystal is formed from two separate molecular components, the feature vector was required to be a combination of the two. The descriptors of the co-former were appended to those of the API, with the Hunter's descriptor being appended to the end of this, creating a feature vector containing 391 descriptors (Figure 2.9). It

was important to ensure that the descriptors of the components were appended in the correct order, so that the inputs to machine learning algorithm were consistent, otherwise the model could be incorrectly trained. The consequence of this is that the API and co-former are identified differently in the input, which means that the same consideration must be made when preparing test co-crystals for prediction by the model.

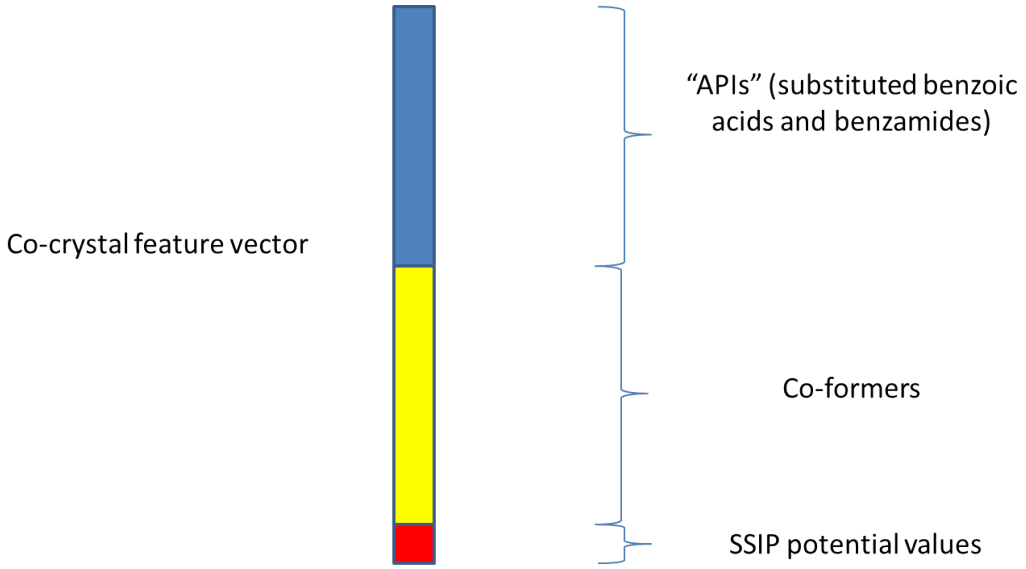


Figure 2.9 Schematic diagram of the feature vector for a co-crystal, made up of the API descriptors, the co-former descriptors, and the descriptor derived from SSIP electrostatic potential values.

2.3 Machine learning algorithms

All machine-learning algorithms and performance metrics were implemented using version 17.0 of the scikit-learn package.

2.3.1 Pre-processing

Scaling

RF algorithms can be used with the raw values of the descriptors. However, for SVM algorithms, which can be dominated by descriptors with larger numeric ranges, it was

necessary to scale the data to have zero mean and unit variance for each descriptor. This scaling was calculated using the training set, and then applied to both the training set and the test set.

2.3.2 Hyper-parameter tuning

The hyper-parameters of each model can only be tuned in an empirical manner. A grid-search method was employed to tune the hyper-parameters of the SVM estimator, using a logarithmic grid for C with a linear kernel, and an equivalent logarithmic grid for C and γ with the RBF kernel. For the RF classifier, the number of trees was optimised.

In each case, a k-fold cross-validation was carried out on the training set of points, where the data was split into k sets of equal size, with each set being used as the test set for a model trained using the remaining k−1 sets as the input data for the algorithm. The value of k to choose is a compromise; a large value of k reduces the bias at the cost of giving a larger variance and being computationally expensive, while a small value of k can lead to a large bias. A common value of k to use is 5, as it is a good compromise between speed and reducing the bias.[172] The data were split using a stratified splitting algorithm to ensure that the distribution of "crystallizable" and "non-crystallizable" molecules in the split data was the same as in the original training dataset. This method allows the best hyper-parameters to be determined by analysing both the mean predictive accuracy of the 5-fold cross-validation, and the spread of accuracies. Since each molecule is tested only once, this allows all of the training data to be used to evaluate the hyper-parameter performance.

Holding out an external validation set to test the performance of the final model with the chosen parameters prevented any information from the test set "leaking" into the training of the model by ensuring that molecules in the validation set were unseen by the algorithm. This prevented these molecules from having any influence on the parameter selection, which might otherwise artificially improve the accuracy

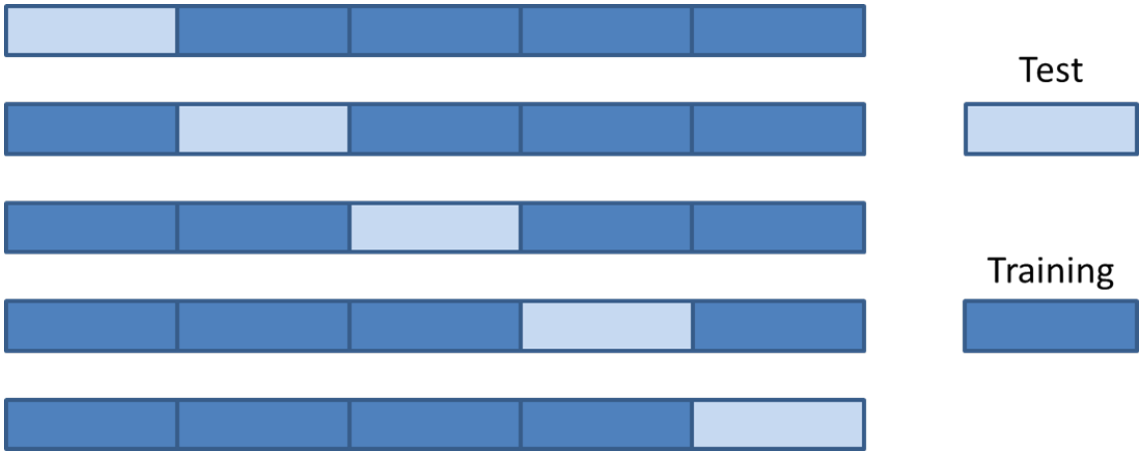


Figure 2.10 Schematic diagram of the splits of the data for k-fold cross-validation, with k=5.

of the model.

2.3.3 Learning curve

A learning curve displays the change in predictive accuracy of the classifier with varying training subset size. This is an important thing to consider as it demonstrates how much extra information is being learned by the algorithm as the number of training examples increases.

To generate a learning curve, the training data was split into 5 subsets using a shuffle-split algorithm to give 5 random subsets in the usual cross-validation approach. Each subset was used as the test set, with the remaining 4 subsets being combined to create a training set each time. This smaller training set was then systematically reduced in size and used to train an algorithm and produce a model which was tested on the same test set each time. The training and test scores for each training set size were averaged across the 5 folds to find an average score for each training set size, with the standard deviation of these scores being used to plot error bars.

For random forest, the training score is an inappropriate metric to use, as each training point will have a path through a tree to a correct prediction and so the training score will always be perfect. The score on the training set was replaced with the out-of-bag score (the average score on the held-out data from each tree in the forest).

2.3.4 Feature extraction

Identification of the most important features involved in the classification is useful from both a scientific and a practical perspective; as well as providing insight into the features of a molecule which provide the best classification, removal of irrelevant features can reduce the computational cost and time.

For the linear SVM and random forest algorithms, the feature importance is relatively easy to extract; the random forest automatically computes the feature importance on the out-of-bag data for each tree, as described in Section 1.3.2, while the coefficients of the linear separating hyperplane are indicative of feature importance for the linear SVM.

The black-box nature of the SVM algorithm when using a non-linear kernel prevents the direct determination of the importance of each descriptor in performing the classification, so an independent feature selection analysis was required. Feature selection can be divided into two types; filter methods, which select a subset of variables in a pre-processing step, and wrapper methods, which directly use the performance of the classifier to select variables.[173] Wrapper methods were primarily used for this work.

Single and two variable classifier approach

Each variable in the feature vector was used in turn as the input data to create a single variable classifier[174] and the accuracy of each was determined using a five-fold cross-validation strategy as above. This was also compared with the classification accuracy on the external test set.

Once the most important variable had been determined, this could be used in conjunction with every other variable to create a two variable classifier.

Decision tree analysis

Rule extraction techniques can be used to mimic how the SVM is performing the classification. The algorithm is trained on the training set of data as usual, then the predictive model from this is used to predict on the training set and reassign the training labels based on this. A conventional decision tree classifier can then be trained on these reassigned labels to provide a representation of the SVM predictive model in terms of a rule-based decision tree, a human-readable format.

2.4 Experimental validation

2.4.1 Blind test

The best-performing predictive model was used to select molecules for the micro-crystallization screen. The prices (as available from well-known commercial chemical suppliers) were obtained for a set of randomly chosen ZINC molecules, and 20000 molecules below a certain price threshold were selected. These molecules were then separated into clusters in order to obtain the widest possible variation in functional groups and features. This helped to ensure a fair test by removing the possibility that the model only used one or two functional groups to make decisions about its predictions. Morgan fingerprints[175, 176] of each molecule were used to generate a distance matrix based on the Tanimoto similarity coefficients[177] between each molecule and every other molecule. The Ward hierarchical clustering algorithm,[178] was used to create clusters of molecules, and a molecule was randomly chosen from 20 clusters, giving 12 materials strongly predicted to be crystallizable (with a probability greater than 0.75) and 8 materials with a predicted crystallization probability below 0.75.

These were purchased and decanted by a colleague into numbered vials in order to perform a blind test. Attempts were made to recrystallize each molecule to form crystals of sufficient size and quality for SXRD to be carried out and gain a structure of the molecule. The method of recrystallization used was slow evaporation from a pure

solvent. A range of solvents of varying polarity and volatility were chosen in order to provide the broadest possible range of conditions so that each molecule had the maximum possible opportunity to form crystals. These solvents were ethanol (EtOH), dichloromethane (DCM), dimethylformamide (DMF), ethyl acetate, chloroform and diethyl ether. The first solvent was used for all 20 molecules, with any molecules which crystallized being removed from the screen for subsequent solvents.

2.4.2 Controlled cooling

The level of supersaturation of a solution (and therefore the crystallization) can be controlled either by solvent evaporation or solution cooling. As it is practically easier to control the rate of cooling of a solution than to control the rate of evaporation of the solvent, this method is used for the controlled crystallization experiments.

A set of 24 in-house materials containing a secondary amide functional group, with varying numbers of rotatable bonds, were assigned predicted crystallization probabilities based on the best-performing predictive model. In order to test the hypothesis that molecules with a lower predicted crystallization probability require a slower rate of increase in saturation to form crystals, the molecules were subjected to three different rates of cooling. First, a supersaturated solution of each molecule in acetone was made by heating the solutions in a small vial (containing excess of the compound of interest) to 55 °C. The supersaturated solutions were then pipetted into separate, clean vials (which were also heated to 55 °C to make sure the solution did not cool down again straight away). The vials were then sealed and the solutions were cooled over set periods of time; 1 day, 3 days and 1 week.

The experiment was carried out by Sophie Gearing, a Part II student in the group, using a heating block with small vial holders made in the electronics workshop. The experiments were controlled using a Eurotherm 2416 setpoint programming controller which could be programmed to heat up and cool down over a given time period. However, the block had no way of cooling itself down; it allowed the solutions to cool

down naturally and controlled the rate by switching the heating device on and off to maintain a set point temperature. This meant that the lowest controllable temperature the machine reached was around 30 °C, so this was programmed to be the end point of the experiment. At the end of each experiment, the vials were checked for crystals using optical microscopy.

2.5 Synchrotron X-ray powder diffraction

Materials were selected to be studied by synchrotron X-ray diffraction based on their predicted crystallization propensity from the final model. Materials from a Novartis in-house set of publicly available drug-like molecules were clustered based on their Morgan fingerprints,[175, 176] using the Taylor-Butina clustering algorithm.[179, 180] Clusters with varying spreads of crystallization propensities were chosen for further analysis.

All synchrotron radiation measurements were collected on beamline ID22 at the ESRF used a beam of wavelength 0.3997 Å at a temperature of 100 K.

1.5mm Kapton capillaries of length 30mm with a wall thickness of 0.033mm were used to contain the powders, as this gives a lower and more consistent background contribution to the diffraction pattern than borosilicate capillaries. Each capillary was sealed at one end using glue. An electric toothbrush was used to vibrate the capillary as the powder was added, which ensured that the powder was packed as tightly as possible, creating a sample that was uniform within the beam. The other end of the capillary was then sealed with cotton wool.

Capillaries were attached to magnetic brass mounts using a small amount of wax, ensuring that the capillary was correctly aligned to minimise movement within the beam (Figure 2.11). These were then mounted on the 75-space automatic sample changer. Samples were spun at 787 rpm which increases the number of crystallites that contribute to the diffraction pattern at a particular 2θ position by changing the orientation of the individual powder grains, allowing more crystallites to satisfy the

Bragg condition.

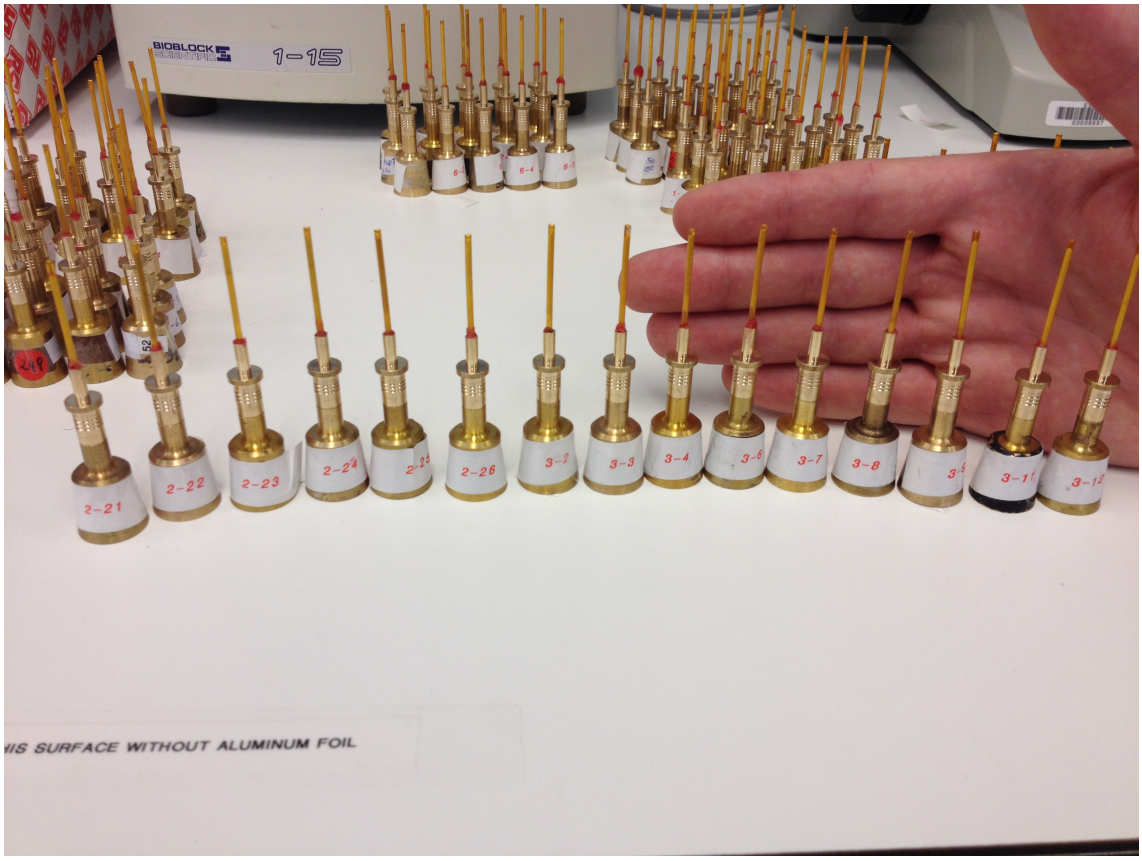


Figure 2.11 Photo of capillaries attached to brass mounts for mounting on the sample changer.

For each sample, measurements were taken at 4 different positions along the capillary to reduce the effect of radiation damage on the diffraction pattern. Scans were taken across a 2θ range of -5 to 25 degrees at a collection rate of 30 degrees per minute. At the final position on the capillary, 5 further scans were taken at the same collection speed to determine the extent of the radiation damage occurring for that particular sample.

2.6 Energy calculations

2.6.1 Geometry optimisation

Geometry optimisation of the crystal structures from the ESRF data was undertaken using CASTEP version 8.0. The PBE exchange-correlation functional[181] was used

with a Grimme06 dispersion correction.[182] This scheme has been shown to be one of the best performing methods for reproducing crystallographically–determined structures[183] and has been successfully used in previous similar studies involving molecular structure validation.[154, 184] Following convergence testing on a representative system, a plane-wave energy cutoff of 600 eV was used, with a kpoint spacing of 0.08 $Å^{-1}$ and with fixed lattice parameters.

2.6.2 Lattice energy calculation

Atom-atom lattice energy calculations were performed using AA-CLP version 3.0. The runclp.bat script was used to run the following 4 modules for the full calculation:

1) Retcif - retrieval of crystal structure from cif file. The geometry-optimised structure was then used as the input.

2) Retcor - used to renormalise hydrogen atom positions and check the symmetry of the crystal.

3) Retcha - generates atomic point-charge parameters used in the final calculation

4) Clpcry - calculates atom-atom lattice energy and classifies short contacts.

Chapter 3

Predictive Models for Crystallization

This chapter presents the development of predictive models for crystallization using machine learning algorithms. The initial parameter selection and model evaluation is carried out on a drug-like set of molecules. The curation method is subsequently improved and used to generate models using various input datasets and descriptor types. The methodology is finally extended to the related problem of predicting co-crystallization. Models with error rates below 10% are obtained for all attempted test cases, while over 2-fold enrichment is achieved for the co-crystallization predictions. The initial model development is published in Wicker, J. G. P., & Cooper, R. I. (2015). Will It Crystallise? Predicting Crystallinity of Molecular Materials. *CrystEngComm*, **17**(9), 1927–1934.[185] The work on co-crystallization prediction has been accepted for publication as Wicker *et al*, Will They Co-crystallize?, CrystEngComm.

Contents

3.1 Initial model

3.1.1 Introduction

The initial parameter selection and model creation was undertaken using a drug-like filter based on Lipinski's Rule of Five[186, 187] (Table 3.1) which kept the model relevant to the problem of pharmaceutical models while also reducing the dataset to a manageable size.

Table 3.1 Drug-like filter based on Lipiniski rule-of-5.

Physicochemical property	Value
Molecular weight	$150 \leq x \leq 500$
Total polar surface area	< 150
Hydrogen bond donors	≤ 5
Hydrogen bond acceptors	≤ 10
CLogP	≤ 5
Rotatable bonds	≤ 7

The breakdown of the training and test molecules for the drug-like data is shown in Table 3.2. The non-crystallizable class size was chosen to be approximately equal to that of the crystallizable class in order prevent bias towards a particular class.

Table 3.2 Breakdown of training and test molecules for the drug-like dataset.

	Non-crystallizable	Crystallizable	Total
Training	13440	13453	26893
Test	4480	4485	8965
Total	17920	17938	35858

The distributions of the molecular features used for the drug-like filtering are shown in Figure 3.1 for the crystallizable and non-crystallizable sets of molecules.

From these histograms, it is clear to see that the descriptors which are most affected by the filtering process are logP and molecular weight, as these have the most abrupt cut-offs at the edges of the histogram. Inspection of these histograms can begin to give an insight into the differences between the two classes of molecules. The most obvious difference is in the molecular weight, where the mode of the non-crystallizable distribution is 100 Daltons greater than the crystallizable distribution, suggesting that the crystallizable molecules are smaller than the non-crystallizable ones.

The smaller size of the crystallizable molecules may have some effect on the distributions of the other descriptors *e.g.* a smaller molecule is more likely to have fewer hydrogen bond donors and acceptors. However, this could also indicate a distinguishing property between the two classes. Other than molecular weight, rotatable bond count is the descriptor which shows the biggest difference between the two classes, with the modal count for crystallizable molecules being 2 and the modal count for non-crystallizable molecules being 5, suggesting that crystallizable molecules may be less flexible than non-crystallizable ones.

While human visual inspection is useful for making some qualitative interpretations of the data, it is impractical to manually carry out such an inspection for every descriptor in the set. Although a rule could be deduced from each histogram which allows classification based on a particular threshold for that descriptor, it is too difficult to determine a decision boundary from these histograms which separates the two classes in more than one of the dimensions of the feature space. This is a situation in which machine learning techniques are useful for quickly building a model from this data.

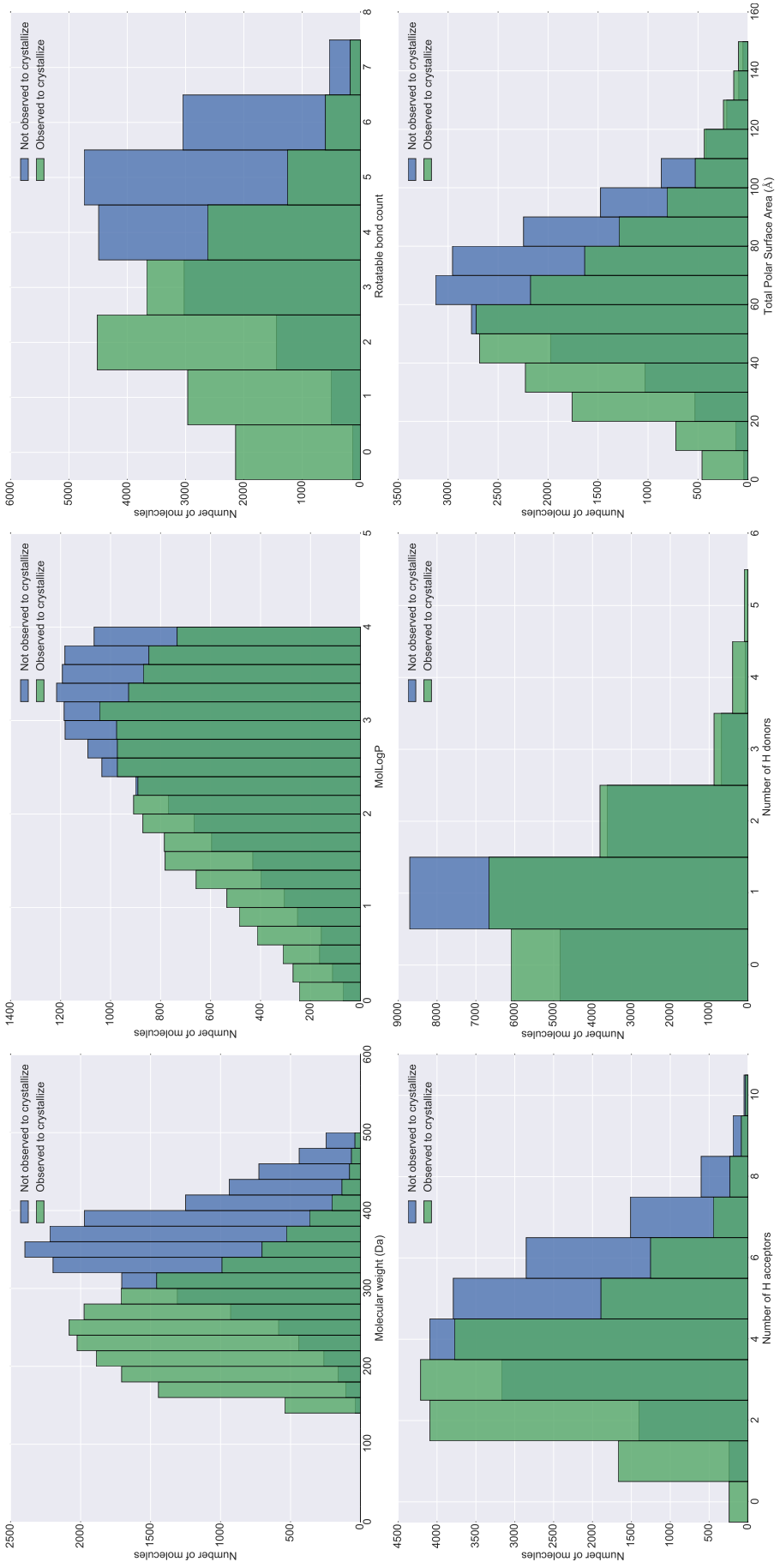


Figure 3.1 Histograms of key molecular descriptors for drug-like molecules.

3.1.2 Parameter tuning

The hyper-parameters were tuned using the 5-fold cross-validation method described in Section 2.3.2. A logarithmic search space was chosen to explore a wide spread of parameter values, allowing the full range of potential accuracies to be seen while ensuring the optimum values for each parameter are not on the edges of the search space.[172]

SVM with RBF kernel

Figure 3.2 shows the results of the hyper-parameter tuning for the SVM using an RBF kernel. A value of 1.0 for the γ parameter, which indicates a relatively small radius of influence, results in a poor mean predictive accuracy. This indicates that the radius of influence of each training point is too small relative to the size of the chemical descriptor space that the molecules exist in, resulting in an overfitted model. Decreasing γ increases the predictive accuracy up to a certain point, after which further reductions in γ causes the accuracy to decrease again. This indicates that the radius of influence of each training point has become too large, and so the model underfits the data.

An increased value of C gives a higher mean accuracy for most values of gamma. This can be attributed to the fact that the complex high dimensional chemical space requires a less smooth decision surface, with correct classification of all training points being desirable to obtain a well-fitted model.

Figure 3.3 displays the standard deviation of the predictive accuracies of the five models trained by the 5-fold cross-validation method. This demonstrates that a lower value of C tends to give greater variability in the predictive accuracy of the resulting model, possibly because a smoother decision surface where there is the potential for more misclassified training examples is more sensitive to variability between training datasets. Similarly, a very high C value causes the standard deviation to be increased, which could be a sign of overfitting.

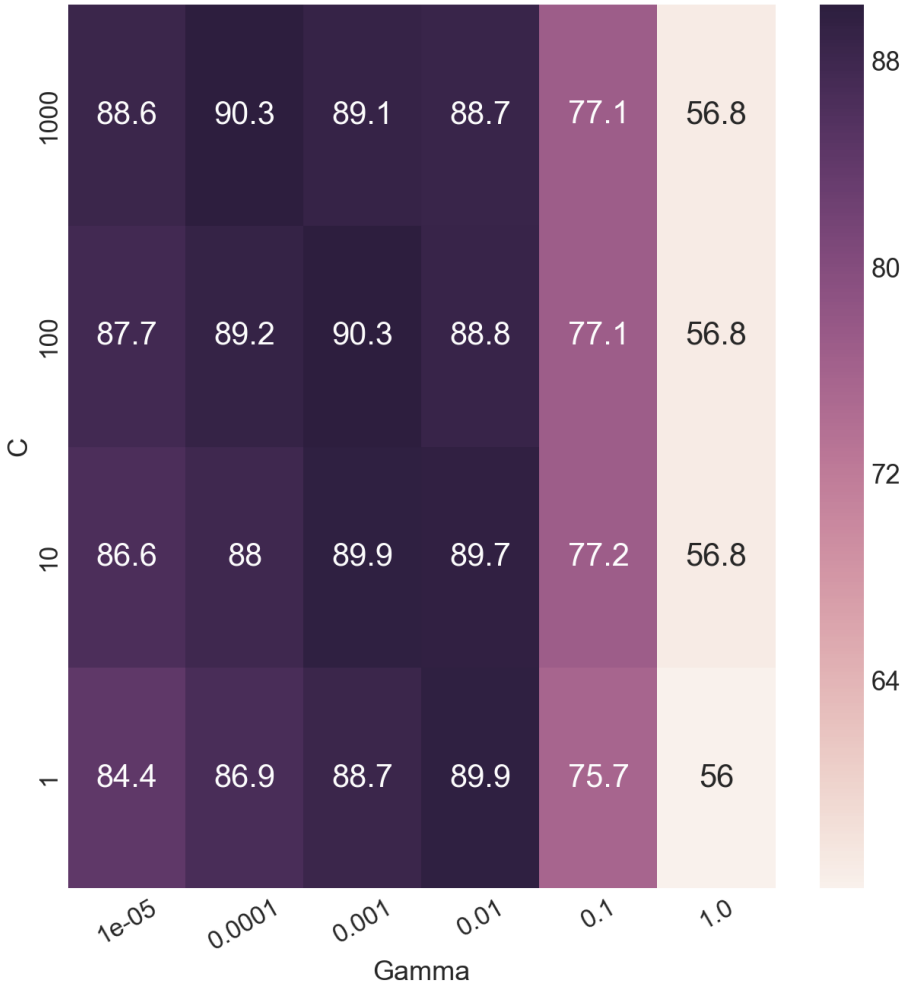


Figure 3.2 Mean predictive accuracies (expressed as percentages) obtained by 5-fold cross-validation on a grid-search of C and γ SVM hyper-parameters.

A lower γ value causes an increase in the standard deviation, which could be a result of the large radius of influence of each training point giving an underfitted model which would be influenced by variability across training sets. However, a γ which is too high and reduces the influence of each training point too much creates an overfitted model which again would be affected by some molecules being removed from or added to the training and test sets. Very high γ values provide a low standard deviation because the model has practically no predictive value, so each model performs equally poorly. This effect can be seen more clearly when comparing the average test accuracy with the mean accuracy on the training set, as in Figure 3.4. At high values of γ, the model is overfitted because the training set is perfectly modelled

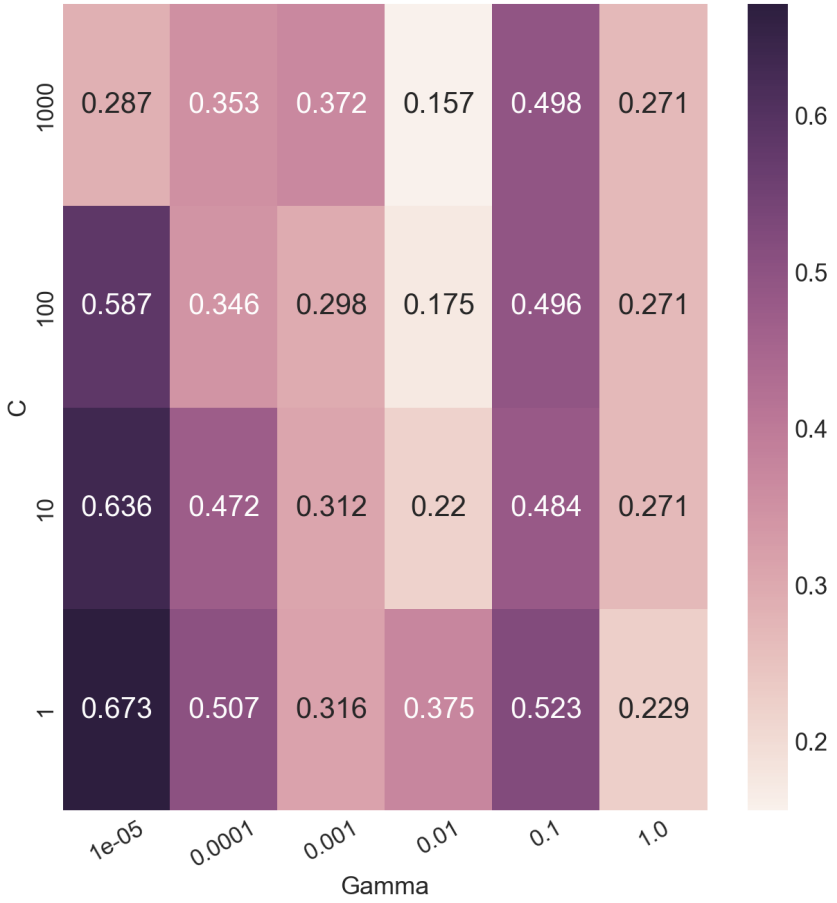


Figure 3.3 Standard deviation of the predictive accuracies obtained by 5-fold cross-validation on a grid-search of C and γ SVM hyper-parameters.

but this does not produce a model which performs well on the test set.

From these grid searches, sensible values of γ and C needed to be chosen to ensure that the resultant model was neither overfitted nor underfitted. This can be achieved by choosing a model with a high mean predictive accuracy but with as low a standard deviation as possible, thereby scoring highly and reproducibly. The highest mean predictive accuracy of 90.3% is obtained for two separate pairs of C and gamma. To minimise complexity of the resulting decision surface (thereby increasing the predictive capability of the model through reduced overfitting), the lower value of C was used, which also gave a lower standard deviation of 0.298% (compared to 0.353% for a parameter choice of C=1000 and gamma=0.0001). This led to a choice of 100 for the C parameter, and 0.001 for the γ parameter.

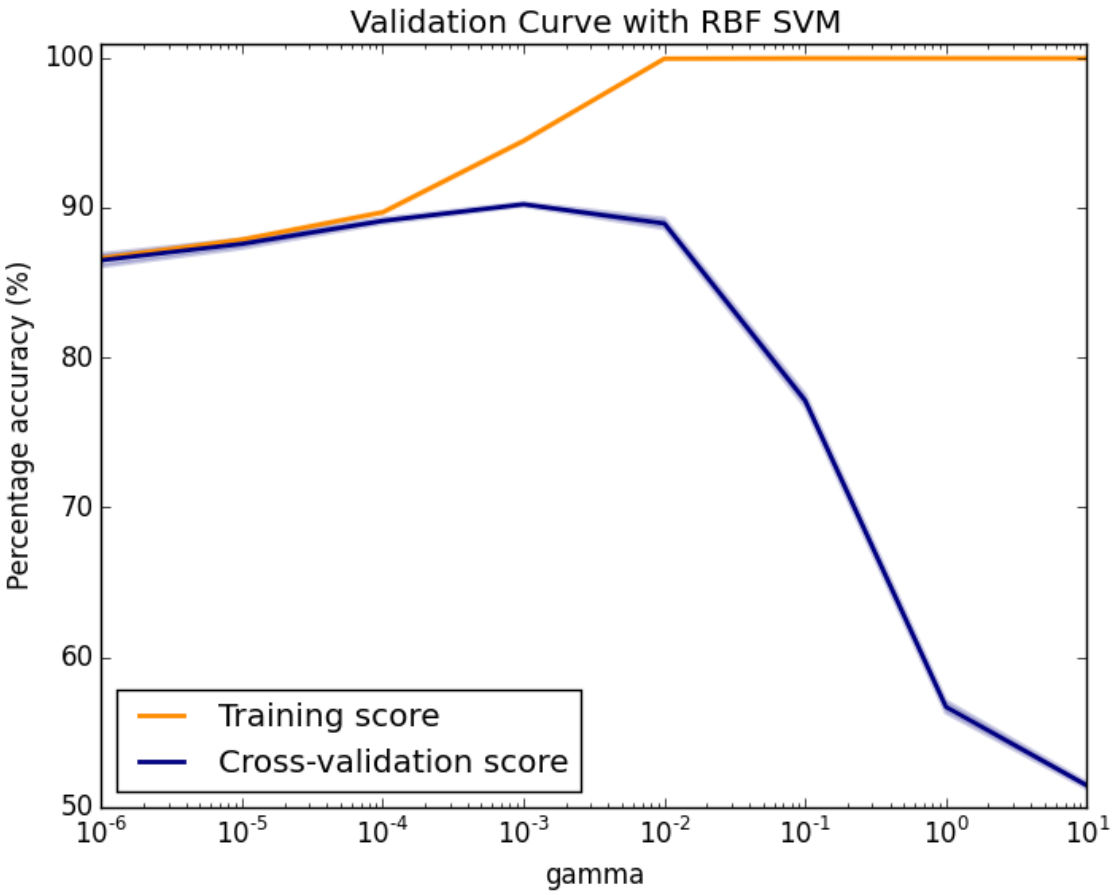


Figure 3.4 Mean predictive accuracies (expressed as percentages) obtained by 5-fold cross-validation with varying γ parameter for a C value of 100. Test and training accuracies are shown, with error bars representing the standard deviation of the five scores obtained from the cross-validation.

RF

The line graph in Figure 3.5 shows that as the number of trees used to build the random forest increases, there is a sharp increase in the mean accuracy from around 80% for 2 trees to 87% for 10 trees. There is then a small increase in accuracy to 88.9% as the number of tree estimators is increased further to 100. Beyond 100 trees, there is no further increase in predictive accuracy, although there is a slight improvement in the variability of the model, at a cost of greatly increased computational time. As a result, a value of 100 was chosen for the number of trees.

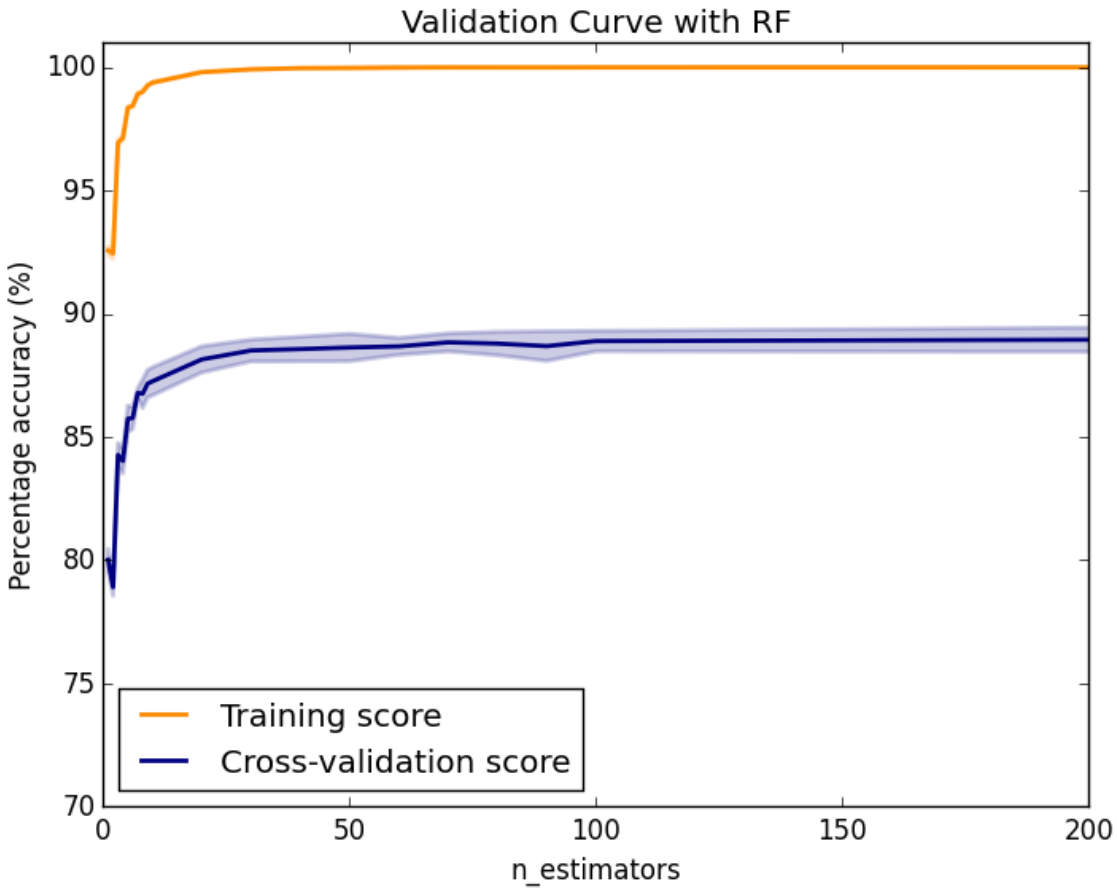


Figure 3.5 Mean predictive accuracies (expressed as percentages) obtained by 5-fold cross-validation with varying number of trees in the Random Forest. Test and training accuracies are shown, with error bars representing the standard deviation of the five scores obtained from the cross-validation.

SVM with linear kernel

Figure 3.6 shows that the best performing C value for the linear SVM in the cross-validation, both in terms of mean predictive accuracy and standard deviation of the accuracies, is a value of 1. As C increases, indicating that the penalty for misclassification of training molecules is greater, there is a dramatic tail off in predictive accuracy, with an accompanying increase in the variability. This is a result of overfitting of the model to the training data, which is particularly susceptible to variations in the training dataset because of the requirement for a linear separating hyperplane, the position of which will vary greatly depending on the training data points. The

overfitting also means that the model becomes less general and less applicable to unseen data points. As a result, a value of 1 was chosen for the C parameter.

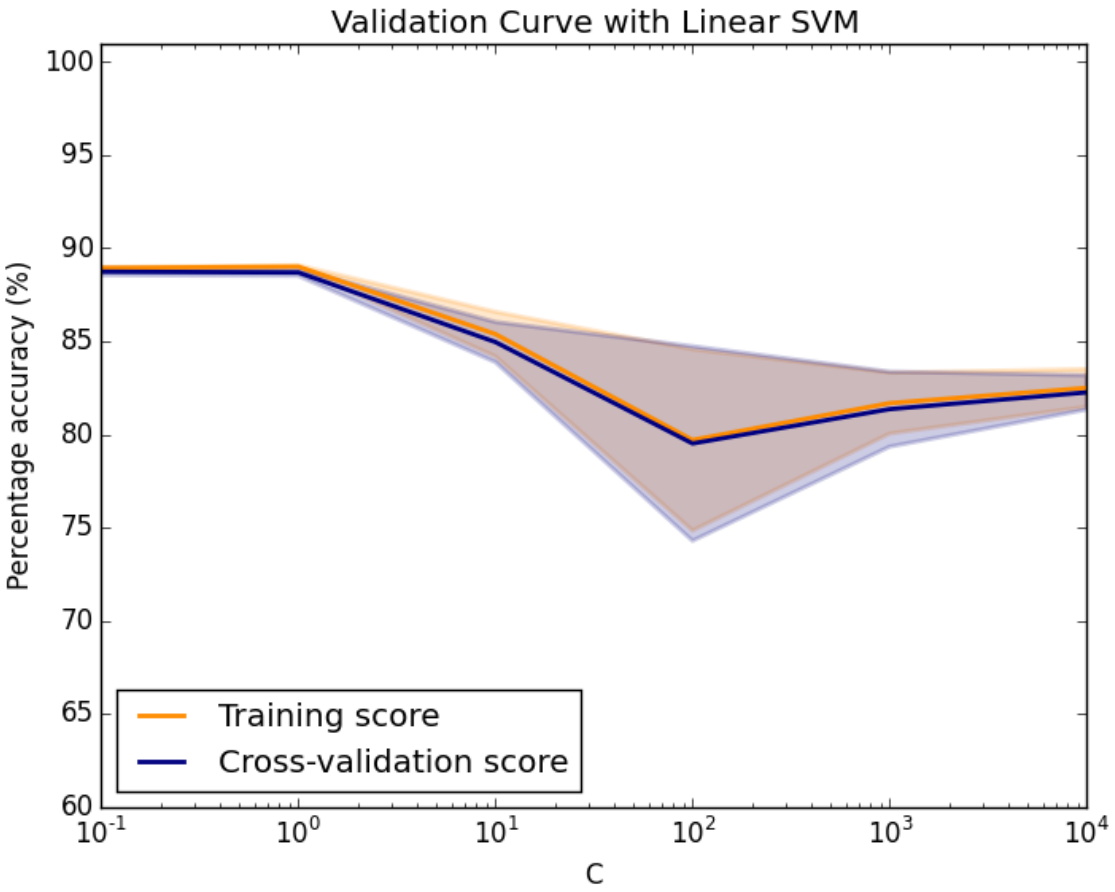


Figure 3.6 Mean predictive accuracies (expressed as percentages) obtained by 5-fold cross-validation with a logarithmic change in C value for the SVM with linear kernel. Test and training accuracies are shown, with error bars representing the standard deviation of the five scores obtained from the cross-validation.

3.1.3 Predictive accuracy

The predictive accuracy of the three machine learning algorithms trained using all 177 descriptors generated by RDKit, using the optimised parameters from 5-fold cross-validation, are compared below. The highest percentage accuracy was obtained by the model created using the SVM algorithm with an RBF kernel, as shown in Table 3.3. This model achieves classification accuracies of 90.3% on the drug-like data sets.

Table 3.3 Confusion matrices for models trained on the drug-like dataset with RDKit descriptors for a) Linear SVM b) RBF SVM c) RF. Key: T (NC) = Non-crystallizable correctly predicted; F (NC) = non-crystallizable but predicted to be crystallizable; F (C) = crystallizable but predicted to be non-crystallizable; T (C) = crystallizable correctly predicted. Overall predictive accuracy is provided, along with the cross-validation accuracy of the method from Section 3.1.2

Key		SVM (linear)		SVM (RBF)		RF	
T (NC)	F (NC)	**86.2%**	13.8%	**87.9%**	12.1%	**86.3%**	13.7%
F (C)	T (C)	8.8%	**91.2%**	7.2%	**92.8%**	9.0%	**91.0%**
Overall		88.7%		90.3%		88.6%	
Cross-validation		88.7(2)%		90.3(3)%		88.7(4)%	
Time (s)		17.3		808		15.5	

Confusion matrices in Table 3.3 show the SVM with RBF kernel misclassifies the fewest molecules for each class and is particularly accurate on the crystallizable dataset. The confusion matrices for the SVM classifiers show no significant imbalance in the misclassification between the two classes. Table 3.3 also shows that while the average from 5 tests for each of the algorithms is similar to the result from the single initial test, the variance in the value for the RF model is greater than for the SVM models, showing that SVM algorithms behave more consistently.

Figure 3.7 shows the ROC curves of the three different models trained using RDKit descriptors of drug-like molecules. Again, the SVM algorithm with an RBF kernel performs best. It has the highest AUC of 0.96 showing the most effective ranking of the molecules according to crystallinity and also has the steepest curve, showing good classification of molecules strongly predicted to be crystallizable.

Percentage accuracy, confusion matrices, and ROC curves all show that the SVM algorithm using an RBF kernel provides the best predictive model, which is consistent with the fact that this kernel is known to be widely used for its high predictive accuracy. The RDKit descriptors clearly capture much of the important information about which molecules will easily crystallize in a given set of molecules, and must be

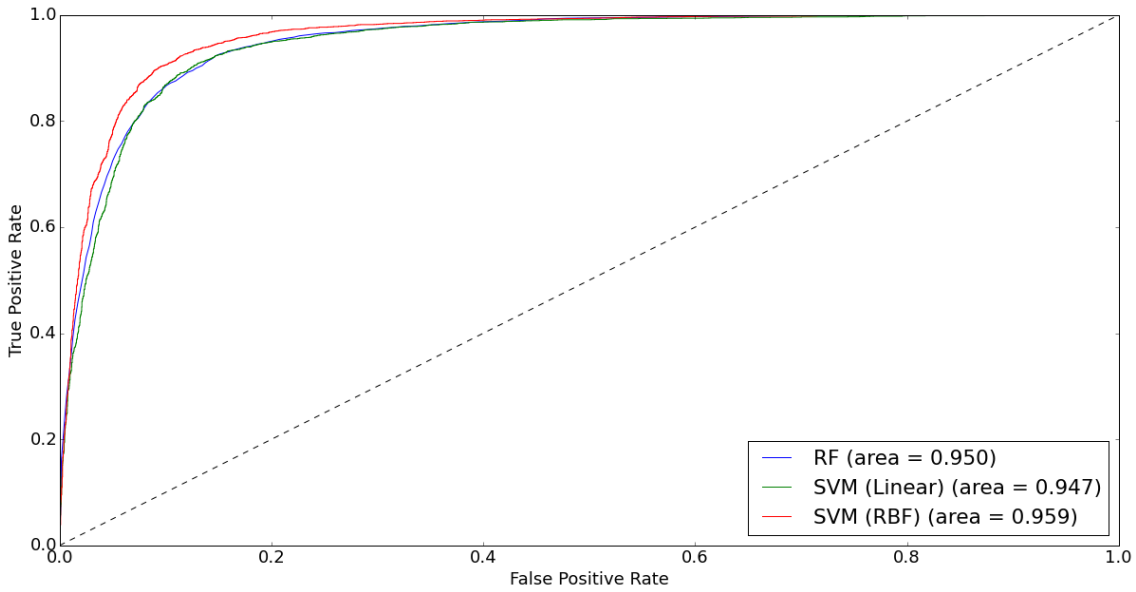


Figure 3.7 ROC curves for SVM (linear), SVM (RBF) and RF models trained using drug-like molecules with RDKit descriptors.

capturing underlying factors such as which materials exist in the solid state at all.

The SVM method using a linear kernel misclassified roughly 1% more of the molecules than the RBF kernel, due to the constraint on the linear algorithm to use only simple hyperplanes to separate the two classes. The confusion matrices show that the majority of the difference in failed predictions between RBF and linear kernels occur for the crystallizable class. However, an advantage of the linear algorithm is that it takes significantly less time to train than the RBF kernel.

The RF algorithm performed only slightly worse than the linear SVM on the test set judging by predictive accuracy, although the cross-validation accuracy and AUC are slightly higher. This, coupled with its ability to be trained using unscaled data and the relative speed of the training step could make this a useful algorithm for large scale calculations where accuracy can be traded off against speed.

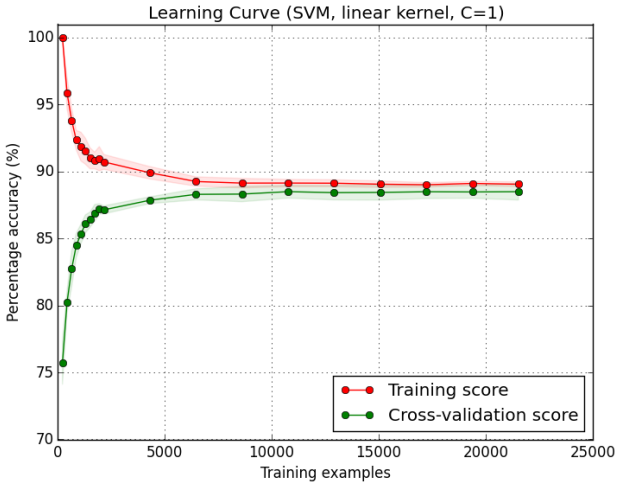
3.1.4 Learning curves

The SVM with linear kernel has a learning curve which shows a relatively high bias, as demonstrated by the poor training accuracy (below 90%) in Figure 3.8a. The variance

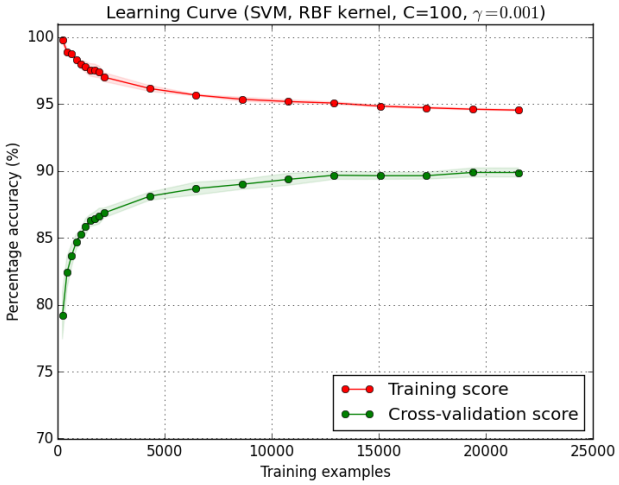
is high and the bias is low for very few samples, as the linear hyperplane can be drawn in many different places for a small sample size which successfully classifies the training data while generalising poorly. The bias increases and variance decreases rapidly with increasing training size. The variance shows no improvement after around 10000 samples, showing that the extra samples beyond this no longer provide any extra useful information to the algorithm.

The SVM with RBF kernel shows a similar low bias and high variance for very small sample sizes due to overfitting. However, at larger sample sizes it has a much lower bias than the linear kernel, showing that the model fits the data well. Although the variance does decrease as more data points are included in the training set, the variance is much greater than for the linear kernel, as shown by the large gap between training and test scores even for large training set sizes, indicating that there is still some overfitting of the data. This could be remedied by increasing the number of training samples, but this is limited by the amount of available data for this particular molecule filter, so removal of the filter could be required.

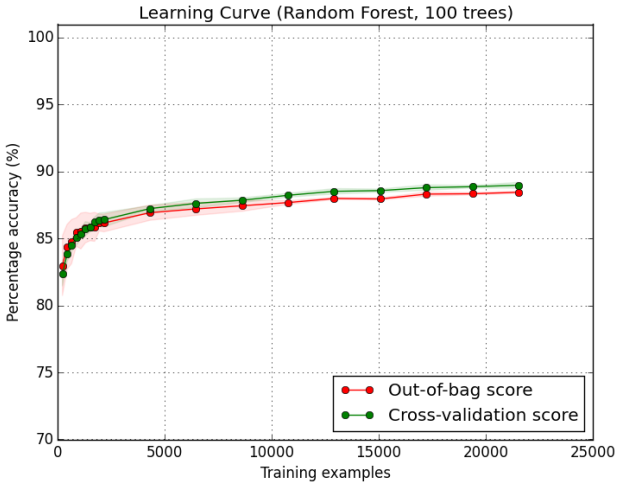
The learning curve for the RF model in Figure 3.8c shows that the out-of-bag training accuracy and the test accuracy are almost equal for all training set sizes, and the test accuracy actually exceeds the out-of-bag accuracy for larger training set sizes, with both scores increasing as the training set size increases. This indicates the variance is always low, while the bias is high for small training set sizes and decreases as the training set size increases. This is because the greater amount of training data results in deeper trees which are better able to represent the data. However, the bias is still relatively high compared to the SVM with RBF kernel.



(a) Linear SVM model.



(b) SVM model with RBF kernel.



(c) Random Forest model.

Figure 3.8 Learning curves for the models trained on the drug-like data. Mean out-of-bag training score and mean test score from 5-fold cross validation are shown, with error bars showing the standard deviation of the scores.

3.1.5 Descriptor Analysis

The C and γ parameters for the SVM algorithm were optimised using a grid search with cross-validation using accuracy as the score function for a single variable, and found to be $C = 1$ and $\gamma = 1$. Single variable classifier accuracies were calculated and ranked for each feature in the algorithm, and these are compared with the RF feature importances and linear SVM coefficients in Figure 3.9.

The number of valence electrons was found to be the most important feature for the RBF SVM, with an accuracy of 77.6%, but was not ranked as an important feature for either of the other two algorithms, while molecular weight was found to be the most important feature for the random forest classifier. However, the ${}^{0}\chi^{v}$ index was found to give the second highest single-variable cross-validation accuracy of 77.2%, and also has the coefficient with the second largest magnitude for the linear SVM separating hyperplane while being ranked 5th for importance by the random forest algorithm, so this was chosen as the most important descriptor. As discussed in Section 1.2.5, ${}^{0}\chi^{v}$ strongly correlates with molecular volume, so the importance of these two descriptors suggests size is a key factor in determining crystallization propensity.

By generating two variable classifiers using six of the top features and cross-validating their predictive accuracies on the training set, an improvement of 1-2 percentage points on the single variable classifiers, with predictive accuracies of between 78.5 and 80%, was obtained (Figure 3.10). This indicates that the majority of the accuracy obtained with 177 descriptors can be achieved using a single descriptor, with a small improvement on inclusion of a second descriptor. The most successful pair of descriptors was found to be ${}^{1}\kappa$ and SMR VSA3, giving a mean accuracy of 80%. This result shows the value of machine learning, as these two descriptors are not ones which would be identified as important by human intuition.

However, the standard deviations of the accuracies are relatively large and several other two variable classifiers also have comparable performance when this is taken

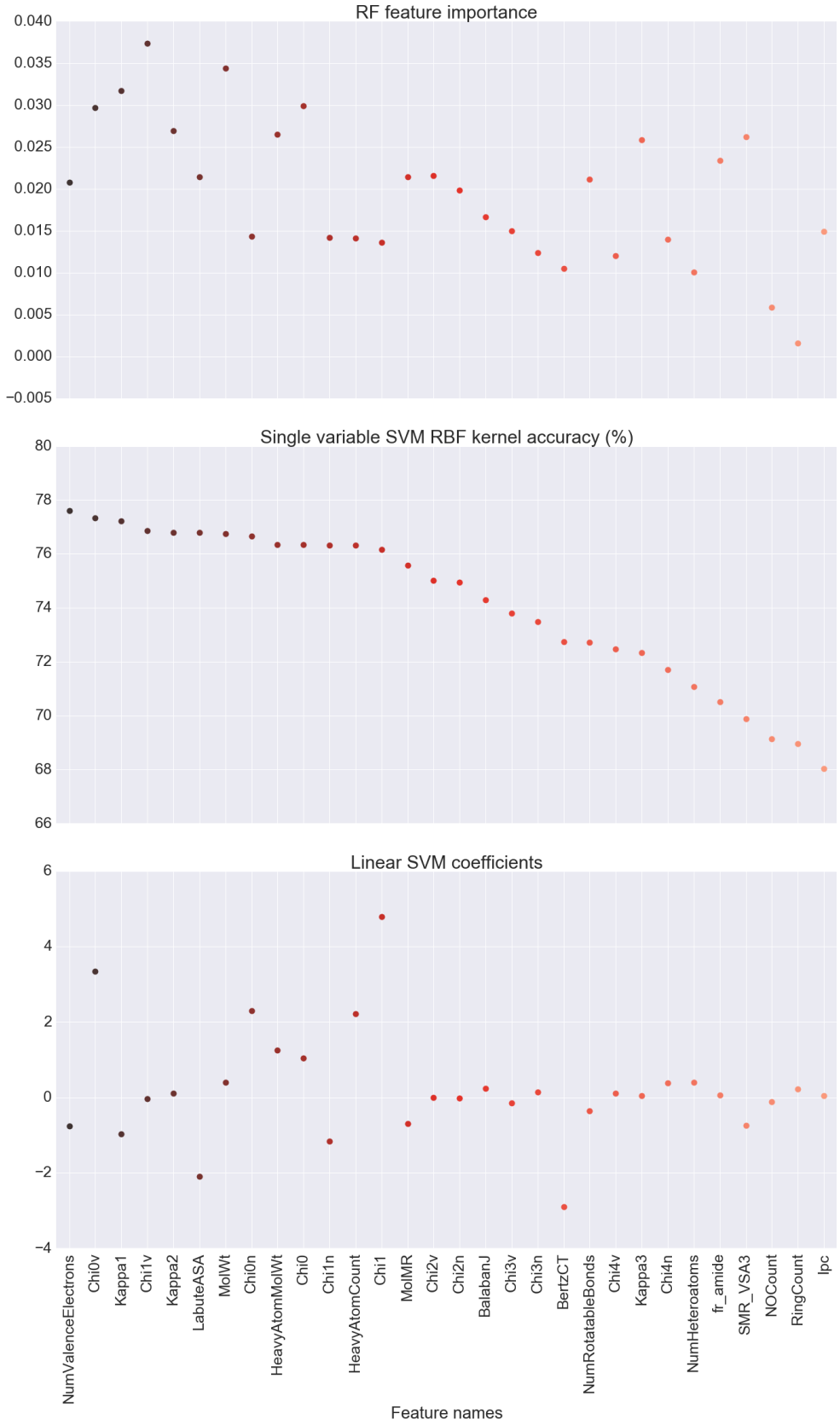


Figure 3.9 Feature importances for a) RF b) SVM with RBF kernel c) linear SVM, using the drug-like data, ranked and colour-coded by single variable SVM RBF accuracy

into account. There is a clear indication of the importance of using rotatable bond count (RBC) in combination with a descriptor of the size of the molecule, since RBC appears in the top 4 for each of the three descriptors associated with the size of the molecule ($^0\chi^v$, molecular weight and number of valence electrons). This shows that a combination of descriptors of size and flexibility is important for determining crystallization propensity.

This is backed up by the decision tree analysis of the RBF SVM model in Figure 3.11, which shows that $^0\chi^v$ and RBC are both important descriptors as they appear in the second and third nodes of the tree respectively, while the first node in the tree is a split using the number of valence electrons. This provides the best initial split of the data and therefore indicates the most important classification feature for this particular algorithm; the best single-decision approximation of the RBF SVM can be obtained by assuming that the majority of molecules with fewer than 109 valence electrons are observed to crystallize.

Following the tree to the right, the "non-crystallizable" side of the split, the next split, by whether the molecule contains amide groups, provides a set which is roughly a third of the test set in which molecules are large and contain at least one amide group, where 90% of the molecules are found to be "non-crystallizable". Conversely, on the left, "crystallizable", side of the valence electron split, a further split of $^0\chi^v$ (below 10.5) gives a set which is roughly a third of the entire test set in which the model would predict over 90% of the molecules to be "crystallizable", after which a split of fewer than three rotatable bonds is required to continue the tree.

The distribution of $^0\chi^v$ and RBC for the crystallizable and non-crystallizable materials in the test set is shown in Figure 3.12, with decision boundaries plotted for both the linear SVM and the RBF SVM. The increase in accuracy from linear SVM to RBF SVM is not significant and this can be understood by looking at the decision boundary, which is found to be almost linear even when using a non-linear kernel. The extra complexity of the RBF SVM decision boundary allows it to successfully

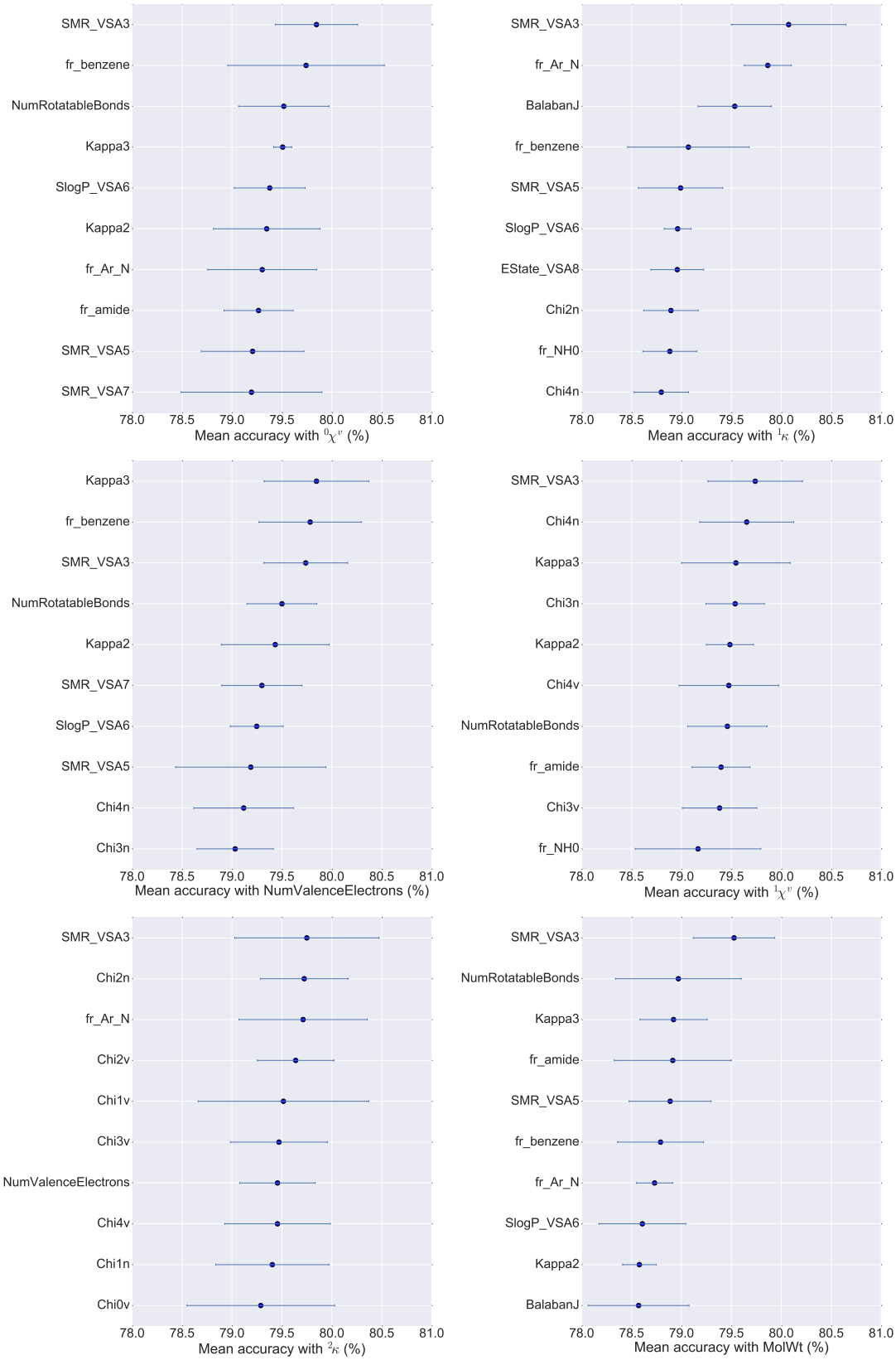


Figure 3.10 Mean predictive accuracy by cross-validation on two variable classifiers trained on the drug-like data for each RDKit feature with six of the best-performing single RDKit features, with error bars showing the standard deviation of the cross-validation scores.

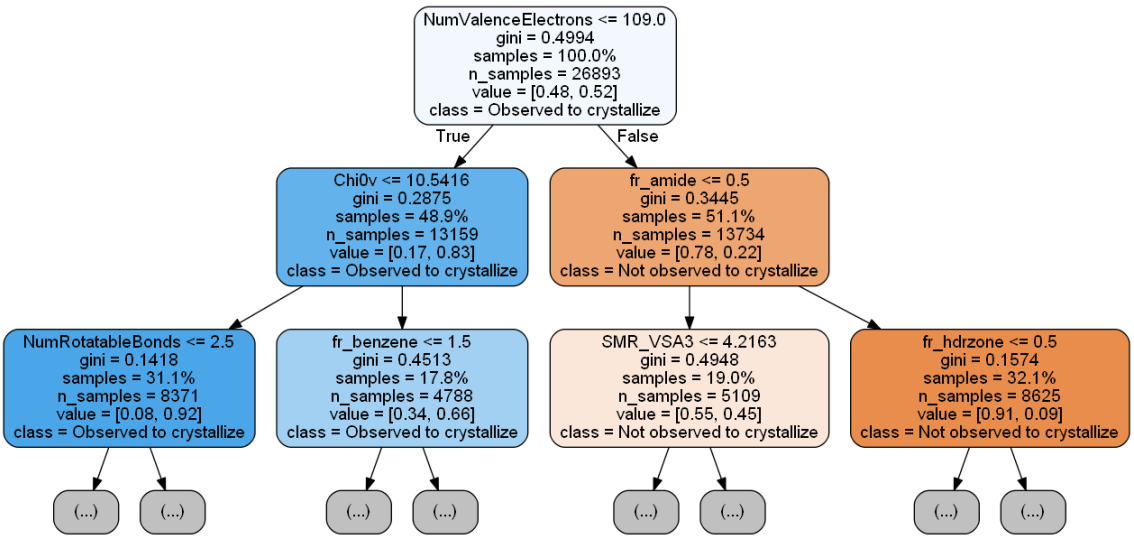


Figure 3.11 Decision tree used for rule extraction for the models trained on the drug-like data with RDKit descriptors (top 3 levels shown). The gini coefficient is a measure of the impurity of the node. "Samples" indicates the percentage of the total dataset present at that node, and "value" is the proportion of "not observed to crystallize" (orange leaves) and "observed to crystallize" (blue leaves) molecules at the node. Each node has been assigned an overall class based on these proportions.

classify around 1% more of the crystallizable molecules than the linear hyperplane.

From the heatmap it can be seen that there is a slight positive linear correlation between ${}^{0}\chi^{v}$ and the number of rotatable bonds of 0.59. The decision surfaces separate the two classes along a line which runs approximately perpendicular to the direction of the correlation, allowing an improved classification of otherwise overlapping data.

The non-crystallizable molecules are concentrated in a region with an RBC of 4-7 and a ${}^{0}\chi^{v}$ value of 12-17, while the crystallizable molecules mostly occupy a slightly more spread out region of both lower RBC and lower ${}^{0}\chi^{v}$. The line obtained from the SVM algorithm seems to effectively distinguish between the majority of crystallizable and non-crystallizable molecules, validating the discriminatory importance of these two descriptors. However, there is significant overlap between the classes at the centre of the graph, at ${}^{0}\chi^{v}$ values of 11-13 and RBC values of 3-5. It is in these cases where more descriptors are necessary to improve the predictive accuracy of the

calculation.

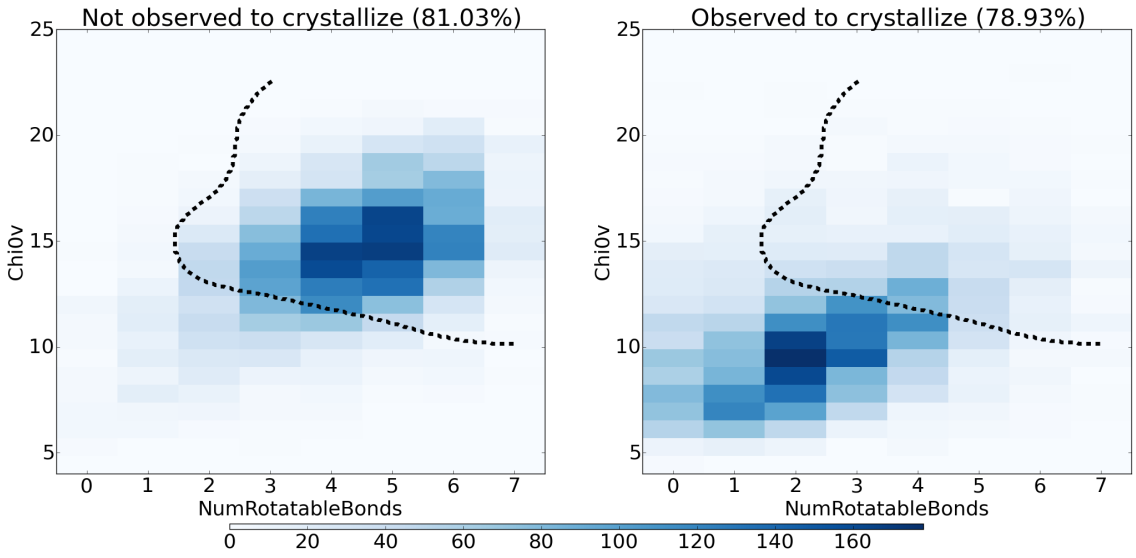
The ${}^0\chi^v$ index can be thought of as a simple descriptor of the size of the molecule as it has a correlation coefficient for the training set of 0.96 with molecular weight and 0.97 with number of valence electrons. This explains why all three of these descriptors are found to be important by at least one of the algorithms and suggests size has a key influence on the crystallization propensity. The propensity of molecules with a low ${}^0\chi^v$, and therefore a lower molecular volume, to crystallize could be attributed to the greater ease with which solvent molecules around smaller solute molecules can rearrange to allow access to the surface on crystal growth.[188]

The influence of the rotatable bond count on the tendency to crystallize supports the theory proposed in Section 1.1.2, that a more conformationally flexible molecule will have a reduced tendency to crystallize due to the lower supersaturation ratio caused by the presence of conformers other than the crystallizing conformer. In general, the more rotatable bonds a molecule has, the greater the number of potential conformers that will exist for that molecule. Hence a molecule with fewer rotatable bonds, and therefore a higher concentration of the crystallizing conformer, will have a greater propensity to crystallize.

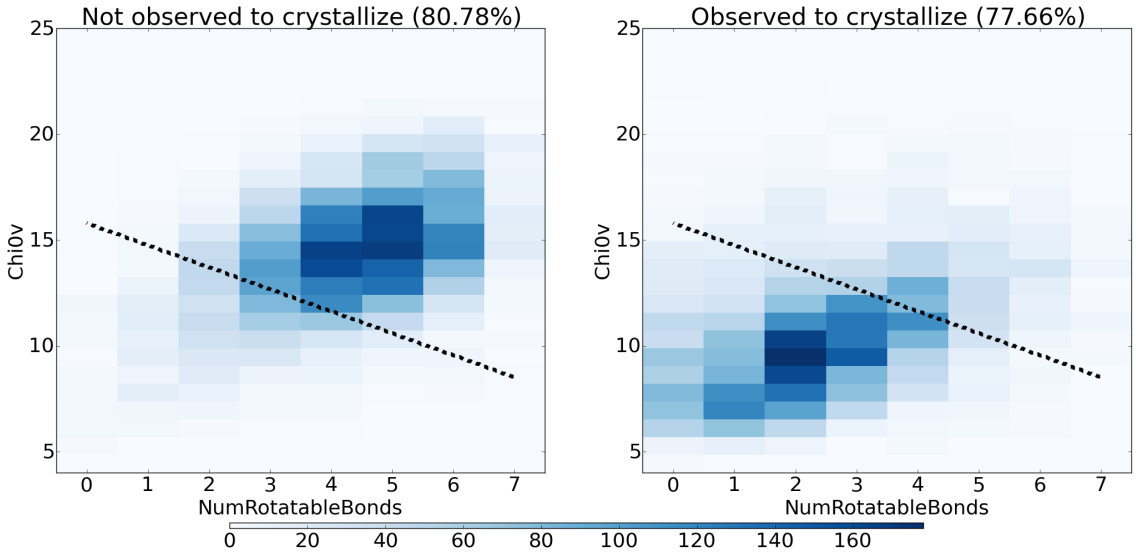
Test set descriptor balancing

To investigate the effect of removing the ability of the model to classify using the two principal variables, a test subset was created using molecules where the values of both of these descriptors was constant. The numbers of "crystallizable" and "non-crystallizable" molecules were balanced by choosing descriptor values for which there were similar numbers of each class in the dataset, which on inspection of the histograms in Figure 3.13 led to choosing ${}^0\chi^v$ to lie in a narrow range of 12–13 (as it is a continuous variable) and a rotatable bond count of 3.

The accuracy of the model on this subset of 254 molecules was 85%. This is a decrease in accuracy compared to the full test dataset, as expected when preventing



(a) RBF SVM



(b) Linear SVM

Figure 3.12 Distribution of rotatable bond count against ${}^{0}\chi^{v}$ for all test molecules in the drug-like dataset, colour-coded by density of molecules. The dashed line shows the boundary between the crystallizable and non-crystallizable regions as predicted by a) RBF SVM b) linear SVM.

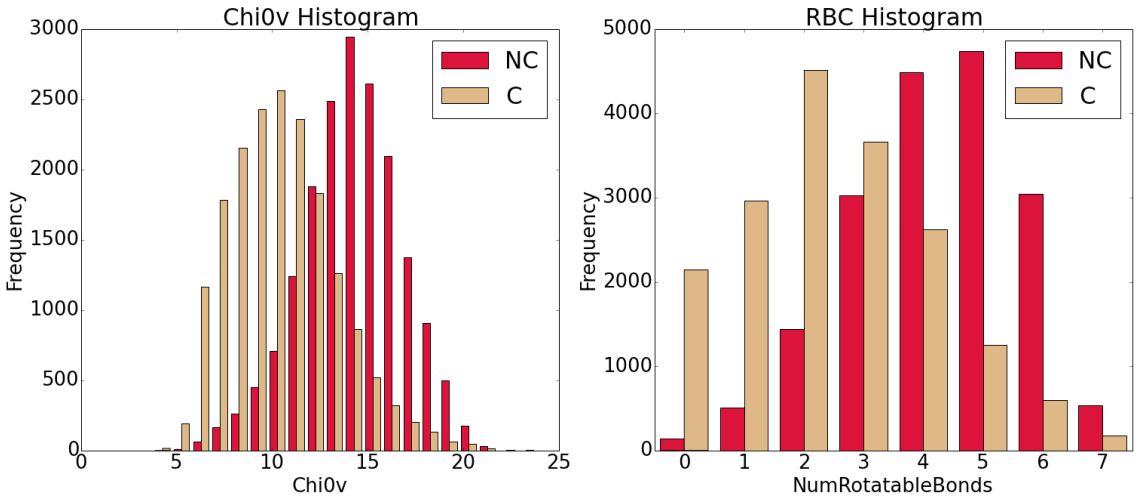


Figure 3.13 Distributions of ${}^0\chi^v$ and rotatable bond count for all molecules in the drug-like dataset. The subset was chosen to include molecules with a ${}^0\chi^v$ between 12 and 13 and a rotatable bond count of 3. Note that ${}^0\chi^v$ is a continuous variable and molecules are divided into bins of width 1.0 to produce this plot.

the algorithm from classifying using the two most important descriptors. However, this does show that there is appreciable predictive accuracy to be obtained from some of the 175 other descriptors, because the different information they provide is in fact useful for classification in the region of chemical space where the values of the top two descriptors overlap for both "crystallizable" and "non-crystallizable" molecules, as shown in Figure 3.12.

3.1.6 CSD update validation

In order to further validate the use of the approach using another independent test dataset, the CSD update for February 2014 was used to provide a second set of "crystallizable" molecules which could be used to test the predictive accuracy of the model. This set was filtered in the same way as the original test set, and any molecules which were already present in the CSD were removed. Only those which were already present in ZINC (but had not been used previously in either the test or training datasets) were included. This gave us a set of 354 new crystallizable molecules independent of the initial training and test datasets, which were then classified using the original

model. Of these, 312 molecules were successfully predicted to be crystallizable, giving a classification accuracy of 88%, which is very similar to the accuracy obtained from the original test data. The plot of the two most important descriptors as found by the single-variable classifier method shows that the update molecules (Figure 3.14) have a similar distribution in this chemical space to the crystallizable molecules in the original test set (Figure 3.12).

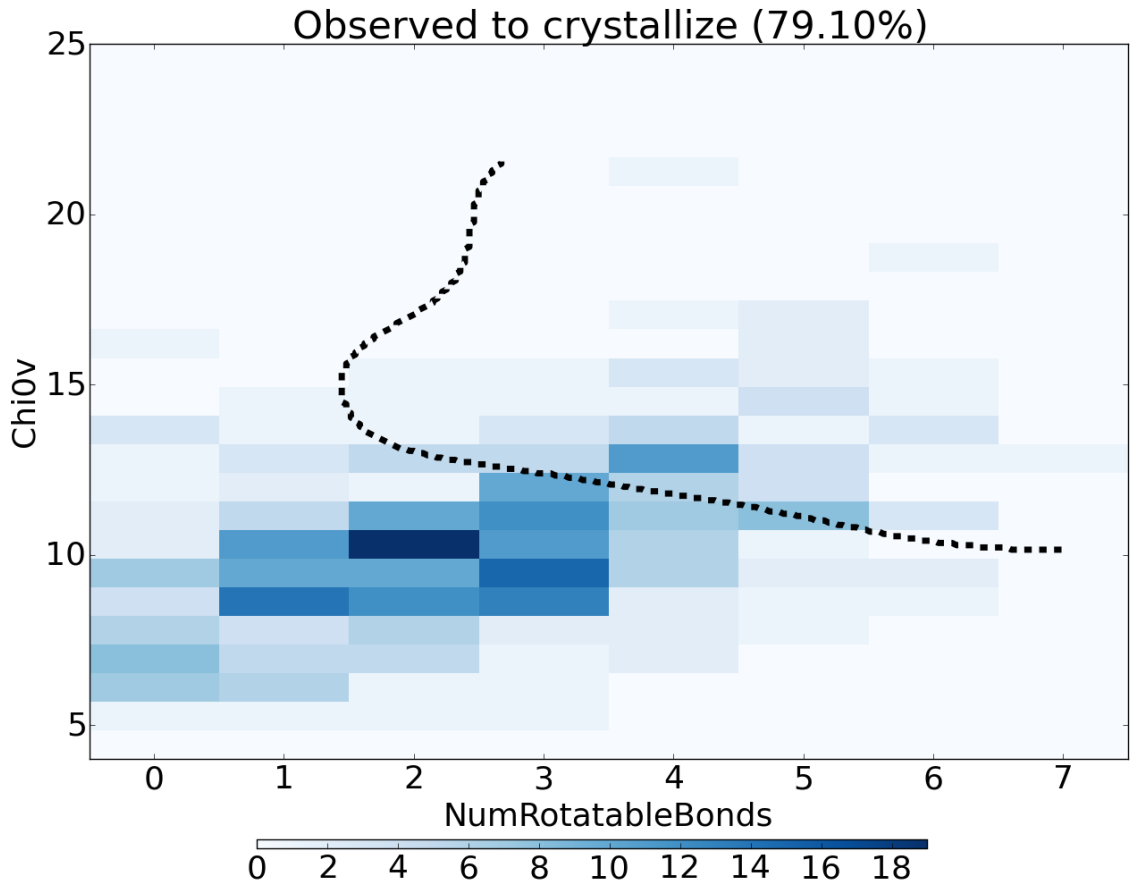


Figure 3.14 Distribution of rotatable bond count against ${}^{0}\chi^{v}$ for the February 2014 update of the CSD, colour-coded by density of molecules. The dashed line shows the boundary between the crystallizable and non-crystallizable regions as predicted by the SVM algorithm using RBF kernel.

3.1.7 Conclusions

This section demonstrates that a machine learning approach can be used to train a classification algorithm to predict whether a molecule may be apt to form crystals,

with an accuracy of 90.3% on an unseen external test set. The best parameters were found to be C=100 and γ = 0.001 for the SVM algorithm with RBF kernel, C=1 for the linear kernel SVM, and 100 trees for the RF algorithm. The comparison of these three machine learning algorithms shows that the SVM algorithm with RBF kernel provided the best model.

Only a few features of a molecule dominate its propensity to crystallize, and these can be identified by using single variable classifiers. The approach has been validated against a set of crystallizable molecules from an update to the CSD, which has been demonstrated to have a similar spread of descriptors to the crystallizable set of molecules. However, the relatively high variance of the best performing model suggests that there is room for improvement, which could be achieved by expanding the size of the training set.

3.2 Model improvement

3.2.1 Introduction

There are a number of improvements that can be made to the model. The two most important of these are the input molecules that are used to train the algorithm, and the descriptors which are used to represent these molecules.

The quality of the data was first improved by modifying the method of extraction of molecules from the CSD. By using the CSD Python API, the manual parsing of SMILES strings was avoided. The definition of the "main component" of the crystal structure by the CSD ensured a more robust method of identifying the molecule of interest than simply sorting by number of heavy atoms in the SMILES string. The updated version of the CSD from the November 2013 to the November 2015 version provided a more complete set of molecules from which to work with.

The cross-referencing and duplicate removal was improved significantly with the generation of canonical tautomers for each molecule, as described on p 60, Section 2.1.1. When applying this approach on the original drug-like set, 23 duplicates were discov-

ered in the original "non-crystallizable" set, 83 duplicates were found in the "crystallizable" set, and there were two molecules which were present in both sets. These duplicates correspond to approximately 0.3% of the overall dataset, and so are unlikely to have much effect on the performance of the algorithm other than slightly overestimating the predictive accuracy, since molecules which are present in both the training and test sets are likely to be correctly predicted. However, the possibility of missing molecules due to the existence of inequivalent tautomers in the two databases means that the improved cross-referencing should result in a larger "crystallizable" ZINC set, allowing more information to be incorporated into the model.

This dataset with improved curation was used to investigate the effects of a) filtering the molecules and b) changing the descriptors used to represent the molecules.

3.2.2 Effect of dataset

The effect of varying the dataset was studied on this more thoroughly curated overall set of molecules, using the standard RDKit descriptors. Balanced sets of training and test molecules were prepared with no filter, to give the distribution of training and test molecules shown in Table 3.4. The drug-like filter was applied to generate a model for comparison with the original model, to identify the improvement obtained by the improved curation. The filter was then removed to assess the effect of increasing the dataset size on the predictive accuracy. Finally, the two classes were balanced by molecular weight to ensure that predictive capability was not simply due to the larger size of the molecules in ZINC relative to the CSD molecules found in ZINC.

Table 3.4 Breakdown of training and test molecules for the updated unfiltered dataset.

	Non-crystallizable	Crystallizable	Total
Training	20032	19953	39985
Test	6678	6652	13330
Total	26710	26605	53315

Drug-like

The drug-like filter was applied to both the training and test sets independently. Around 2000 more of the crystallizable molecules than the non-crystallizable ones were rejected by this filter, as illustrated in Figure 3.15, which is interesting because it suggests that the non-crystallizable molecules generally have a more drug-like character according to Lipinski's rules than the crystallizable ones. The training set was rebalanced by randomly removing molecules from the larger set, so that each class contained equal numbers of molecules to remove any bias in the training of the algorithm. The filtered test set molecules were all kept, to give the training and test set sizes shown in Table 3.5.

Table 3.5 Breakdown of training and test molecules for drug-like molecules in the updated dataset.

	Non-crystallizable	Crystallizable	Total
Training	15560	15560	31120
Test	5777	5200	10977
Total	21337	20760	42097

Table 3.6 shows that this more thorough database curation and removal of duplicates causes an increase in percentage accuracy for every algorithm tested, with an increase of over 5 percentage points for the best performing algorithm. The highest performance is still obtained by the SVM algorithm with RBF kernel, although this model comes at the cost of taking over 20 times longer to train than the other two

Table 3.6 Confusion matrices for models trained on the updated drug-like dataset with RDKit descriptors.

Key		SVM (linear)		SVM (RBF)		RF	
T (NC)	F (NC)	**94.4%**	5.6%	**95.2%**	4.8%	**93.6%**	6.4%
F (C)	T (C)	6.1%	**93.9%**	4.2%	**95.8%**	5.9%	**94.1%**
Overall		94.2%		95.5%		93.8%	
Cross-validation		94.6(3)%		95.9(3)%		94.3(1)%	
Time (s)		18.6		478		21.7	

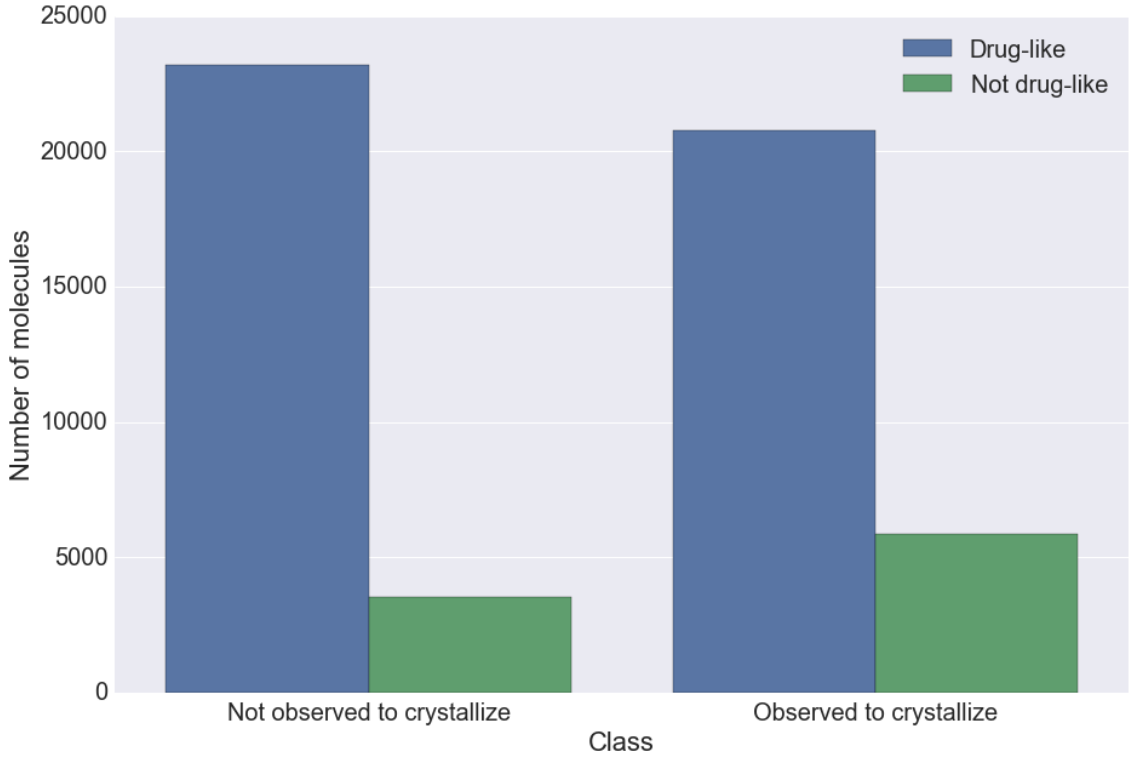
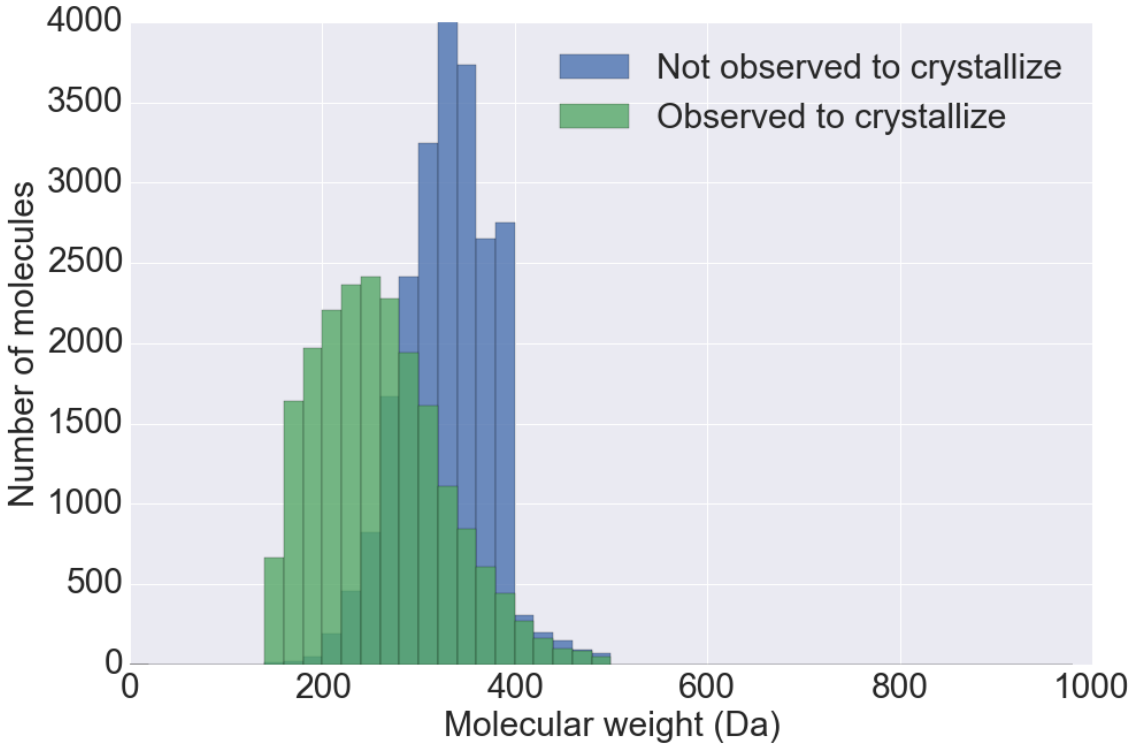


Figure 3.15 Counts of drug-like and non-drug-like molecules for the two classes (before training set balancing).

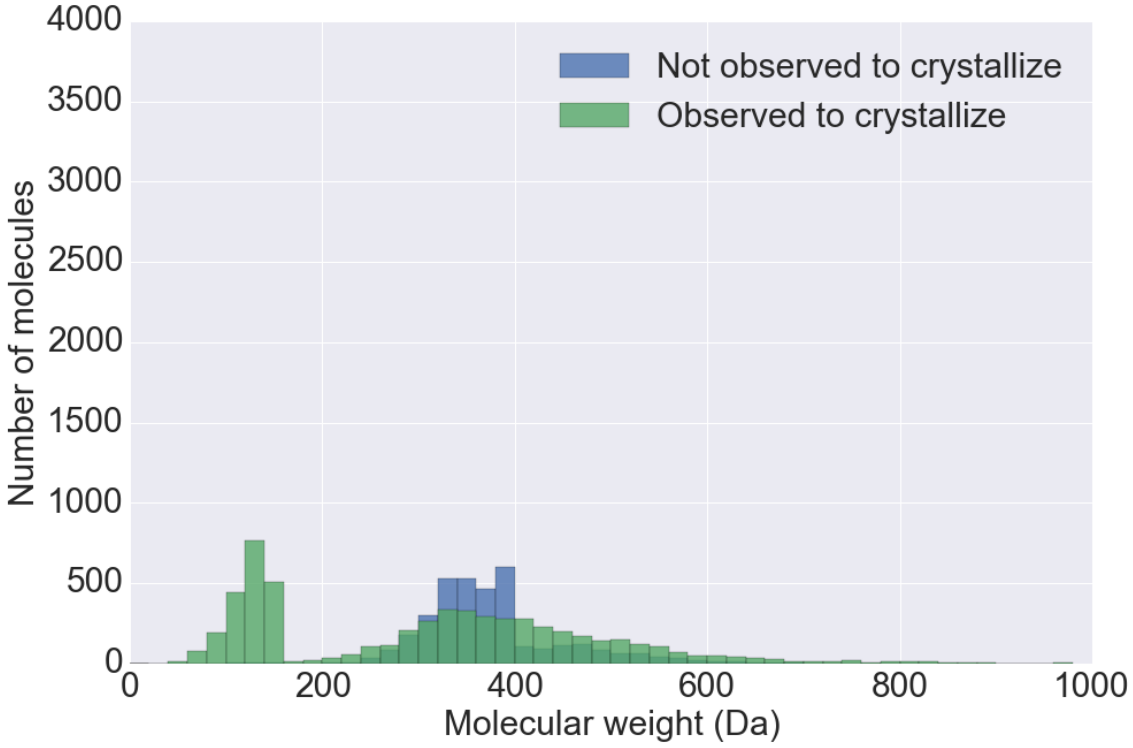
algorithms.

The linear SVM sees a large improvement in specificity, but a smaller increase in recall, meaning that this model performs slightly worse on the crystallizable set than the non-crystallizable set; overall the predictive accuracy increases by around 6 percentage points. This algorithm is also quick to train, although the random forest model has a similar training speed and provides comparable predictive accuracy without the need for scaling the data beforehand. The SVM with RBF kernel still gives the greatest accuracy, and the smallest imbalance in recall and specificity, suggesting that this is still the best algorithm to use.

One reason for the greater number of non-drug-like crystallizable molecules may be that the smaller size of these molecules means that many fall below the filter size of 150 Daltons. This can be seen in Figure 3.16b, where the number of crystallizable molecules which are classed as non-drug-like due to their small molecular weight is much greater, with hardly any non-crystallizable molecules falling into this category.



(a) Drug-like



(b) Non-drug-like

Figure 3.16 Molecular weight distribution by class for a) drug-like and b) non-drug-like molecules.

Table 3.7 Confusion matrix for a non-drug-like test set from prediction by a model trained on drug-like molecules.

Key		SVM (RBF)	
T (NC)	F (NC)	**88.2%**	11.8%
F (C)	T (C)	2.5%	**97.5%**
Overall		93.9%	

If we use a model trained on the drug-like data to predict the labels of the non-drug-like portion of the test set, the predictive accuracy decreases by 2 percentage points, as shown in Table 3.7, and this is mostly due to the reduction in specificity. This is presumably because the model has reduced capability to predict on this set as it is outside the domain of applicability of the model due to the filtering process.

This investigation shows that the improved curation and cross-referencing process roughly halves the error rate for the best performing model, which is a significant improvement. It would, however, be preferable to generate a model which can predict on all molecules in addition to the drug-like ones.

All molecules

The training and test set distribution for the set with no filtering is given in Table 3.4, showing both the training and test sets are balanced by class. The confusion matrices shown in Table 3.8 demonstrate that although the number of test molecules has increased by around 25%, there is little appreciable increase in predictive accuracy.

Table 3.8 Confusion matrices for models trained on the unfiltered dataset with RDKit descriptors.

Key		SVM (linear)		SVM (RBF)		RF	
T (NC)	F (NC)	**94.1%**	5.9%	**95.4%**	4.6%	**93.9%**	6.1%
F (C)	T (C)	7.2%	**92.8%**	4.3%	**95.7%**	5.5%	**94.5%**
Overall		93.5%		95.6%		94.2%	
Cross-validation		93.9(31)%		95.9(1)%		94.7(2)%	
Time (s)		23.2		971		28.4	

However, if the *test* set is split back into drug-like and non-drug-like sets it can be seen that, due to the improvement in specificity for the non-drug-like set, there is now little difference between the filtered and unfiltered sets, as shown in Table 3.9. The extra information provided to the model by the enlarged training set has successfully extended the domain of applicability to all molecules without compromising the predictive accuracy on the drug-like set.

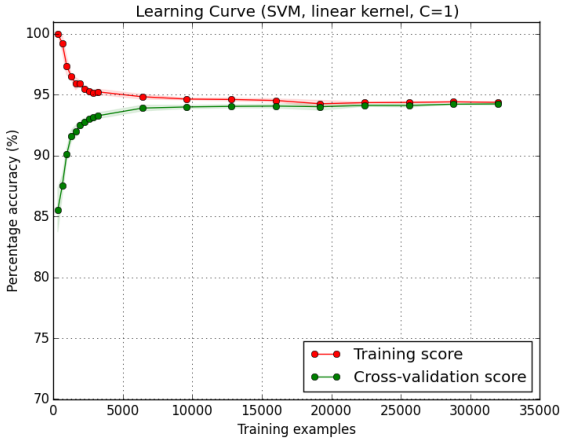
Table 3.9 Confusion matrices of a) the drug-like test set b) the non-drug-like test set from prediction by a SVM model with RBF kernel trained on the unfiltered dataset using RDKit descriptors.

Key		Drug-like		Non-drug-like	
T (NC)	F (NC)	**96.5%**	3.5%	**94.5%**	5.5%
F (C)	T (C)	4.3%	**95.7%**	3.1%	**96.9%**
Overall		96.1%		96.0%	

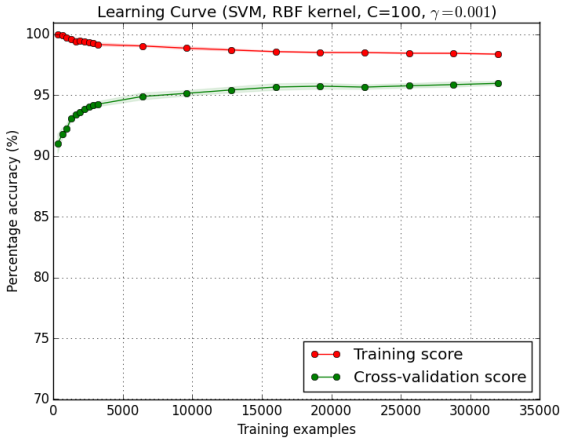
The learning curves in Figure 3.17 demonstrate that the bias has decreased for all three algorithms as a result of the better curation of the databases. The biggest improvement in bias is seen for the random forest model, which now shows smaller bias than the linear SVM. The increase in the number of training data points by removing the drug-like filter has also caused a decrease in variance, most notably for the RBF SVM model. However, there is still some variance, suggesting that there is a small amount of improvement still to be attained in order to create the best possible model.

The learning curves for RF and RBF SVM are much shallower than for the drug-like set, as accuracy is already relatively high even for a small number of training points. This suggests that the extended training data captures the difference between the two datasets more thoroughly and allows the classification to succeed with fewer input points.

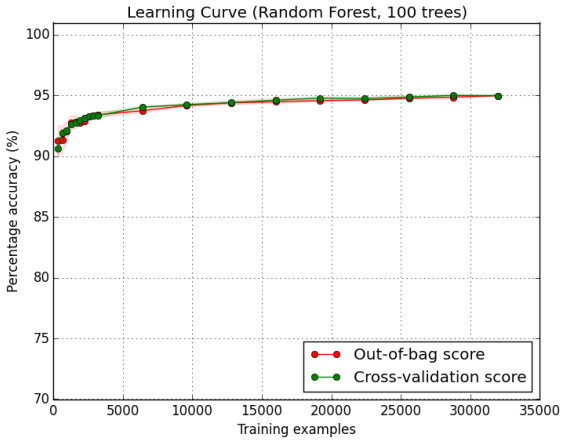
The feature importances in Figure 3.18 show some differences compared with those for the initial drug-like models. The number of valence electrons is still an important factor for both the RF and RBF SVM models, and the zero order connec-



(a) Linear SVM model.



(b) SVM model with RBF kernel.



(c) Random Forest model.

Figure 3.17 Learning curves for the models trained on the unfiltered data set. Mean out-of-bag training score and mean test score from 5-fold cross validation are shown, with error bars showing the standard deviation of the scores.

tivity indices ($^0\chi$, $^0\chi^v$, $^0\chi^n$), which are all very closely related, each have high importance for at least one of the algorithms. Furthermore, the shape index $^1\kappa$, which has a correlation coefficient of 0.98 with the number of valence electrons, is the third most important feature for the RF and linear SVM models, and produces the second highest predictive accuracy for a single variable classifier. This suggests molecular size is still an important predictor, yet the best performing single-variable classifier is given by the number of amides, which only has a correlation coefficient of 0.32 with the valence electron count, as well as being the second most important variable for the RF model, a factor that was only previously discovered by the decision tree rule extraction method for the initial model.

The decision tree rule extraction analysis in Figure 3.19 supports this, showing that the best initial split of the data is now obtained by using the number of amide groups as the splitting criterion. Around 80% of the accuracy of the RBF SVM can be achieved by assigning molecules with no amide groups as being crystallizable. Flexibility is still an important feature of the molecule, since rotatable bond count is used as the next split in this side of the tree, with any molecules containing fewer than 4 rotatable bonds being classed as "crystallizable". The importance of size has decreased, as the only descriptor related to this that appears in the first three nodes of the tree is the number of valence electrons.

Interestingly, Figure 3.20 shows that amide count does not combine well with other descriptors, only achieving a maximum increase of 3.8 percentage points by using it in a feature vector with $^1\chi^v$. The best performing two variable classifier is obtained by using a feature vector of $^1\kappa$ and SMR VSA3, as discovered for the initial model in Section 3.1, but now the cross-validation accuracy is 85.7%, an increase of nearly 6 percentage points on the original model, and an increase of 8 percentage points compared to the best performing single variable classifier. This is despite the fact that SMR VSA3 only gives the 18$^{\text{th}}$ best single variable classifier, showing the value of combining descriptors which are not necessarily valuable predictors on

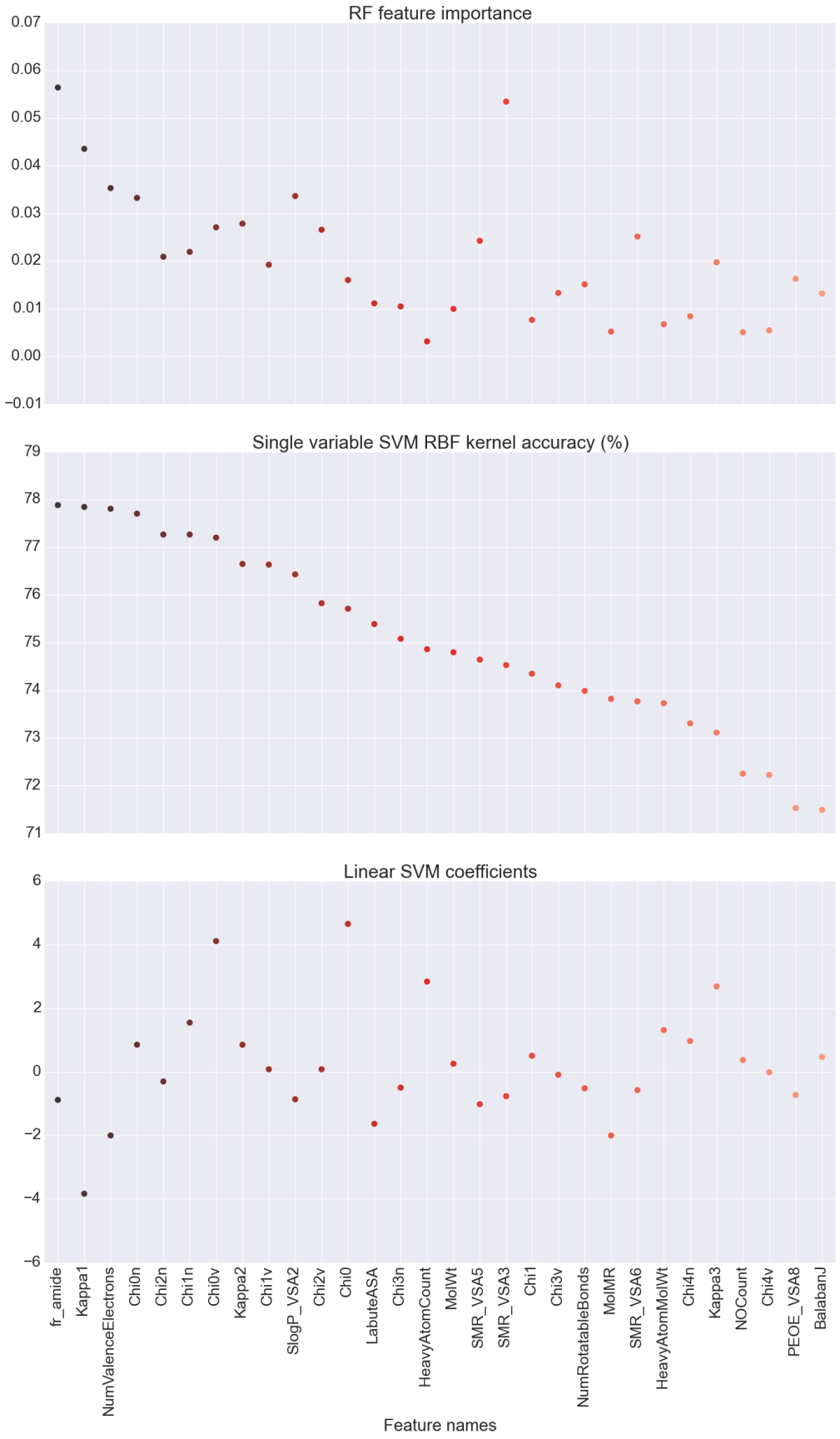


Figure 3.18 Feature importances for a) RF b) SVM with RBF kernel c) linear SVM, trained on the unfiltered set of molecules, ranked and colour-coded by single variable SVM RBF accuracy

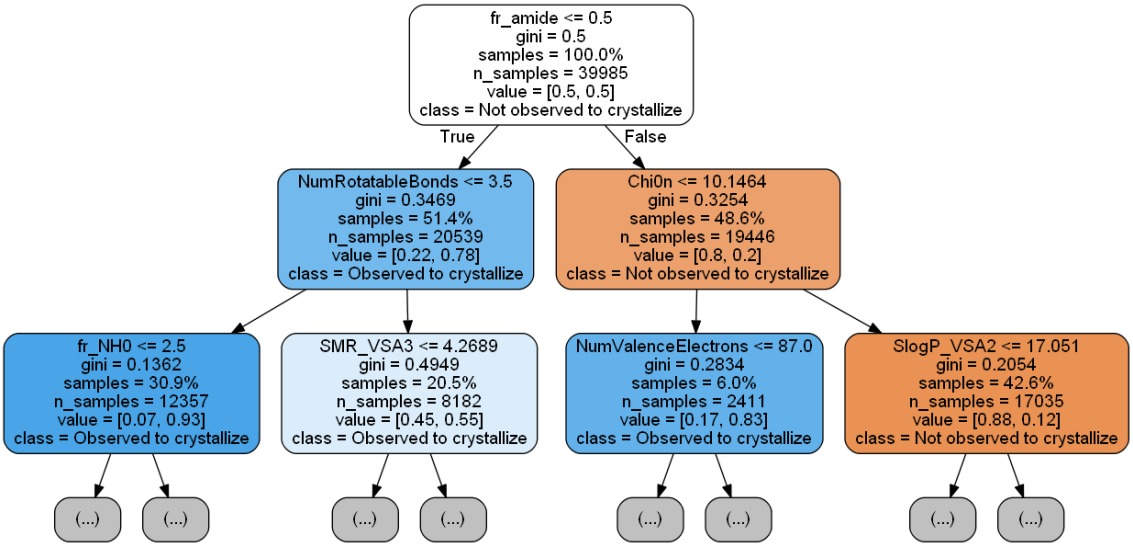


Figure 3.19 Decision tree used for rule extraction from the model trained on the unfiltered data with RDKit descriptors (top 3 levels shown). The gini coefficient is a measure of the impurity of the node. "Samples" indicates the percentage of the total dataset present at that node, and "value" is the proportion of "not observed to crystallize" (orange leaves) and "observed to crystallize" (blue leaves) molecules at the node. Each node has been assigned an overall class based on these proportions.

their own. Furthermore, the standard deviations of the cross-validation accuracies are relatively small, so the best performing two variable classifiers obtained by using SMR VSA3 are now significantly better than even other two variable classifiers.

By examining the histograms for the Lipinski rule-of-5 descriptors, as shown in Figure 3.21, and comparing them to those for the drug-like data used to create the initial model in Section 3.1 (Figure 3.1), it can be seen that the distributions of values for rotatable bond count, hydrogen donors/acceptors and TPSA are very similar to the original set. However, for both MolLogP and molecular weight the distribution of non-crystallizable molecules is much narrower and taller than for the crystallizable ones, which reflects the distribution of molecules in the full ZINC set due to the random sampling of the database. Since the molecules at both tails of the distribution are more likely to be crystallizable, a more complex decision surface is required to model this distribution, while such vastly different size distributions may artificially

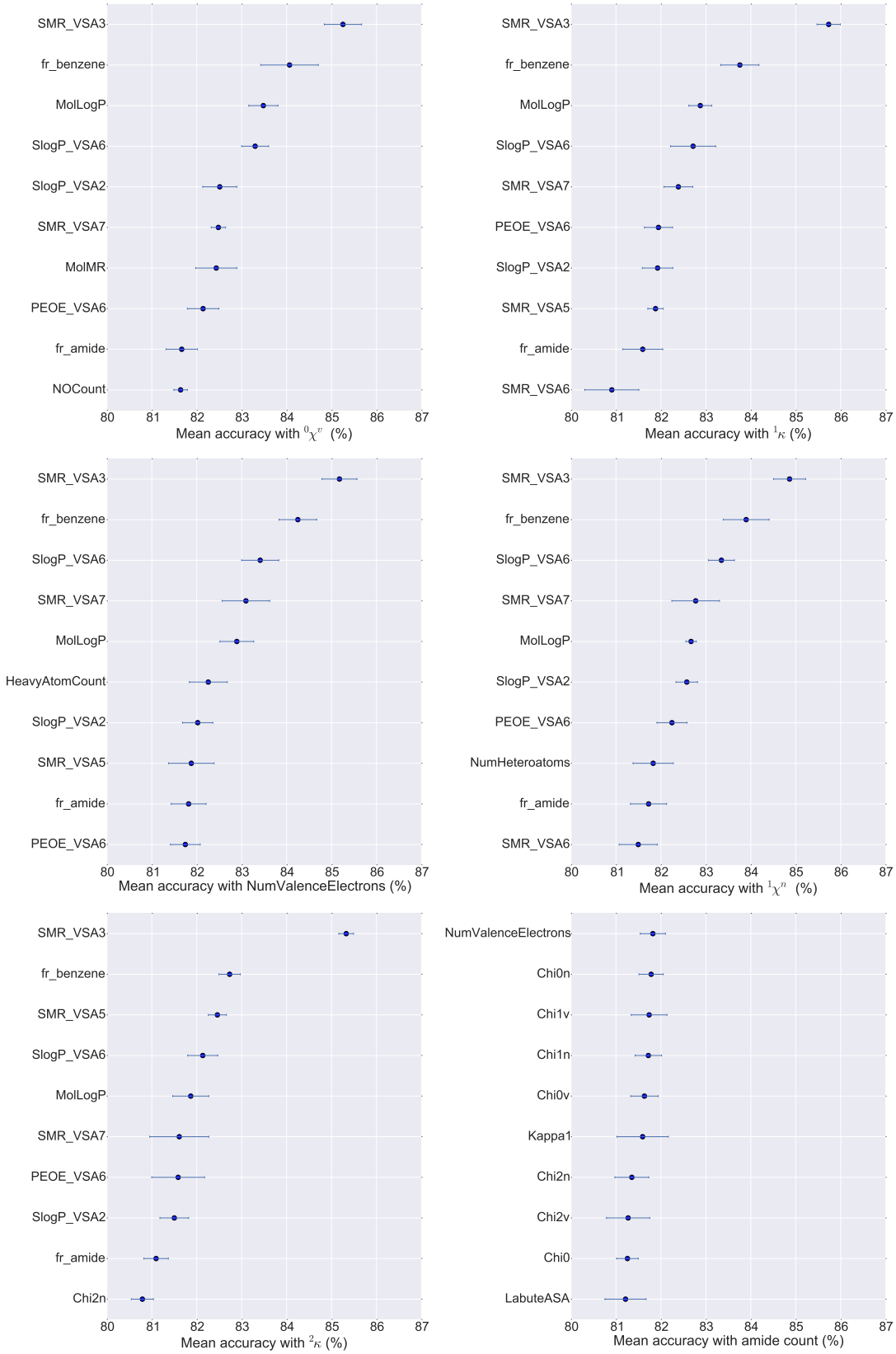


Figure 3.20 Mean predictive accuracy by cross-validation on two variable classifiers trained on the unfiltered data for each RDKit feature with six of the best-performing single RDKit features, with error bars showing the standard deviation of the cross-validation scores.

enhance the predictive accuracy if the model is able to correctly classify many of the molecules based solely on a simple split using the size.

If we examine the histogram of values for the best-performing descriptor, amide count, in Figure 3.22 it can clearly be seen that more crystallizable molecules have no amide groups present than non-crystallizable ones, information that was identified by the decision tree rule extraction process.

It would appear that increasing the amide count increases the chance of a molecule being non-crystallizable, and a high predictive accuracy could be achieved by predicting all molecules with no amides to be crystallizable. The test set distributions, shown in Table 3.10, indicate that there are three times as many crystallizable molecules containing no amide groups as those that do, while this distribution is reversed for the non-crystallizable molecules. To identify if the amide group descriptor was skewing the predictions towards one class or the other, the test set was split according to the amide group count.

Table 3.10 Breakdown of the unfiltered test set by amide count.

	Non-crystallizable	Crystallizable	Total
Amide	5050	1382	6432
No amide	1628	5270	6898
Total	6678	6652	13330

Table 3.11 shows that the model has a very low error rate of 2.4% for prediction of crystallizable molecules with no amide groups, and an even lower error rate of 1.9%

Table 3.11 Confusion matrices for test sets containing molecules with a) no amide groups b) at least one amide group from prediction by a SVM model with RBF kernel trained on the unfiltered dataset with RDKit descriptors.

Key		No amide group		Contains amide group(s)	
T (NC)	F (NC)	**89.5%**	10.5%	**98.1%**	1.9%
F (C)	T (C)	2.4%	**97.6%**	10.5%	**89.5%**
Overall		95.2%		96.0%	

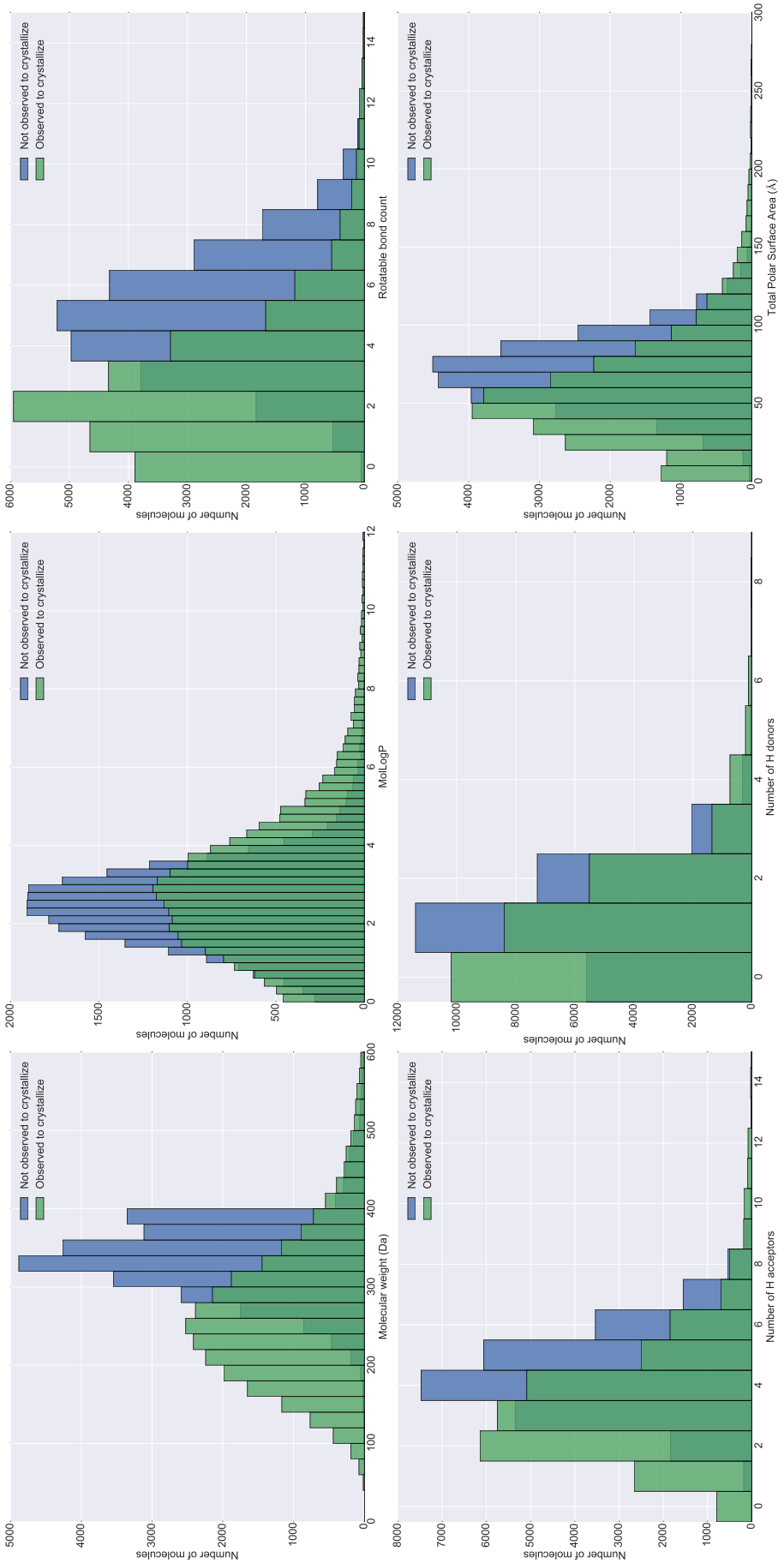


Figure 3.21 Histograms of key molecular descriptors for the unfiltered dataset.

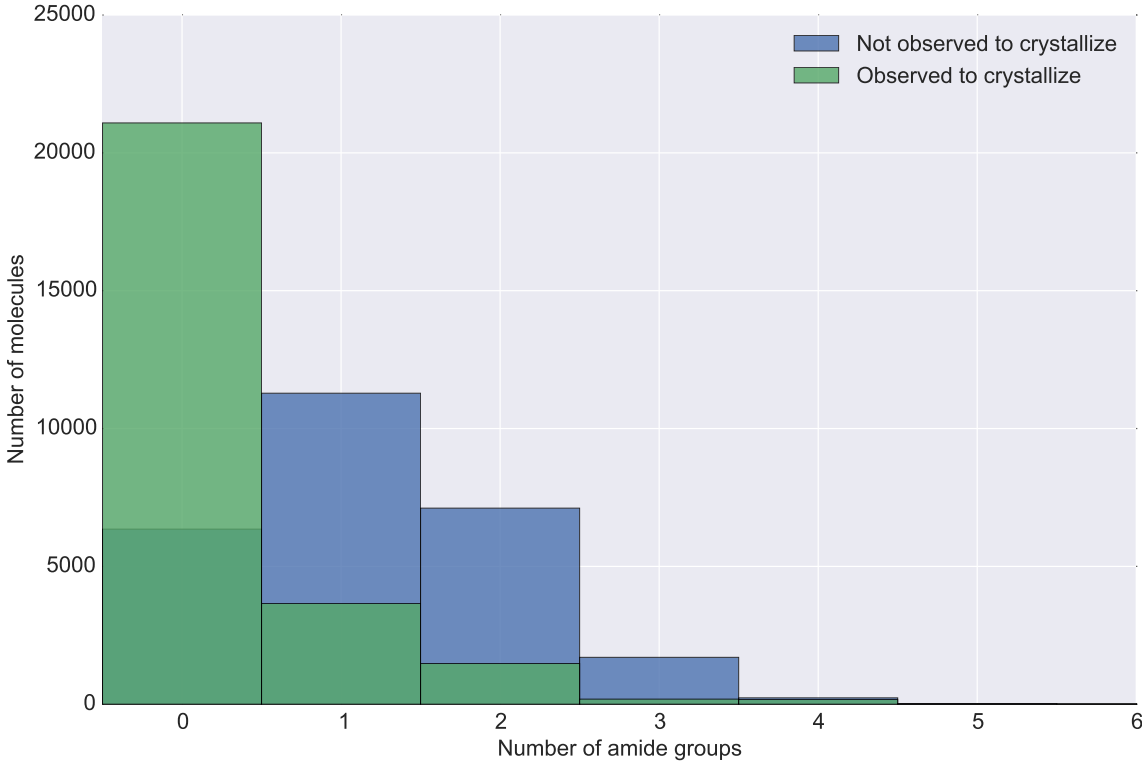


Figure 3.22 Histogram of amide group count for the unfiltered dataset.

for prediction of non-crystallizable molecules containing amide groups. This is to be expected, as the amide count is an important descriptor and has a strong influence on these predictions, and this is a reflection of the distribution of amide count in the two classes. Although the error rate increases five-fold for the crystallizable molecules containing amide groups and increases four-fold for the non-crystallizable molecules with no amide groups, the model still achieves around 90% accuracy on these subgroups of molecules. This suggests that while these molecules would be misclassified based on their amide count, the other descriptors contribute towards identifying the correct class in the majority of cases. Therefore the amide count is not having a confounding effect on the prediction, and is an effect that we can attempt to rationalise.

If we examine the type of amide group present across the training and test datasets, we find that of the 25878 molecules which contain at least one amide group, 77% of these contain only one type of amide group (primary, secondary or tertiary). Table 3.12 shows that, although the overall prevalence of primary amide groups is much

Table 3.12 Breakdown of the unfiltered test set by amide group type (for cases where a molecule only contains one amide group).

	Non-crystallizable	Crystallizable
Primary amide	214	288
Secondary amide	8904	3016
Tertiary amide	5886	1714

lower than for the other two types of amide, 57% of them crystallize, compared to 25% for secondary amides and 23% for tertiary amides.

By examining the types of interaction that are commonly found in the structures of such molecules that do crystallize, the thermodynamic reasons for the reduced crystallization tendency of secondary and tertiary amides over primary amides can be understood. Full interaction maps,[189] which use the information present in the CSD to visualise the interaction preferences of a particular conformation of a molecule, were generated for the three related compounds shown in Figure 3.23; one primary amide, one secondary amide, and one tertiary amide. These maps allow the assessment of multiple types of non-covalent interactions in the crystal structure.

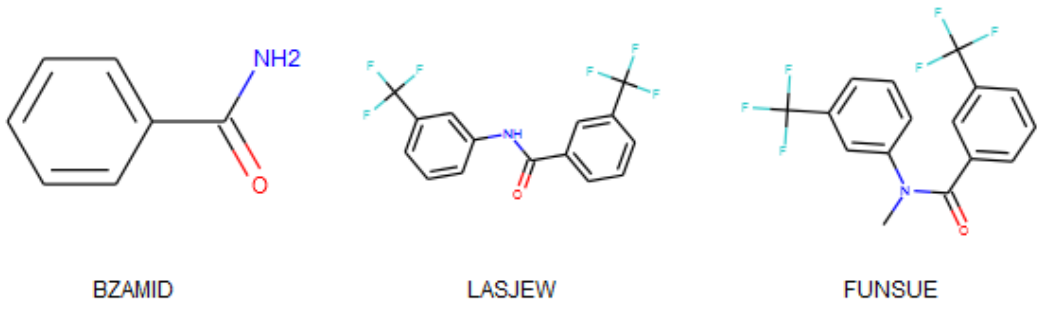


Figure 3.23 The amides used to generate full interaction maps, with their CSD refcodes.

For primary amides such as benzamide (Figure 3.24), the hydrogen bonding patterns have been extensively studied, and are dominated by strong N–H ···O hydrogen bonds.[190] The primary amide group contains a nitrogen atom with two potential hydrogen bond donors, which prefer to have interactions at the positions shown in

red in the interaction map, and two potential hydrogen bond acceptor positions for the lone pairs of the carbonyl group, which interact with neighbouring groups at the positions shown in blue on the map. This leads to two motifs in the same crystal, one of which is almost always a centrosymmetric dimer in which a N–H group forms a hydrogen bond to the carbonyl group on the neighbouring molecule and *vice versa*.[191] On each side of the dimer, this leaves one remaining N–H donor which forms a hydrogen bond to the remaining carbonyl acceptor position of a neighbouring dimer, and one carbonyl acceptor which bonds to the N–H donor of a neighbouring dimer. The same neighbouring dimer is often the source of both interactions resulting in a structure where the pairs are related by a translation of 5 Å,[192] which is the case here as shown in Figure 3.24, although other structures (such as a glide motif where the dimer is connected to two neighbouring dimers on each side) have been observed depending on the identity of the group connected to the amide.[193] Each interaction is formed in the centre of the favourable area identified by the full interaction map, showing that the network is close to ideal. The reliable two-dimensional nature of this hydrogen-bonding network, the four strong hydrogen bonds that each molecule makes, and the relative lack of steric hindrance which allows these interactions to form easily all contribute towards making crystallization of these molecules a favourable process.[192]

The interactions in secondary amides are similar in that they also involve strong N–H ···O hydrogen bonds, but the replacement of one of the hydrogen atoms on the nitrogen means that the nitrogen atom can now only hydrogen-bond to one other molecule. The interaction map shows that the carbonyl group now favours a single linear interaction,[194] which leads to the formation of one-dimensional chains of the trans conformation of the amide through the structure as shown in Figure 3.25. This is still a strong intermolecular interaction, and in cases where this motif can occur the molecule will still crystallize. However, the strength of the interaction along the chain favours crystal growth in this direction, leading to the growth of needles

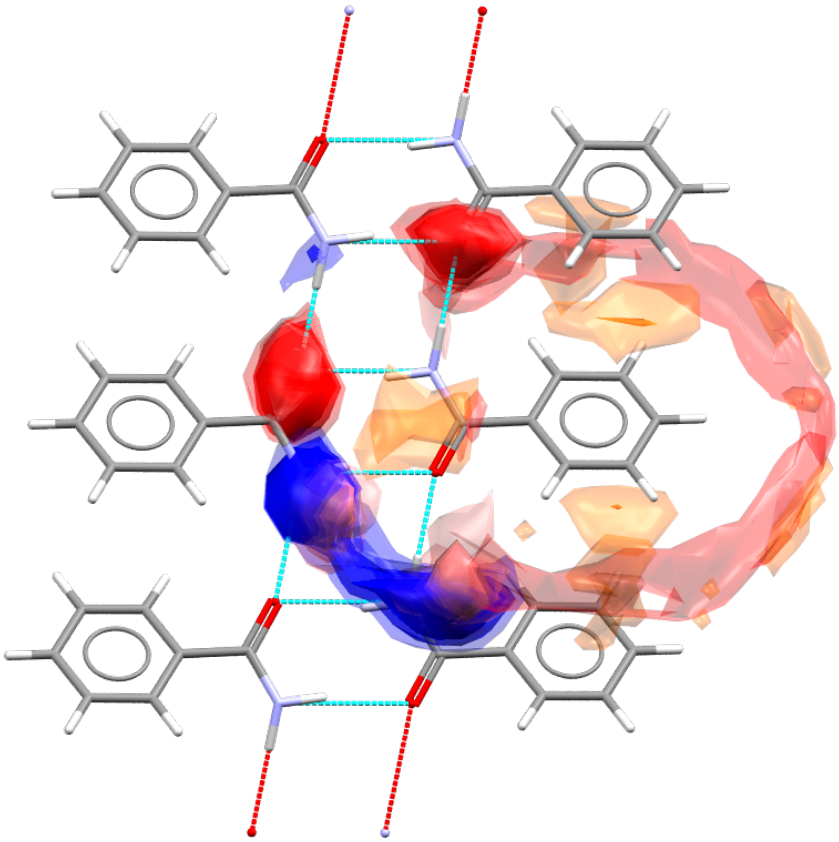


Figure 3.24 Full interaction map of a primary amide (benzamide, CSD refcode BZAMID) showing the interactions satisfied by the neighbouring molecules. Red areas show preferable locations for neighbouring H-bond donor groups, blue areas show preferable locations for neighbouring H-bond acceptor groups.

(particularly in solvents which do not interfere with the formation of the chain such as benzene),[195] which are often not suitable for SXRD. In addition, the more sterically hindered location of the secondary amide due to the two groups either side of it can often mean that this motif cannot be accessed, reducing the crystallization tendency. Even for this relatively sterically unhindered amide group, the N-atom is slightly displaced from the region where the carbonyl group would prefer to form an interaction, and in this case the structure is stabilised by a weak C(sp^2)–H ···F–C(sp^3) hydrogen bond as well as π–π stacking.[196]

By replacing the remaining hydrogen atom on the nitrogen atom with a methyl group, the secondary amide is converted into a tertiary amide. This group can now only act as a hydrogen-bond acceptor, so strong N–H ···O hydrogen bonds can no

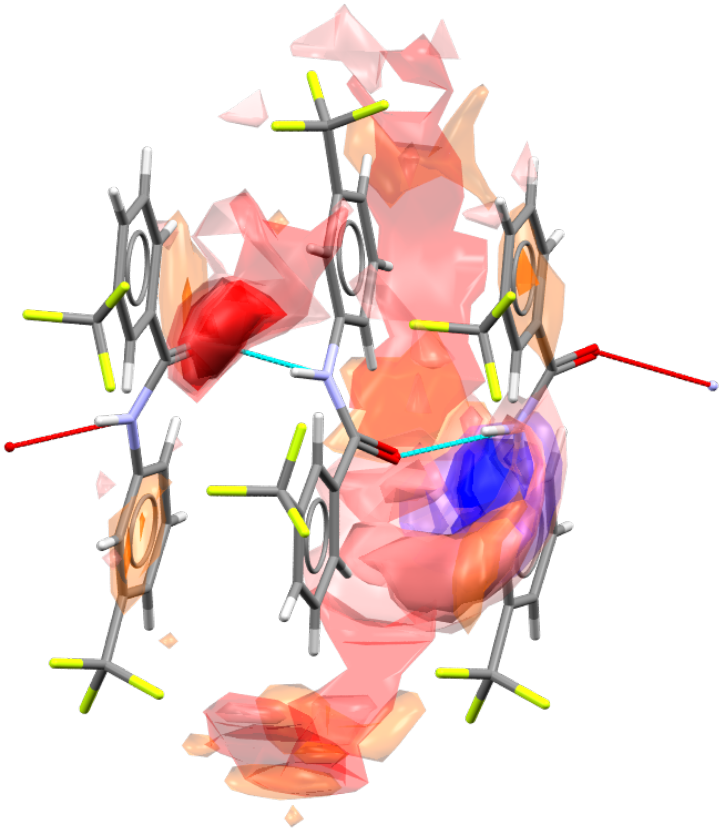


Figure 3.25 Full interaction map of a secondary amide (3-(Trifluoromethyl)-N-(3-(trifluromethyl)phenyl)benzamide, CSD refcode LASJEW) showing the interactions satisfied by the neighbouring molecules. Red areas show preferable locations for neighbouring H-bond donor groups, blue areas show preferable locations for neighbouring H-bond acceptor groups.

longer form and self-complementary bonding networks are not possible in the absence of other potential interacting groups.[192] Figure 3.26 shows that the carbonyl group exhibits a strong preference for an almost linear acceptor interaction with a donor while there are no strong donor regions of the molecule. In fact, the strongest interaction in this structure is a weak C(sp^3)–H ···O=C hydrogen bond which lies outside the zone of most favourable interaction of the carbonyl group, with other weak stabilisation being provided by C(sp^2)–H ···π hydrogen bond, π-π interactions, and C(sp^2)–H ···F–C(sp^3) halogen bonds.[197] The fact that several other similar molecules in this series which have been successfully synthesised remain as oils even at low temperature, and the necessity to use low temperatures for crystallization of those

materials which did form single crystals, highlights the difficulty of crystallizing materials with this functional group.

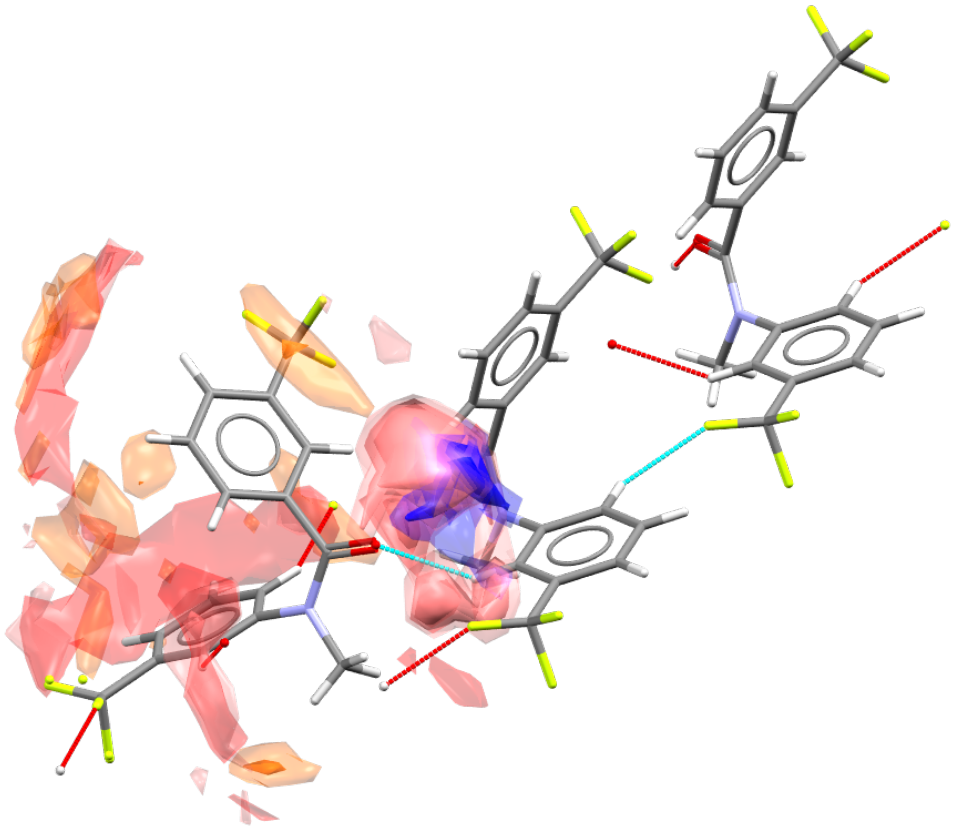


Figure 3.26 Full interaction map of a tertiary amide (N-methyl-3-(trifluoromethyl)-N-(3-(trifluromethyl)phenyl)benzamide, CSD refcode FUNSUE) showing the interactions satisfied by the neighbouring molecules. Red areas show preferable locations for neighbouring H-bond donor groups, blue areas show preferable locations for neighbouring H-bond acceptor groups.

Combining amide count with a descriptor related to molecular size, such as ${}^{1}\kappa$, does not provide the model with much extra information, as displayed in Figure 3.27. Again it can be observed that crystallizable molecules are generally smaller than non-crystallizable molecules, and that molecules containing no amide groups tend to be crystallizable. Although amide count performs well on its own, there are still a significant number of non-crystallizable molecules with no amide groups present, and combining this descriptor with a descriptor of molecular size such as ${}^{1}\kappa$ does not allow these molecules to be successfully classified, giving only 74% accuracy for the

non-crystallizable class, which explains the relatively poor two variable classifier accuracy with amide count.

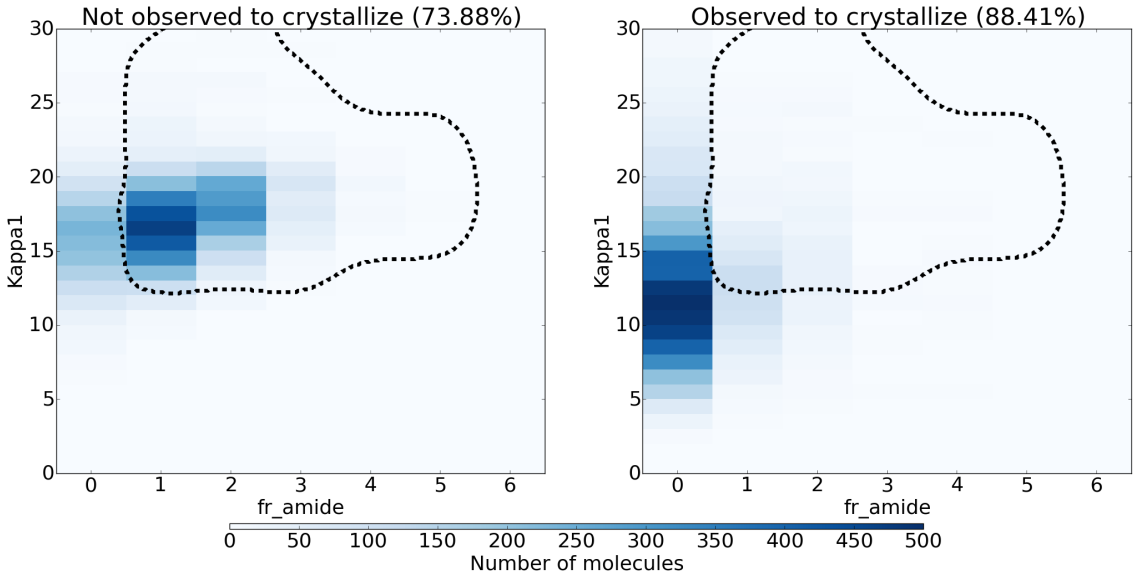


Figure 3.27 Heatmap of $^1\kappa$ and amide count distributions for the unfiltered test set.

The heatmap of the distributions of the descriptors which give the best performing two variable classifier ($^1\kappa$ and SMR VSA3) in Figure 3.28 shows that in general crystallizable molecules again tend to have a lower value of both descriptors. However, the influence of SMR VSA3 is less easy to determine, with the highest correlation coefficients being with the number of aromatic nitrogens (0.79) and the number of tertiary nitrogens (0.75), so it is necessary to delve deeper into the molecular features that contribute to the descriptor.

As described in Section 1.2.3, SMR VSA3 is a subdivided surface area descriptor which encodes information about the van der Waals surface area of the molecule with a molar refractivity of between 1.82 and 2.24. It is calculated by identifying all of the atoms with contributions to the molar refractivity in this range and summing their contributions to the van der Waals surface area. The only types of atom with molar refractivity values within this range are nitrogen atoms in either secondary or tertiary amines (which are not connected to a phenyl ring), or unprotonated aromatic environments. The definition encompasses not only amines but also amides, as illustrated in Figure 3.29, indicating that the explanation for the importance of this de-

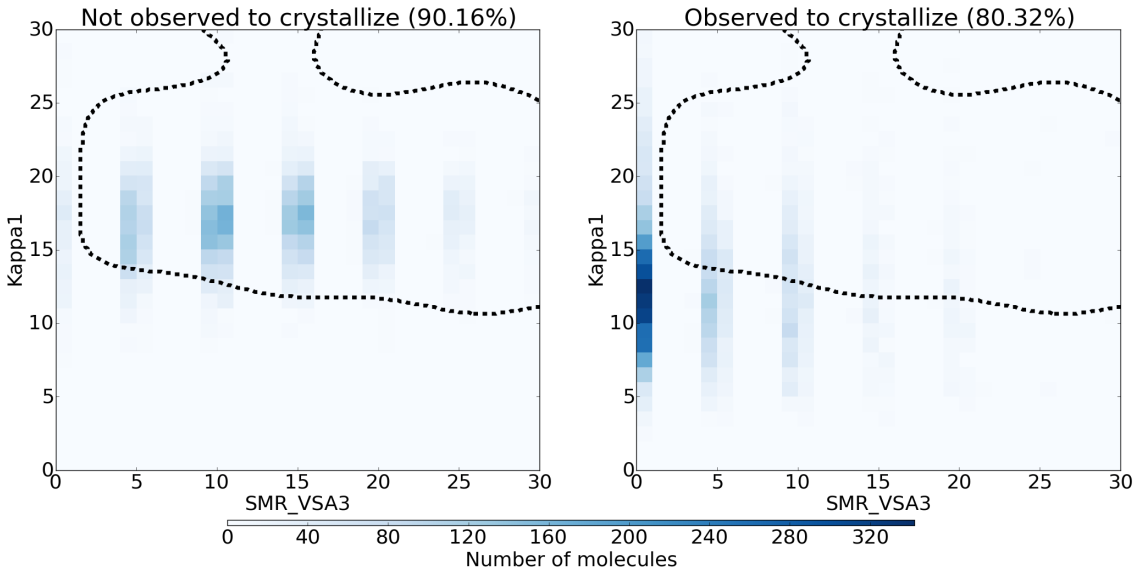


Figure 3.28 Heatmap of $^{1}\kappa$ and SMR VSA3 distributions for the unfiltered test set.

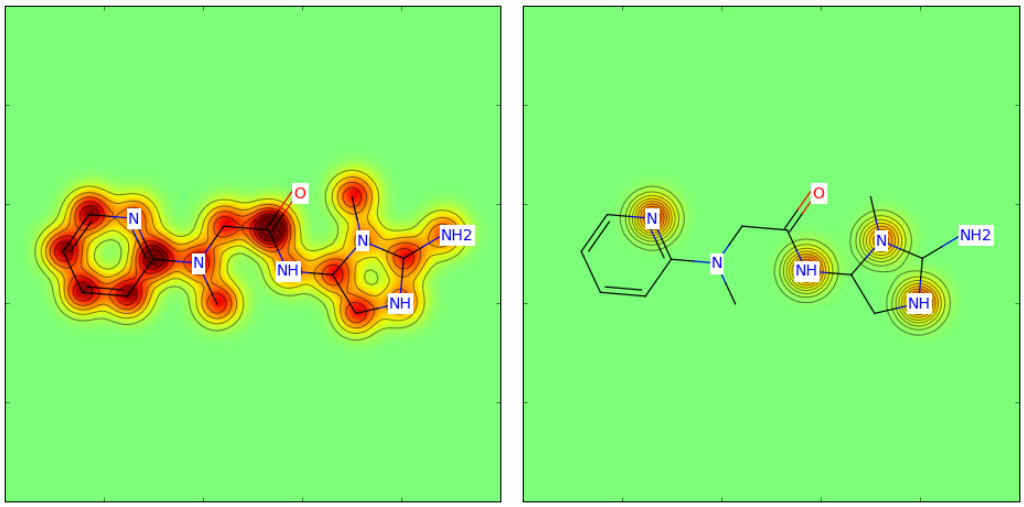


Figure 3.29 Atomic contributions to a) overall molar refractivity b) SMR VSA3.

scriptor will be tied to that of the amide count importance. Of the individual atomic contributions to the overall molar refractivity, all of the nitrogen atoms present in that particular molecule contribute to SMR VSA3 apart from the tertiary amide adjacent to the aromatic ring and the primary amide.

Each of these types of nitrogen atom can act as a hydrogen bond acceptor, but cannot act as a hydrogen bond donor, as is the case for tertiary amines, tertiary amides and unprotonated aromatic nitrogen atoms, or can only act as hydrogen bond donors

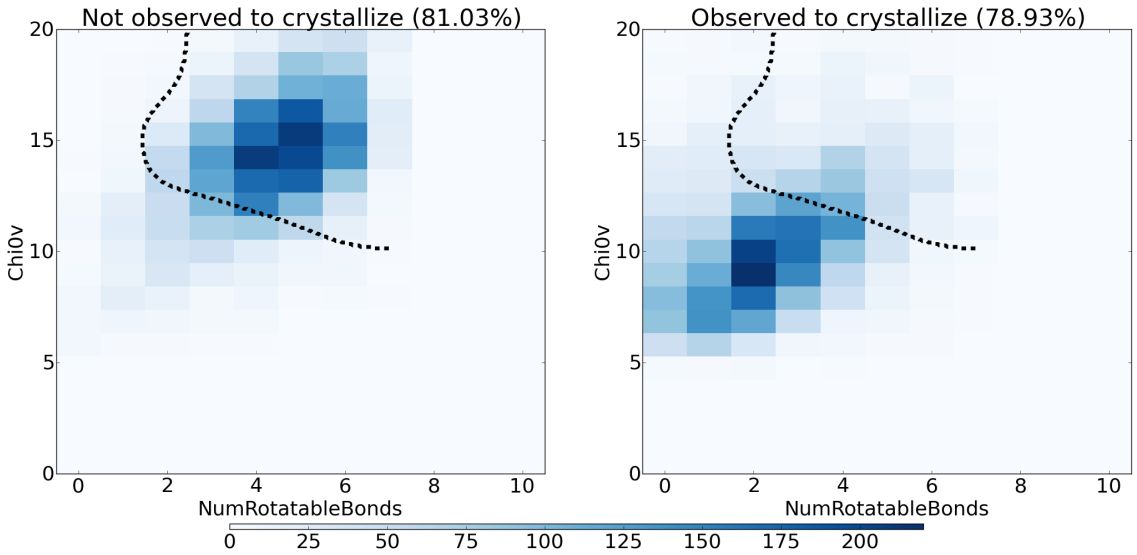
in structures where the amine or amide N–H group is relatively sterically unhindered, as is the case for secondary amides and secondary amines. Since many of the crystallizable molecules have a SMR VSA3 of zero, while almost no non-crystallizable molecules have a value of zero for this descriptor, it appears that molecules containing nitrogen atoms in these environments are less likely to crystallize than molecules that do not contain these atoms at all, due to the difficulty in forming self-complementary hydrogen bond networks with only acceptors and no donors. The accuracy on the non-crystallizable set is much improved relative to using amide count as a descriptor, showing that a molecule containing any type of nitrogen atom which cannot act as a hydrogen bond donor causes a molecule to be non-crystallizable, not just amide groups.

However, there are a significant number of molecules with a SMR VSA3 greater than zero that do crystallize. Crucially the heatmap shows that these molecules tend to be smaller than those that do not crystallize, information that is provided by combining SMR VSA3 with ${}^1\kappa$. There are several possible reasons for this; smaller molecules have less steric hindrance around the amide or amine group which reduces the disruption of the hydrogen bonding network, and they can more easily find the right orientation and conformation to allow the hydrogen bond to form. This information is not obtained by using SMR VSA3 on its own, which explains the poor performance of a single variable classifier using this descriptor, and the subsequent improvement using a two-variable classifier with this descriptor.

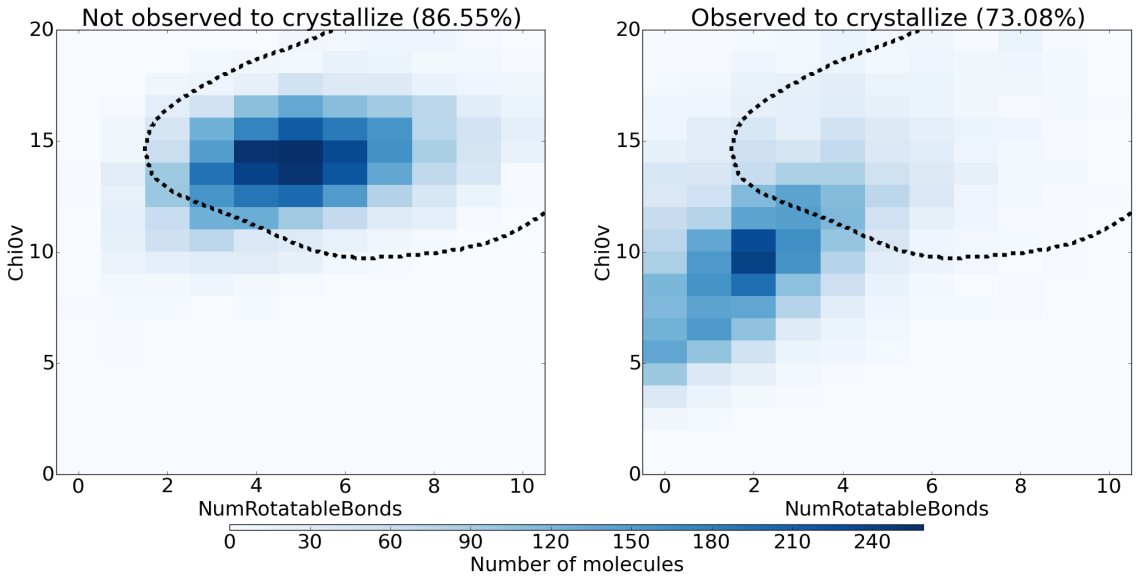
It should be noted that molecules containing tertiary nitrogen atoms are relatively common in the ZINC database, enabling descriptors (such as SMR VSA3) which encode this feature to become dominant, with a measurable effect on the predictions. Other motifs may have similar effects, but as they are less well represented in ZINC, they are determined to be less important by the model for performing the classification.

The continued influence of size (as shown with ${}^1\kappa$ for single and two variable clas-

sifiers) and flexibility (as demonstrated by the decision tree rule extraction) indicate that these factors are still important for determining the crystallization propensity with expansion of the set to all molecules, as was also discovered by the analysis of the initial model. This implies that the important descriptors from the original model remain significant on removal of the filter, and the distributions of rotatable bond count and ${}^{0}\chi^{v}$ for the unfiltered test set are indeed similar to those for the drug-like molecules in the initial test set (Figure 3.12). The comparison in Figure 3.30 shows that the decision boundary is also similar, but extends into new areas of chemical space for this new two-descriptor model, illustrating the increase in the domain of applicability of the model into these areas. However, the removal of the drug-like filter causes an increase in the spread of the crystallizable molecules, which results in a greater number of crystallizable molecules falling on the non-crystallizable side of the decision boundary, whereas the extra non-crystallizable molecules added on removal of the filter remain on the non-crystallizable side. Consequently, the predictive accuracy on the non-crystallizable set increases by nearly 6 percentage points while that of the crystallizable set decreases by 6 percentage points, leading to a very small overall increase in predictive accuracy.



(a) Original drug-like test set



(b) Unfiltered test set

Figure 3.30 Distribution of rotatable bond count against ${}^{0}\chi^{v}$ for test molecules colour-coded by density of molecules for a) the drug-like molecules for the original model b) the updated dataset with no drug-like filter. The dashed line shows the boundary between the crystallizable and non-crystallizable regions as predicted by RBF SVM.

Expanding the drug-like set to include all molecules without filtering extends the domain of applicability of the resulting model while improving both the bias and the variance of the models. The importance of size in making the classification has decreased but is still significant, while flexibility is still a key feature, and descrip-

tors encoding information about the type of nitrogen atoms present in a molecule have been discovered to have an increased influence in performing the classification. The fact that the molecular weight distributions are so different for the two sets of molecules could be skewing the results by making it easier to achieve a high classification accuracy with an easy initial split, an effect that can be investigated by replacing the random sampling of ZINC with one where the sampling is balanced according to the physical properties of the molecules.

Balanced by molecular weight

In the field of virtual screening for ligand discovery, the performance of molecular docking programs is assessed quantitatively by calculating the enrichment of ligand hits in a docking list. A benchmarking set of known ligands and background decoys is required for this, such as the Directory of Useful Decoys (DUD) dataset,[198] which contains 3000 ligands for 40 targets and 36 decoys per ligand. Since molecular weight was found to artificially enhance enrichment for a random set of drug-like decoys,[199] such databases match certain physical properties across ligands and decoys to ensure that enrichment is not achieved by identifying a simple difference between the two sets. For DUD, these descriptors were molecular weight, number of rotatable bonds, hydrogen bond donors, hydrogen bond acceptors and LogP, while for the subsequent DUD-E improved database, net charge was also considered.[200]

A similar approach can be applied here to assess the effect of balancing the physical properties of the crystallizable and non-crystallizable datasets. While the ability of a model to distinguish between crystallizable and non-crystallizable molecules within a subset of the test set with certain physical properties was assessed in Section 3.1.5, it would be informative to balance both the training and test data to create a new model in which the effect of other descriptors on the prediction can be identified.

The most obvious descriptor to balance the data by was molecular weight, since several of the most important descriptors for the classification are correlated with

Table 3.13 Confusion matrices for models trained on the unfiltered dataset balanced by molecular weight with RDKit descriptors.

Key		SVM (linear)		SVM (RBF)		RF	
T (NC)	F (NC)	**84.8%**	15.2%	**88.7%**	11.3%	**88.0%**	12.0%
F (C)	T (C)	13.4%	**86.6%**	9.0%	**91.0%**	10.3%	**89.7%**
Overall		85.7%		89.9%		88.9%	
Cross-validation		85.7(5)%		89.1(3)%		88.4(4)%	
Time (s)		28.6		2443		26.2	

it, and the distributions of this descriptor for the two classes are clearly different, as displayed in Figure 3.21 (p 111). As all of the CSD molecules present in ZINC are included in either the training or test set, it is easier to replace the random sampling of the entirety of the remainder of ZINC with a random sampling within specific molecular weight ranges. The ZINC molecules were split into bins with a width of 20 Daltons, and a random selection of molecules was made from each bin to roughly match the number of CSD molecules within that range.

Figure 3.31 shows that the molecular weight balancing has successfully matched the distributions of this descriptor between the two classes. This has caused little difference other than a slight shift to lower values for the distributions of the other descriptors, showing that although they may be slightly correlated with the size of the molecule, their distributions are not greatly affected by this.

The effect on the model accuracies is shown in Table 3.13, with a decrease in predictive accuracy observed for all models. The error rate has doubled for all algorithms, which is a consequence of the increased similarity of the descriptor distributions of the two classes, which makes distinguishing between the crystallizable and non-crystallizable molecules more difficult.

The feature importances show that molecular weight and any descriptors which are highly correlated with molecular size are no longer useful for the classification on their own, as might be expected when the weight distributions of the two classes have been balanced. Figure 3.32 shows that the classification is now performed using

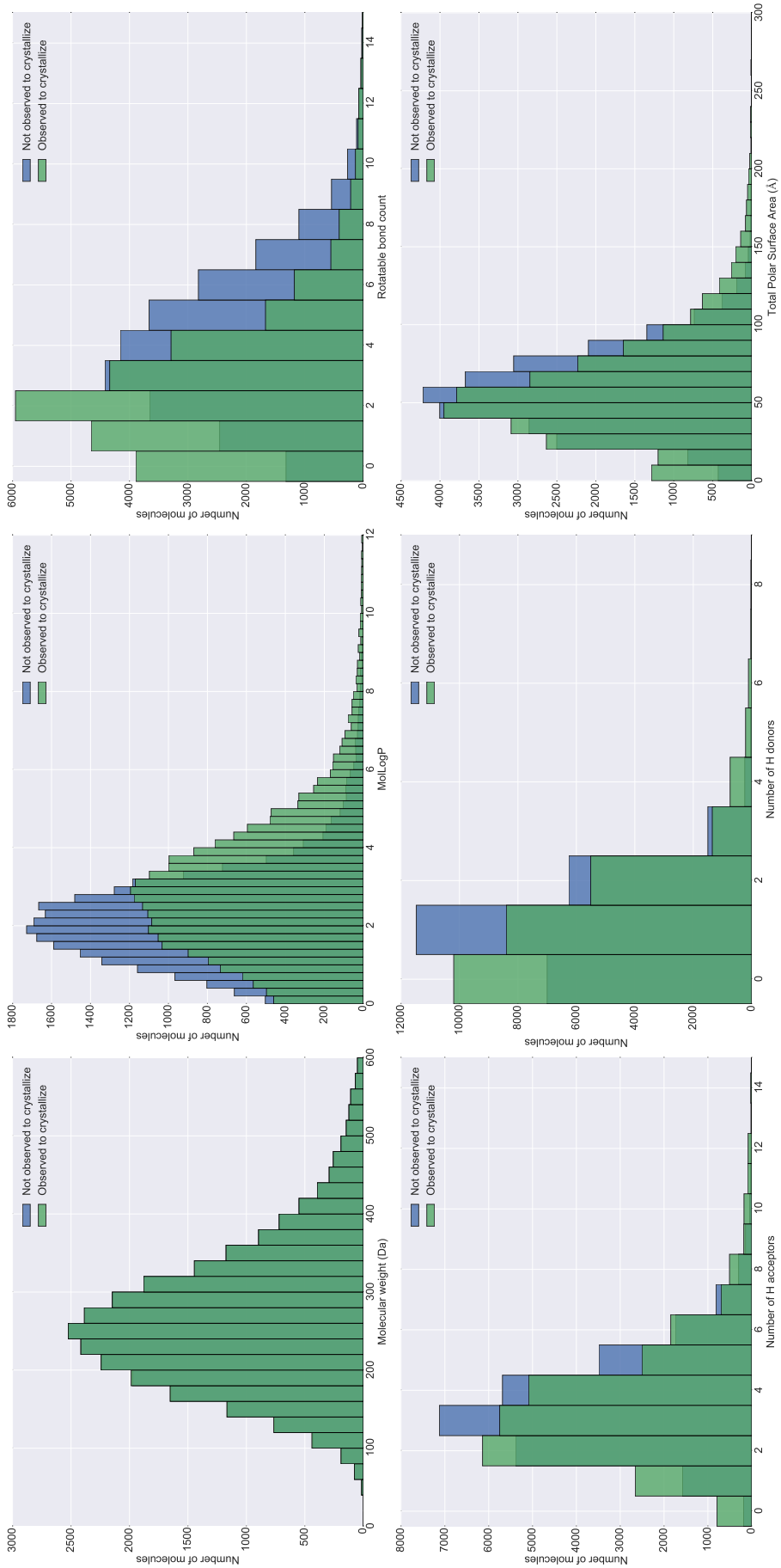


Figure 3.31 Histograms of key molecular descriptors for the dataset with balanced molecular weights.

descriptors related to nitrogen atoms and amide counts. The best performing single variable classifier, with SMR VSA3, shows a decrease in predictive accuracy of around 8 percentage points relative to the unbalanced model, a greater drop than that of the overall accuracy of the model.

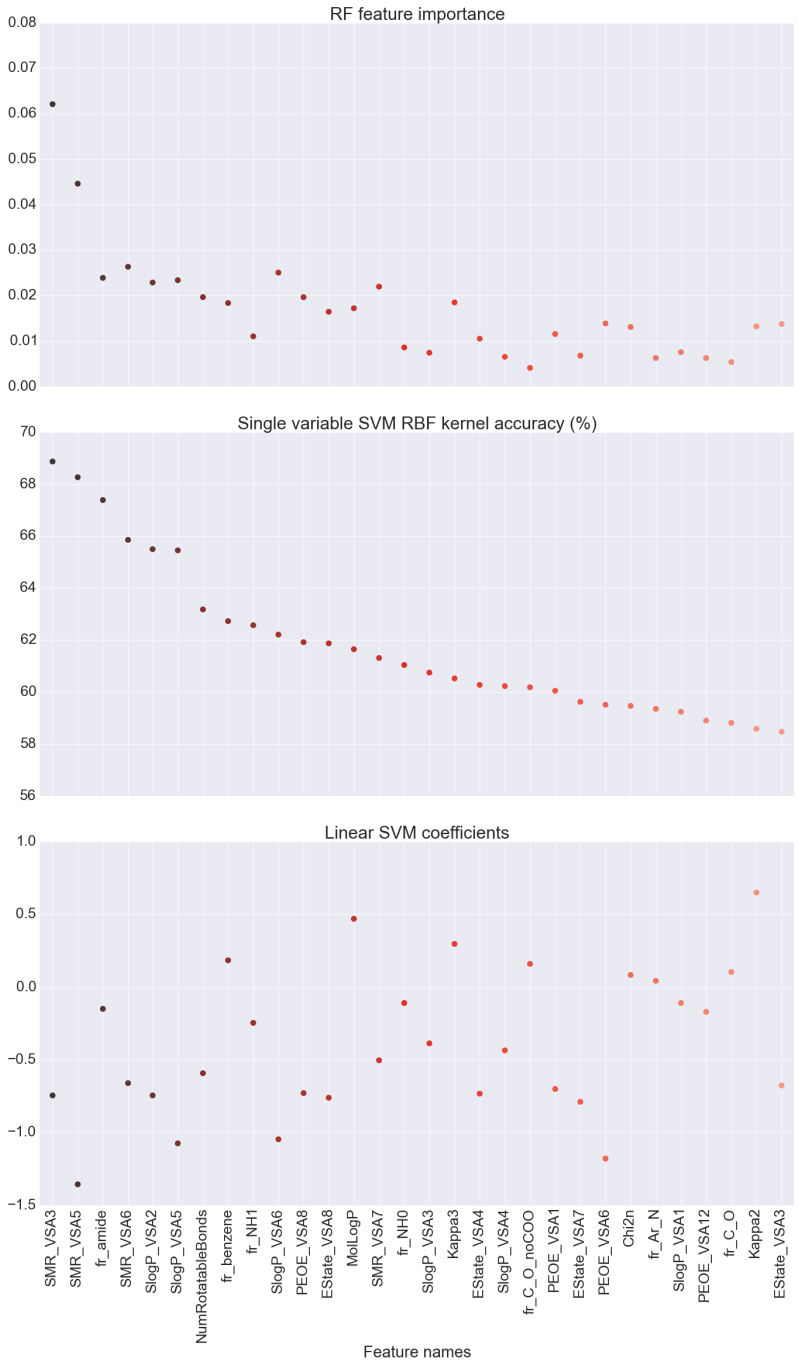


Figure 3.32 Feature importances for a) RF b) SVM with RBF kernel c) linear SVM, trained on the unfiltered set of molecules balanced by molecular weight, ranked and colour-coded by single variable SVM RBF accuracy

The balancing of molecular weight has reduced the predictive power of the other descriptors, since they are now generally more similar, especially as reducing the molecular weight reduces the number of potential functional groups a molecule can have, which in turn reduces the potential differences between the two classes. However, the continued importance of descriptors related to nitrogen atom types, such as SMR VSA3 and amide count, suggests that this is not an effect caused by the differing size of the molecules in the unbalanced dataset, and that the differences in number and type of nitrogen atom are a genuine difference between the two classes.

The model with unbalanced descriptors provides information about the likelihood of a ZINC molecule appearing in the CSD. Balancing the training set so that the two classes contain "similar" molecules provides a model which gives the likelihood of a given molecule crystallizing. This study highlights that the factors governing CSD membership and those governing crystallization likelihood are similar.

3.2.3 Effect of descriptors

Thus far, all of the models have been trained using the standard descriptors calculated by the RDKit cheminformatics toolkit. This contains a standard set of 177 descriptors as detailed in the appendix. However, this set of descriptors does not include two important sets of descriptors; the simple MQN descriptors described in Section 1.2.4, and the flexibility indices described in Section 1.2.6.

MQN descriptors

The MQN descriptors, as described in Section 1.2.4, are a simple set of integer counts of atoms, bonds, polarity features and topological features of the molecule. The 42 MQNs give a chemical space with fewer dimensions, where the features are much simpler to calculate. Although this can potentially lead to classification problems for MQN-isomers, which have the same value for all 42 of the numbers, Table 3.14 shows that in fact the performance of each model only decreases by around a percentage

Table 3.14 Confusion matrices for models trained on the unfiltered dataset with only MQN descriptors.

Key		SVM (linear)		SVM (RBF)		RF	
T (NC)	F (NC)	**87.7%**	12.3%	**94.0%**	6.0%	**93.7%**	6.3%
F (C)	T (C)	12.0%	**88.0%**	8.0%	**92.0%**	6.9%	**93.1%**
Overall		87.8%		93.0%		93.4%	
Cross-validation		88.1(3)%		93.2(3)%		93.7(2)%	
Time (s)		9.4		314		6.7	

point for the RF algorithm and 2.5 percentage points for the RBF SVM (although this still represents over a 50% increase in the error rate for the case of the RBF SVM), while there is an appreciable decrease in the length of training time due to the smaller dimensions of the feature space. Linear SVM suffers the most, with the error rate almost doubling from 6.5% to 12.2%. The RF algorithm now provides the best model.

The feature importances shown in Figure 3.33 show that there is a much more obvious correlation between the important features for the RF algorithm and the best performing features in single-variable classifiers, while the linear SVM importances are fairly evenly spread, with no stand-out features, perhaps a symptom of the relatively poor fit to the data. The two most important variables are the number of heavy atoms and the number of acyclic single bonds, which are both indicative of the size of the molecule, while the third most important descriptor is again the number of rotatable bonds, providing further evidence that size and flexibility are important factors for distinguishing between “crystallizable” and “non-crystallizable” molecules.

However, the best performing MQN single-variable classifier, which uses the number of heavy atoms, still gives a mean accuracy of only 74.9%, which is significantly worse than the best performing RDKit descriptor, the amide count, which has an accuracy of 77.9%, and is only placed 15th in the list of RDKit feature importances. This indicates that other descriptors of molecular size are more informative in performing the classification, possibly because they correlate better with molecular size, since heavy atom count takes no account of the atom identity.

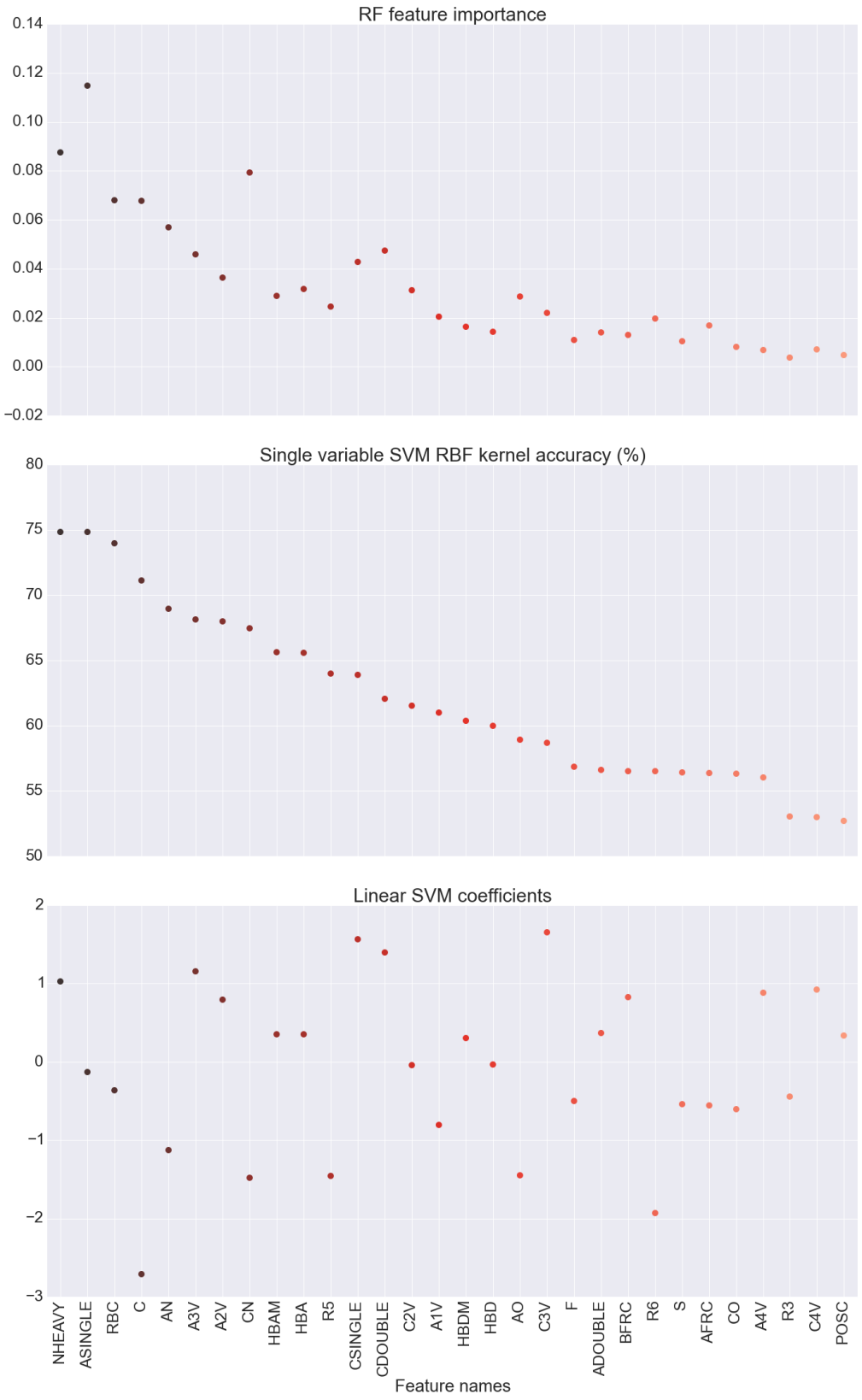


Figure 3.33 Feature importances for a) RF b) SVM with RBF kernel c) linear SVM, trained on the unfiltered set of molecules using MQN descriptors, ranked and colour-coded by single variable SVM RBF accuracy.

This is supported by the decision tree in Figure 3.34, which shows that the best possible split is obtained using the heavy atom count, but this only accounts for around 75% of the accuracy of the RBF SVM, which is worse than the best possible split for the RDKit descriptors. On the non-crystallizable side of the tree, the number of cyclic nitrogen atoms appears at the third level of the tree. This will be correlated with the nitrogen atom types encoded by SMR VSA3 (number of secondary and tertiary amines and amides, and unprotonated aromatic nitrogen atoms), but does not account for those amines and amides which are secondary or tertiary and also acyclic. Since the MQN numbers only count atom types rather than functional groups, the information is less specific than that provided by the initial descriptors, which could be an indicator for why the performance is worse. Since the overall accuracy is not much worse than for the RDKit descriptors, the MQN descriptors must combine in a way which captures most of the same information that the RDKit descriptors do.

The models built using MQN descriptors confirm that the size and flexibility of the molecule are important for distinguishing between crystallizable and non-crystallizable molecules, and also use information about nitrogen atom types to make the classification, but the MQNs capture this information less well than the RDKit descriptors, leading to slightly worse overall performance but much worse single-variable importance.

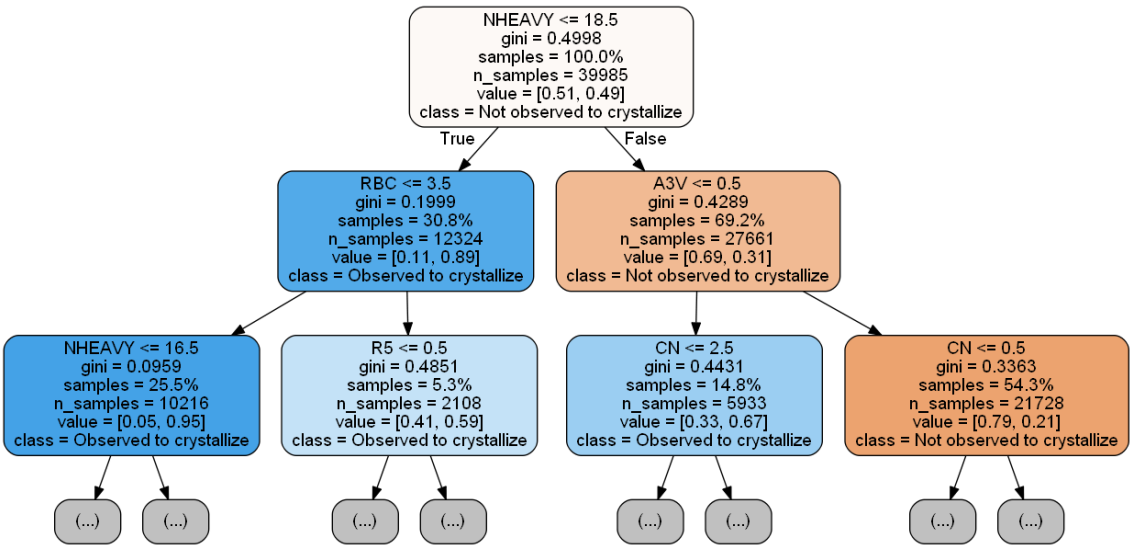


Figure 3.34 Decision tree used for rule extraction from the model trained on the unfiltered data with MQN descriptors (top 3 levels shown). The gini coefficient is a measure of the impurity of the node. "Samples" indicates the percentage of the total dataset present at that node, and "value" is the proportion of "not observed to crystallize" (orange leaves) and "observed to crystallize" (blue leaves) molecules at the node. Each node has been assigned an overall class based on these proportions.

Flexibility descriptors

Following the confirmation that the flexibility of a molecule plays a key role in determining the likelihood of crystallizing that material, steps were made to incorporate descriptors into the model which captured this information more directly than the rotatable bond count. The limitations of RBC in this regard are clear. The spread of RBC values is relatively small, and as they are integer values rather than a continuous variable, many molecules have the same rotatable bond count. In addition, some bonds which are classed as rotatable actually cause no change to the molecule due to symmetry, information which is not captured by this descriptor.

Other flexibility indices have been developed which are not contained within the standard list of RDKit descriptors. These include the Kier flexibility index and the path length flexibility index,[106] which are described in Section 1.2.6 and implementations of which were self-coded to compare with existing descriptors.

Table 3.15 Confusion matrices for single variable SVM models trained on the unfiltered dataset using a) rotatable bond count b) Kier flexibility index c) Path length flexibility index.

Key		Rotatable bond count		Kier flexibility		Path length flexibility	
T (NC)	F (NC)	**76.0%**	24.0%	**84.9%**	15.1%	**83.5%**	16.5%
F (C)	T (C)	28.1%	**71.9%**	33.2%	**66.8%**	34.5%	**65.5%**
Overall		74.0%		75.9%		74.5%	
Cross-validation		74.0(2)%		75.8(2)%		74.9(5)%	

However, as can be seen from Table 3.15, these other flexibility descriptors, despite performing slightly better than rotatable bond count, do not perform as well as other descriptors.

This is most likely because they are still derived from values which are based on the two-dimensional connectivity of the molecule. Since flexibility in the case of crystallization is referring to the number of three-dimensional conformations which exist in solution, it is unlikely that descriptors derived from a 2D representation will capture this very well. A better approach would be to actually sample the potential conformations that a molecule could adopt and create a descriptor from this, thereby introducing some three-dimensional information into the model, as described in Chapter 4.

3.2.4 Conclusions

The improved curation of the dataset results in an approximate halving of the error rate, due both to the removal of duplicates and the increased size of the training set, which reduces the bias and the variance of the models. Removal of the drug-like filter extends the domain of applicability of the model to include non-drug-like molecules, without loss of predictive capability on the drug-like set. Size and number of rotatable bonds continue to be important features for predicting crystallization, as are descriptors encoding information about nitrogen atom environments in the

molecule. Balancing the dataset by molecular weight causes a slight loss of predictive accuracy, but shows the difference between the classes is not simply due to molecular size imbalance.

The MQN descriptor set captures much of the same information as the original RDKit descriptors in a simpler fashion using integer counts of molecular features, but with a higher error rate. The importance of flexibility and size are confirmed, but two-dimensional flexibility descriptors have no greater predictive capability than rotatable bond count.

3.3 Co-crystal prediction

To test if a machine-learning approach could be applied to the prediction of co-crystal formation, a machine-learning model was trained on the in-house co-crystallization screen described in Section 2.1.2. This consisted of 657 experimentally determined data points, (401 unsuccessful, 254 successful). Of these 254 successful data points, 44 were already reported in the literature (39 co-crystals and 5 salts) and the remaining 210 represent novel solid forms. We were confident that this training data is a useful set for predicting co-crystal formation rather than salt formation due to the low number of salts found in the CSD, and the low success of salt formation from dry grinding, although the likelihood of co-crystal over salt formation can be assessed by comparing co-former and API pKa values.[201] The descriptors used to represent these data points were calculated as explained in Section 2.2.2.

In addition to the classification accuracy and ROC curve for the predictions on the external validation set of paracetamol co-crystals, the capability of the approach to "enrich" the number of successful hits in the ranked list was calculated, since this has practical application in reducing the number of experiments required to find co-crystal forms. For the external validation set, the co-crystals were ranked according to the probability estimate given by the predictive model. This was compared to the actual hits in order to quantify how many successful co-crystals were identified

Table 3.16 Confusion matrix for prediction on the paracetamol co-crystal validation set.

Key		Paracetamol co-crystals	
T (NC)	F (NC)	**12**	1
F (C)	T (C)	11	**10**
Overall		64.7%	
Cross-validation		75.0(1)%	

by this selection method. From this ranking, an enrichment factor (EF) provides a numerical score that quantifies the observed success rate at the top of the list relative to randomly sampling the list. For the top x% of the ranked list:

$$\mathrm{EF}_x = \frac{N_{\mathrm{hits}}}{N_x} \Big/ \frac{N_{\mathrm{total\ hits}}}{N_{\mathrm{total}}} \tag{3.1}$$

where N_{hits} is the number of hits in the top x%, N_x is the total number of successful and unsuccessful co-crystals in the top x%, $N_{\mathrm{total\ hits}}$ is the number of hits in the whole list and N_{total} is the total number of successful and unsuccessful co-formers in the whole list.

The predictive accuracy of the model in classifying the co-formers as forming successful or unsuccessful co-crystals with paracetamol was poor at 64.7%, much lower than the cross-validation accuracy on the training set (75.0(1)%). The confusion matrix (Table 3.16) shows that the model has a tendency towards predicting more of the combinations as being successful co-crystals, which reduces the number of false negatives (successful co-crystals that are incorrectly marked as unlikely to form and so would be missed). This may be a result of differences between paracetamol and the co-formers making up the training set, which could affect the reliability of the probability cut-off that the model uses to make its predictions.

However, Figure 3.35 shows that the list of co-formers ranked by the probabilities obtained from the model successfully identifies 9 of the 13 co-crystals of paracetamol within the top 11 suggestions in the list. The AUC 0.85 is significantly better than the

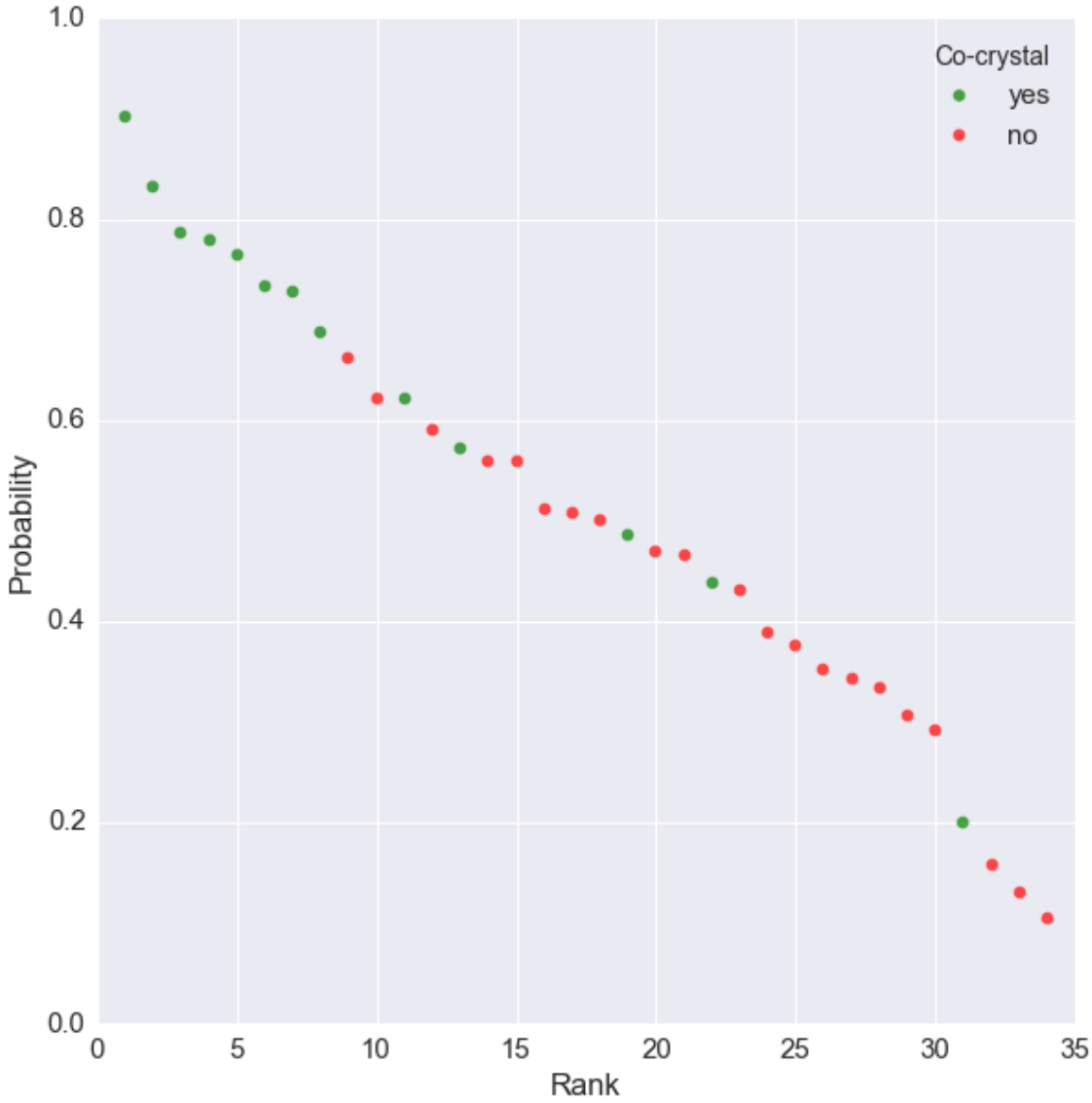


Figure 3.35 Probability ranking by the model for the paracetamol external validation set. Green dots indicate successful co-crystals, whereas unsuccessful co-formers are represented as red dots. More information is given in the Appendix.

AUC of 0.66 obtained from the HBP method, as shown in Figure 3.36.[43] The EF_{25} of 2.6 corresponds to 100% successful prediction in the top 8 (25%). This suggests that although the probability cut-off between successful and failed co-crystals used by the algorithm to perform the binary classification may be wrongly positioned, the probability ranking itself provides a way of identifying co-formers which are likely to form co-crystals, reducing the number of experiments required to successfully identify co-former pairings.

Figure 3.36 ROC curves for the paracetamol external validation set. The blue line is this work, the green line uses the predictions made in Wood *et al.*,[43] and the dashed line indicates a random classification.

The importance of ensuring that co-former molecules lie in a similar area of chemical space to the molecules used for training the model is illustrated by salsalate, for which the current model predicts the same probability of co-crystal formation regardless of the co-former paired with it. On examination of the scaled descriptors, 39 salsalate descriptors are found to have values greater than 3 standard deviations from the mean of the training set (compared to 5 descriptors for paracetamol). This is an indicator that the distance between salsalate and the training molecules in chemical space is too great for the model to provide a sensible prediction. Consequently, the descriptors of test molecules need to be examined carefully after scaling to ensure that the existing model is suitable for use with that particular molecule, and extension of the training set to sample a more appropriate area of chemical space should be considered if this is not the case.

The ability of the model to successfully rank co-formers other than those included in the training dataset indicates that the co-formers and descriptors used to train the model provide enough information to allow the model to be applied to a wide range of co-formers. A machine learning algorithm trained on an in-house set of co-crystallization experiments using simple descriptors as the input can be used to guide selection of co-formers for a particular API. This is likely to assist industry by saving both time and resources on experimental screens, particularly in the early stages of co-former selection. Increasing the scope of the model can be envisaged by retraining the algorithm using a larger range of APIs and co-formers. Such issues are not encountered by the HBP method as it relies on interactions rather than molecular descriptors, so generalises to diverse chemical species more easily.

3.4 Conclusions

This chapter described the development of predictive models for crystallization using machine learning algorithms. The method was found to provide an error rate of 10% for a drug-like set of molecules, with improved curation leading to a halving of the error rate. The size and flexibility of the molecule, in addition to nitrogen atom environments, were found to be important molecular descriptors for predicting crystallization, and the difference between the two classes is demonstrated to not be solely due to the size imbalance. The related problem of co-crystallization prediction can also be tackled using this approach, with a greater than 2-fold increase in enrichment on an external validation set.

Chapter 4

$nConf_{20}$ Descriptor Development

This chapter presents the development and testing of a descriptor to capture the 3-dimensional conformational flexibility of molecules. The new descriptor, $nConf_{20}$, performs significantly better than any existing 2-dimensional molecular descriptor, although incorporation of the descriptor into the overall model of crystallization propensity results in only a negligible increase in predictive accuracy. Much of the work in this chapter has been published in Wicker, J. G. P., & Cooper, R. I. (2016). Beyond Rotatable Bond Counts: Capturing 3D Conformational Flexibility in a Single Descriptor. *Journal of Chemical Information and Modeling*, **56**(12), 2347–2352.[202]

Contents

4.1 Introduction

Molecular flexibility was found to be a key feature of molecules for predicting crystallization propensity from the rule extraction analyses in Chapter 3. However, the only standard RDKit descriptor which encodes this information is the rotatable bond count, and other flexibility descriptors based on 2-dimensional molecular representations do not improve significantly on this, as discovered in Section 3.2.3. The impor-

tance of conformational flexibility in crystallisation has previously been documented,[26, 74] so a descriptor which captures this information more directly should improve the crystallization predictions.

4.2 Method

Molecules were provided to the conformer-generation step as SMILES strings to ensure no residual conformational information was retained, and explicit hydrogen atoms were added to the skeleton as they are required by the force field to ensure that sensible conformers were generated. RDKit cheminformatics toolkit[82] functions were used to generate 50 random molecular conformations, while retaining the starting stereochemistry. RDKit was chosen over other open-source conformer generation tools like BALLOON, CONFAB and FROG2, and commercial platforms such as MOE, due to speed and the ability to generate conformers which are structurally similar to experimentally determined structures.[203] A knowledge-based conformer generator which uses experimental observations of torsional angle distributions is available in the latest release of the CSD tools.[164] These alternatives have not been explored in this thesis, but could potentially be used to sample conformational space in a similar manner to RDKit.

Each randomly generated conformer was optimized using the Merck Molecular Force Field (MMFF94).[204] MMFF94 is a general purpose parameterized force field comprised of several well defined contributions to the total potential energy of a molecule, including bond stretching energy, bond torsion, electrostatic and van der Waals interaction energies. The force field parametrization is determined by training on a large set of computational data derived from *ab-initio* calculations on a diverse range of organic and bio-inorganic structures and has been implemented within the RDKit.[205] Some other force fields suitable for organic molecules include Amber, Gaff and CHARMM.[206] The Universal Force Field (UFF) can be used to compute energies and gradients of molecules containing almost any element and may therefore prove

useful if extending this work to metal-organic complexes or inorganic molecular materials. MMFF94 has been shown to reproduce gas-phase conformer energies more accurately than these other widely available force fields,[207] and was chosen for its significantly shorter computational time compared to a more accurate molecular dynamics calculation including solvent effects.

If the optimization did not converge to a stable minimum the conformer was removed. The force field is then used to calculate the energy of each conformer; its energy relative to the lowest-energy conformer found is stored. The lowest energy conformer is retained and for each other conformer the alignment of all permutations of matching atom orders with every other conformer is checked, to account for symmetry. Any duplicate conformers with a heavy atom root mean square (RMS) distance of less than 1.0 Å to any other conformer are removed.

For the small minority (0.05%) of molecules where the MMFF optimization failed, the molecule was removed from the study.

The entire calculation of the energies takes around 0.2 s for molecules with fewer than two rotatable bonds, 1–2 s for molecules with 4 or 5 rotatable bonds, and up to 5 s for molecules with 8 rotatable bonds.

4.3 Descriptor creation

A new single value descriptor was developed based on the distribution of relative conformer energies. The new descriptor is a count of additional conformers (not including the lowest energy conformer) with energies between selected relative energy thresholds, and is designed to approximate the number of energetically accessible conformations of a molecule.

In order to find the optimal energy thresholds for the descriptor, a five-fold cross-validation was carried out on the training set using the descriptor to create a single variable classifier. Figure 4.1 shows the distribution of accuracies, which has a broad maximum between an upper threshold of 16 kcal mol^{-1} to 20 kcal mol^{-1} and a lower

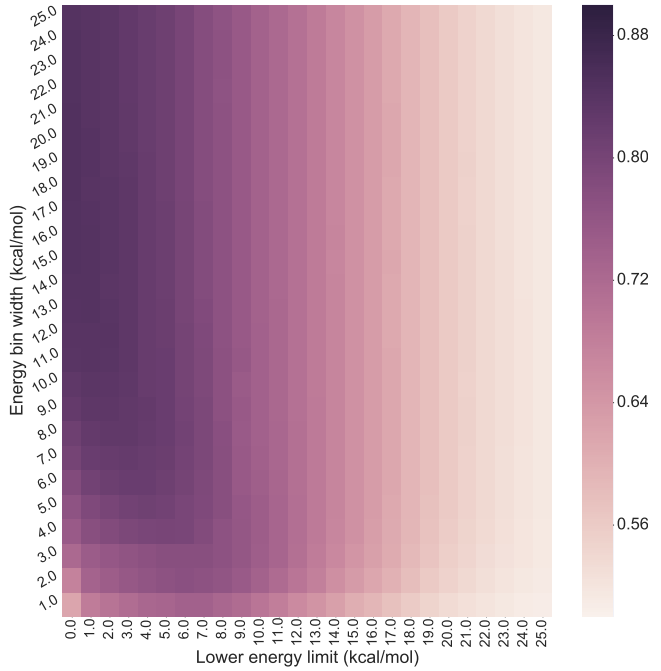


Figure 4.1 Predictive accuracies for the conformer energy descriptor with varying limits, as determined by 5-fold cross-validation.

energy threshold of 0 to 1 kcal/mol, with no significant difference between the predictive accuracies. This led to a choice of 0 as the lower threshold and 20 as the upper threshold. An example of calculating this descriptor using a 20 kcal mol^{-1} cutoff is given in Table 4.1.

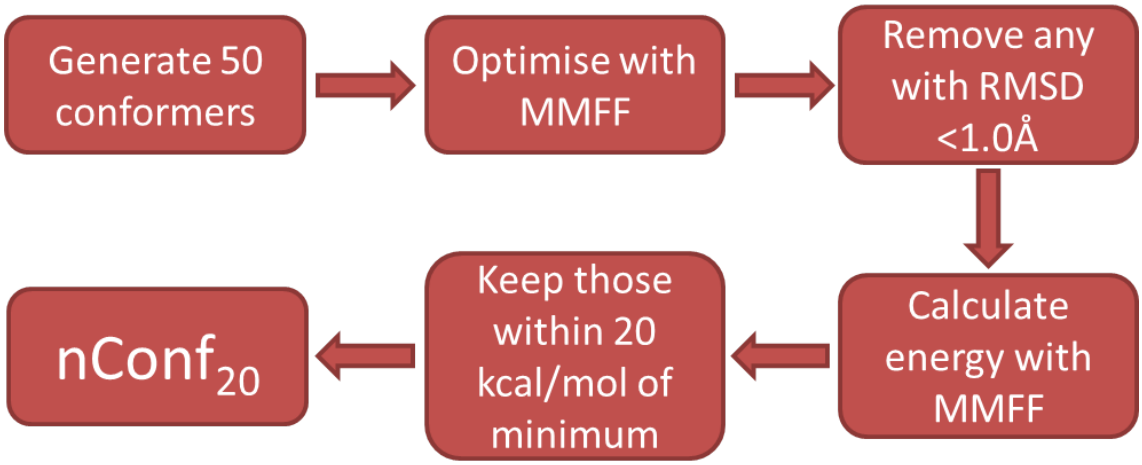
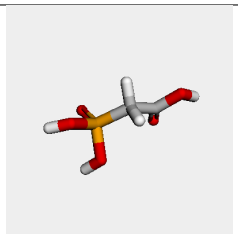
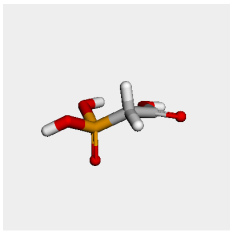
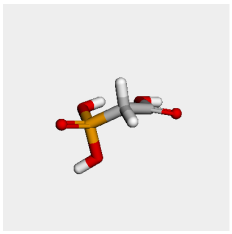
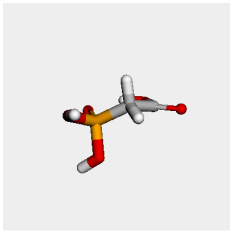
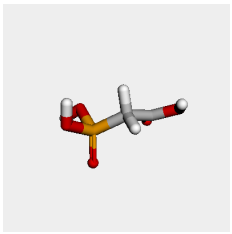
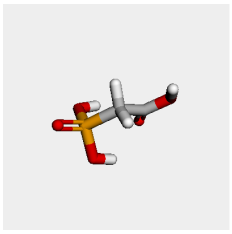


Figure 4.2 Flowchart of the procedure for generating $nConf_{20}$.

Table 4.1 Example nConf$_{20}$ calculation for CSD refcode TERLUX. The lowest energy conformer is not counted, and conformer 41 is above the energy threshold, giving an nConf$_{20}$ value of 4

Conformer ID	Energy (kcal mol^{-1})	Relative energy (kcal mol^{-1})	Conformer
6	-171.3	0.0	
14	-163.3	8.0	
4	-162.6	8.7	
2	-157.3	14.0	
1	-152.5	18.8	
41	-145.9	25.4	

4.4 Descriptor performance

Figure 4.3 shows that rotatable bond count and $nConf_{20}$ capture similar but slightly different information. There is a positive correlation of 0.74 between the two features, but the spread of values of $nConf_{20}$ for each value of RBC is significantly different for those molecules observed to crystallize compared to those which are not. Histograms of the distribution of $nConf_{20}$ values in each class are plotted in Figure 4.4. Those molecules which are not observed to crystallize tend to have a larger value of $nConf_{20}$ than those with the same RBC which are observed to crystallize, indicating that $nConf_{20}$ provides better discrimination between the two classes than RBC. Table 4.2 shows an example molecule where RBC and the new descriptor differ significantly in their estimation of the flexibility of the molecule. Some rotatable bonds cause no change to the molecule, especially when there is symmetry present, information which is captured by $nConf_{20}$.

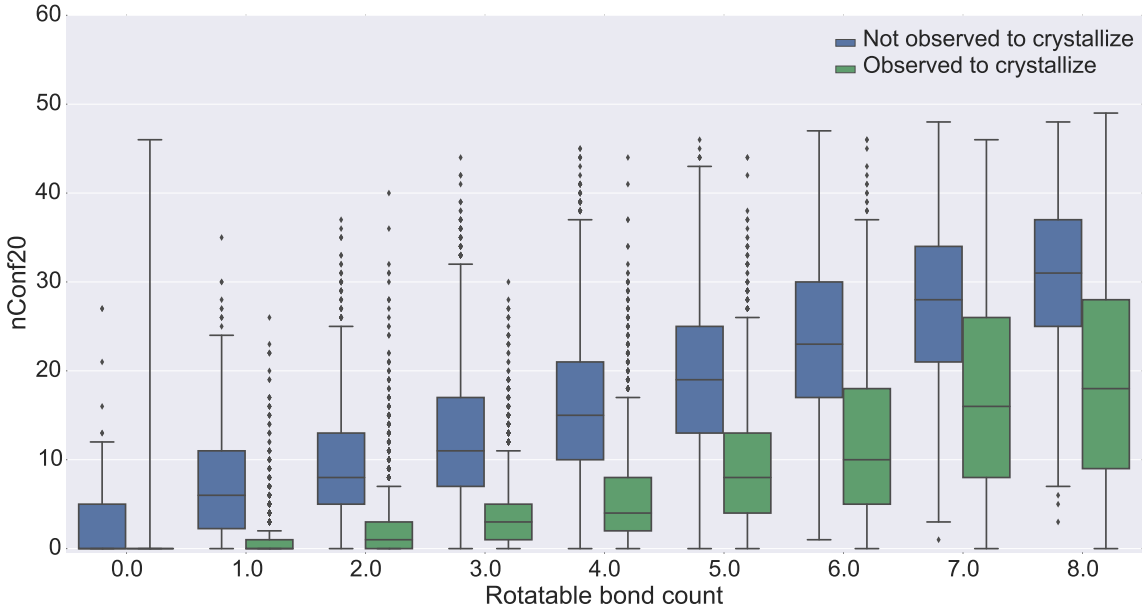


Figure 4.3 Boxplot of the distribution of $nConf_{20}$ for each value of rotatable bond count, split by class. The central line in the box shows the median of $nConf_{20}$ for that value of RBC. The bottom and top of the box denote the 25th and 75th quartiles respectively. The whiskers extend to 1.5 times the interquartile range, and any points outside this are plotted as outliers.

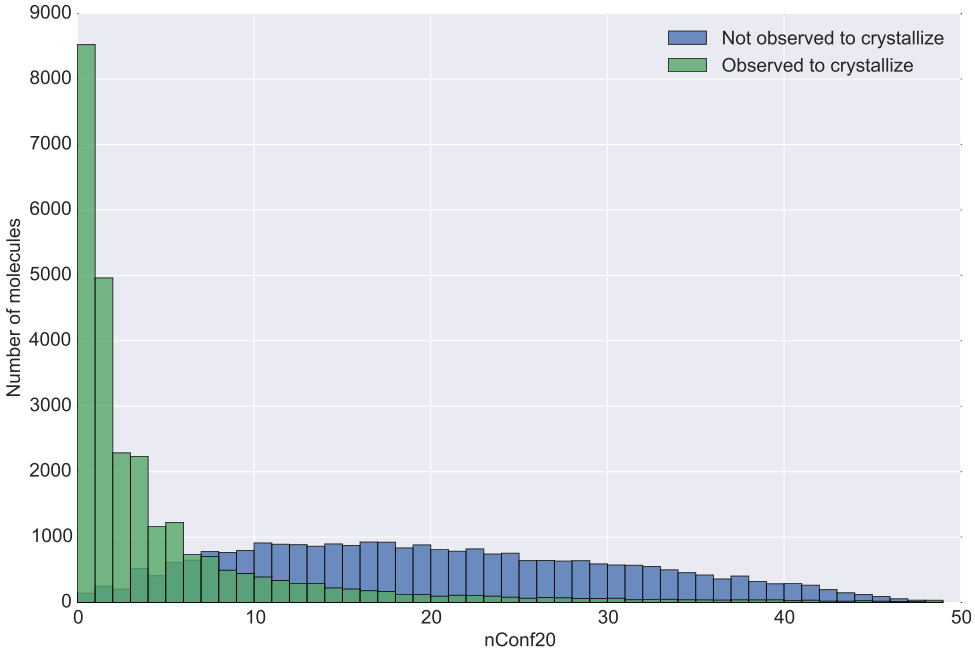
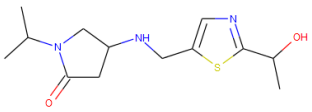
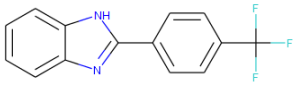
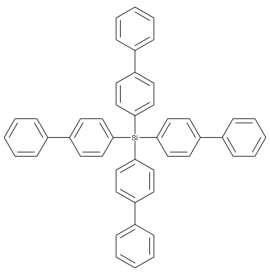
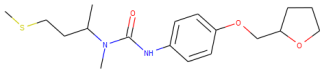


Figure 4.4 Histogram of $nConf_{20}$ for each of the two classes.

Table 4.2 Example rotatable bond counts and $nConf_{20}$ values.

Name	RBC	$nConf_{20}$	Observed to crystallize	Molecule
ZINC000290539224	5	10	No	
ZINC000001235036	1	0	Yes	
ZINC000169816555	8	0	Yes	
ZINC000133698543	8	35	No	

When $nConf_{20}$ is used to make a single variable classifier of molecules observed and not observed to crystallize, the mean predictive accuracy from cross-validation is 84.7%, 7 percentage points better than any other single variable. Figure 4.5 shows that the single variable classifier accuracy is much improved relative to the other descriptors, illustrating that $nConf_{20}$ captures more information than any other single 2D descriptor about the likelihood of a molecule being observed to crystallize.

$nConf_{20}$ was then combined with every other descriptor in turn to create a set of two variable classifiers, and their accuracy was assessed by cross-validation on the training set and prediction on the external test set. In combination with the SMR VSA3 descriptor it produces the best two-descriptor model with a cross-validation predictive accuracy of 88.7%, which is 1.5 percentage points better than any other two variable classifier using $nConf_{20}$, as shown in Figure 4.6.

The heatmap of these two descriptors is shown in Figure 4.7 and shows that while the molecules which are not observed to crystallize have a spread of values for both descriptors, the vast majority of molecules observed to crystallize have a value of 0 for both descriptors. This appears to imply that relatively rigid molecules with no additional conformers and no nitrogen atoms which can only act as H–bond acceptors are likely to be observed to crystallize. The black dotted line denoting the SVM decision boundary between the two classes shows an effective separation, and the predictive accuracy is an increase of 3 percentage points on any other two-variable classifier of crystallization propensity from the original RDKit descriptor set.

When the algorithm was trained with $nConf_{20}$ and all 177 original descriptors, the cross-validation predictive accuracy improves by only 0.1% to 96.0% relative to the model with the 177 descriptors without $nConf_{20}$. The accuracy on the external validation set remains effectively the same, with only 1 further molecule being correctly classified. This suggests that this descriptor provides information to the model that is already indirectly captured by the other original descriptors.

However, the new descriptor captures this flexibility information more directly, as

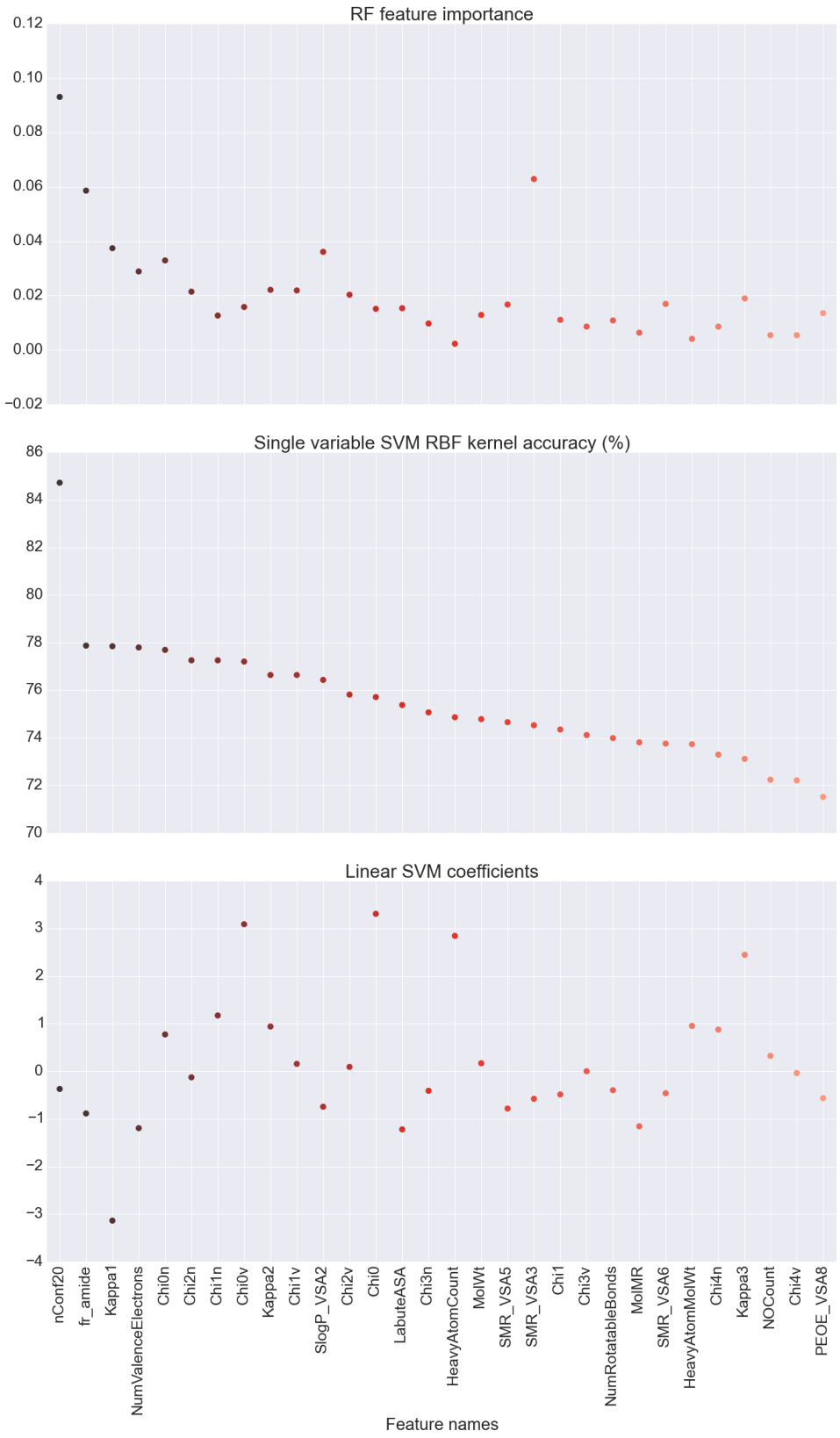


Figure 4.5 Feature importances for a) RF b) SVM with RBF kernel c) linear SVM, trained on the unfiltered set of molecules using RDKit descriptors and nConf20, ranked and colour-coded by single variable SVM RBF accuracy.

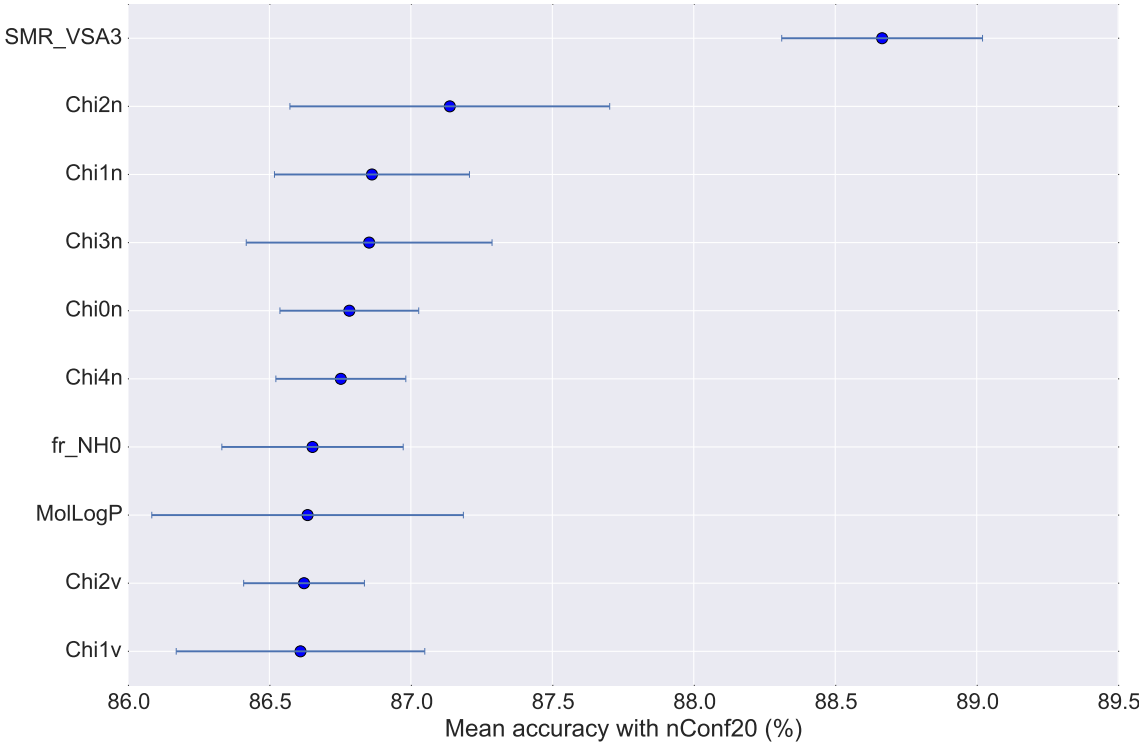


Figure 4.6 Mean predictive accuracy by cross-validation on two variable classifiers trained on the unfiltered data for each RDKit feature with $nConf_{20}$, with error bars showing the standard deviation of the cross-validation scores.

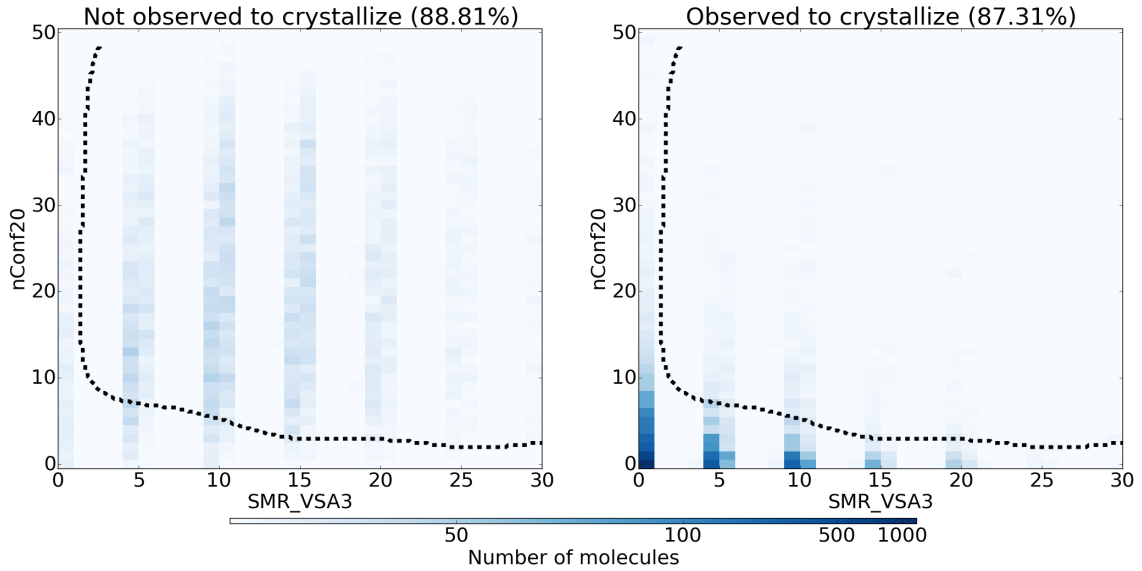


Figure 4.7 Distribution of $nConf_{20}$ against SMR VSA3 for all test molecules in the unfiltered dataset, colour-coded by density of molecules. The dashed line shows the boundary between the crystallizable and non-crystallizable regions as predicted by the SVM algorithm using RBF kernel.

Table 4.3 Confusion matrices for a SVM model with RBF kernel trained on the unfiltered data using a) RDKit descriptors b) nConf$_{20}$ c) RDKit descriptors and nConf20.

Key		RDKit descriptors		nConf$_{20}$		RDKit descriptors with nConf$_{20}$	
T (NC)	F (NC)	**95.4%**	4.6%	**88.8%**	11.2%	**95.4%**	4.6%
F (C)	T (C)	4.3%	**95.7%**	21.1%	**78.9%**	4.2%	**95.8%**
Overall		95.6%		83.9%		95.6%	
Cross-validation		95.9(1)%		84.7(5)%		96.0(0)%	

demonstrated by the high predictive accuracy when used in a single variable classifier. This is important for unpicking and understanding the decisions made by the machine learning process and will also allow it to be used more easily in linear machine learning classifiers and decision trees, which can become very complicated if a combination of variables is required to predict the output, so it would be sensible to include this descriptor in the final model.

The rule extraction analysis further supports the high importance of this flexibility descriptor in performing the classification, as shown in Figure 4.8. The first node in the tree (which mimics the labels provided by the predictive model for the training dataset) provides the best initial split of the data and therefore indicates the most important classification feature. In this case, nConf$_{20}$ is the most important feature; the decision tree shows that the best single–decision approximation of the SVM can be obtained by assuming that the majority of molecules with fewer than 6 low energy conformers are observed to crystallize, while most of those above this cut-off are assumed to not be observed to crystallize. This agrees with the distribution shown in the histograms in Figure 4.4. The leaves below this node show that a single nConf$_{20}$ decision alone reproduces the SVM predictive model with an accuracy of 92% on the crystallizable leaf and 81% on the non-crystallizable leaf (an overall accuracy of 86%).

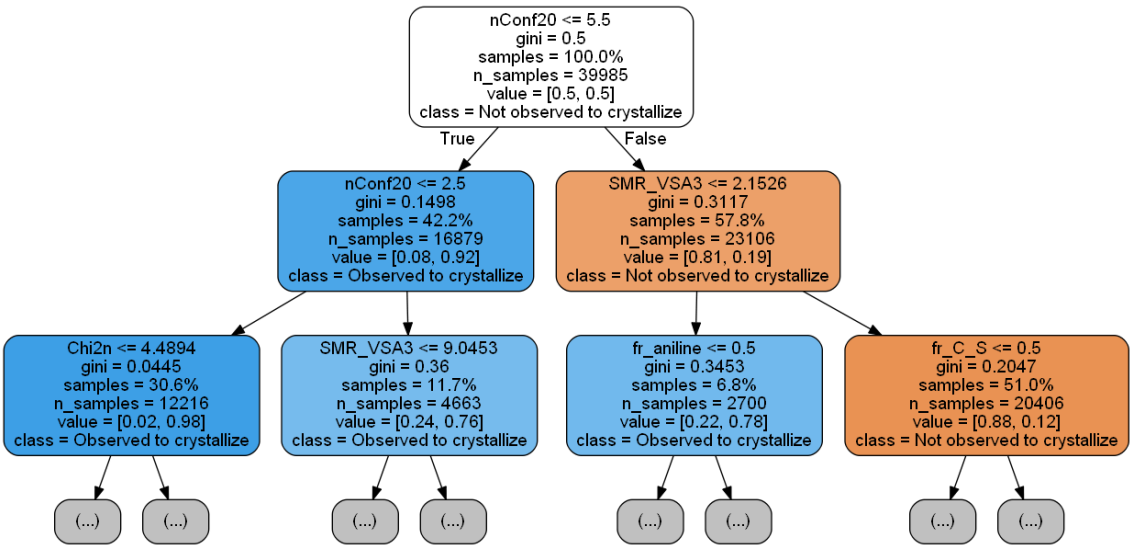


Figure 4.8 Decision tree used for rule extraction from the model trained with RDKit descriptors and $nConf_{20}$ (top 3 levels shown). The gini coefficient is a measure of the impurity of the node. "Samples" indicates the percentage of the total dataset present at that node, and "value" is the proportion of "not observed to crystallize" (orange leaves) and "observed to crystallize" (blue leaves) molecules at the node. Each node has been assigned an overall class based on these proportions.

4.5 Descriptor reproducibility

The stochastic nature of the conformation generation method means that the reproducibility of the method needed to be tested, which was done by repeating the conformer generation step to generate a new set of conformer energies for each molecule. There are two possible changes that could occur: a different energy minimum could be found, or the number of conformers in the energy range could change. The former affects the latter, since the conformers are counted using the energy of the lowest energy conformer as the baseline, so a different baseline could lead to a different $nConf_{20}$ value.

Figure 4.9 shows that in the vast majority of cases, regenerating the conformers finds the same minimum conformer energy. In 76% of cases, the new lowest conformer energy is within 0.5 kcal mol^{-1} of the original conformer energy, so we can assume that the majority of the time the correct global minimum is found. The dis-

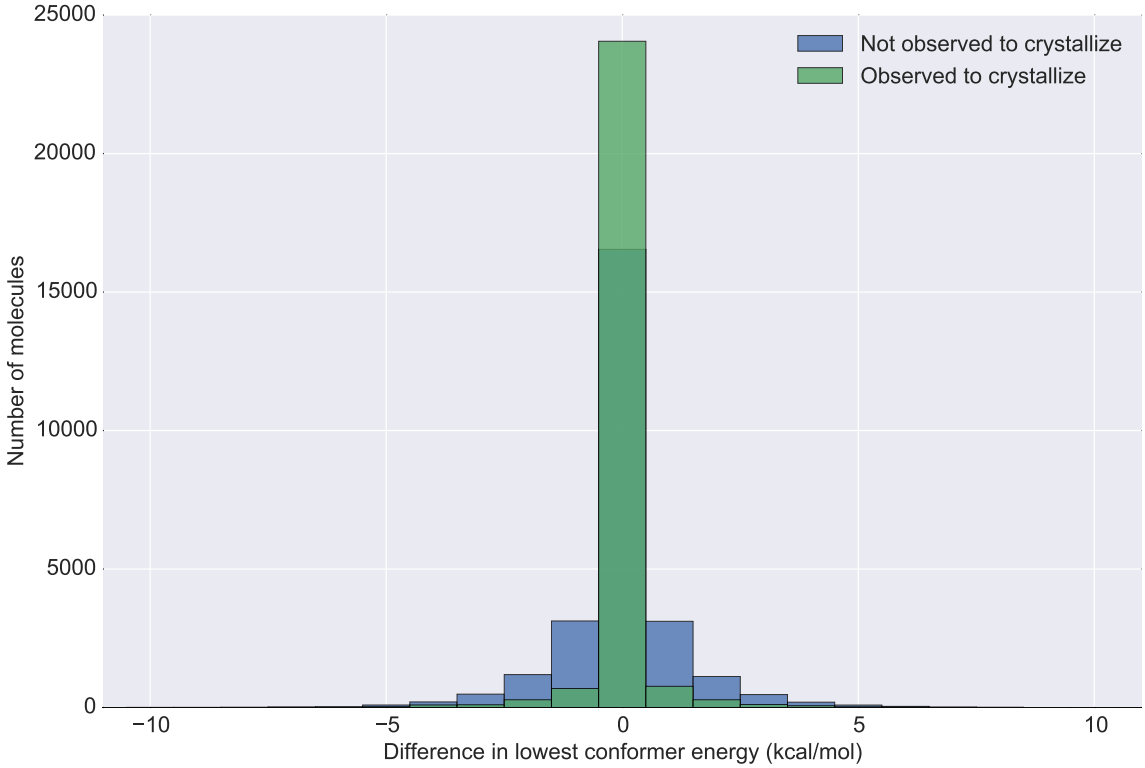


Figure 4.9 Change in lowest conformer energy upon regeneration of conformers with 50 initial conformers.

tribution is not skewed for either class, which indicates that repeating the process with this number of initial conformers is equally likely to find a worse or a better minimum energy. The distribution between the two classes is different, with the correct minimum energy being found for 90% of crystallizable molecules, as opposed to 62% of non-crystallizable molecules. This might be expected as the crystallizable molecules generally have fewer conformers, so the sampling is more likely to find the global minimum. However, the difference in conformer energies is small, with an average difference of 0.43 kcal mol^{-1} and only 0.6% of the molecules showing a difference of more than 5 kcal mol^{-1}, which suggests that the conformer generation method using 50 initial conformers is sufficient to effectively find the global minimum in most cases.

The nConf$_{20}$ value shows little change for the majority of molecules, as illustrated in Figure 4.10. 42% of the molecules show no change in the value of the descriptor, although this is true for far more of the crystallizable molecules (17846) than the

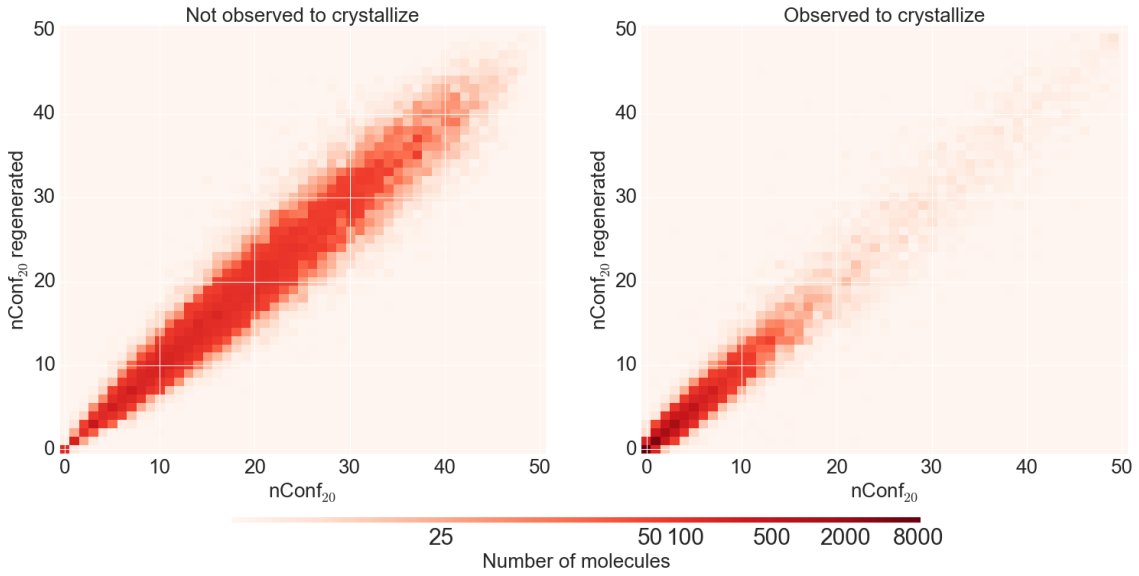


Figure 4.10 Histogram of the original $nConf_{20}$ values and $nConf_{20}$ upon regeneration of conformers with 50 initial conformers for a) non-crystallizable molecules b) crystallizable molecules, colour-coded by number of molecules.

non-crystallizable ones (4766). This is again because the crystallizable molecules are more rigid and have fewer conformers to find, so the sampling is more likely to find them. The correlation coefficient between the original and repeated $nConf_{20}$ values is 0.98, and the average percentage change in $nConf_{20}$ is only 15%, so on the whole the descriptor value is highly reproducible.

To see if the effectiveness of the sampling is affected by the number of initial conformers generated, the conformer generation step was repeated using an initial set of 200 conformers for each molecule. Figure 4.11 shows a similar overall number of molecules have no change in the energy of the lowest energy conformer found, with 76.7% of the molecules having a conformer within 0.5 kcal mol^{-1} of the original lowest energy, compared to 76.1% on repeating with 50 conformers.

However, this distribution differs in that it is skewed towards finding a lower energy conformer in more cases. A higher energy global minimum more than 0.5 kcal mol^{-1} above the original minimum energy conformer was found for only 2.4% of the molecules, of which 267 were crystallizable and 1018 were non-crystallizable. This is to be expected, since sampling from a more extensive set of initial conformers is more likely

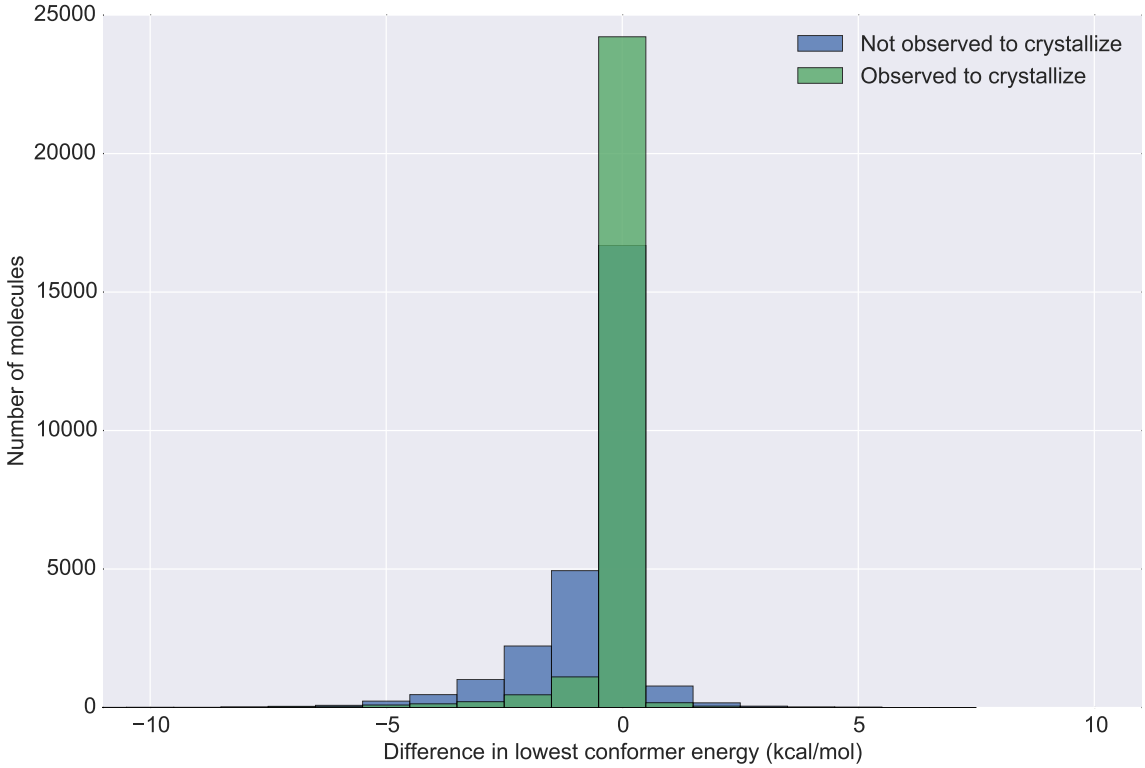


Figure 4.11 Change in lowest conformer energy upon regeneration of conformers using 200 initial conformers.

to contain the global minimum conformer and at the very least should contain the same minimum as was found from a smaller initial set, so it is highly unlikely to find a worse global minimum.

Although a conformer more than 2 kcal mol^{-1} lower in energy than the original global minimum was found for 3490 of the molecules, this represents only 6.5% of the total dataset, so for the vast majority of molecules the global minimum is found to an acceptable accuracy that has little effect on the value of nConf$_{20}$ even for an initial set of 50 conformers.

Overall there are only 1265 molecules which have a lower nConf$_{20}$ by this method, showing that in most cases a greater sampling of initial conformers leads to more nonidentical conformers being found. Figure 4.12 shows that a very high proportion of the crystallizable molecules show no change in the number of conformers within the chosen energy range, and in fact 63% of them have an identical nConf$_{20}$ value to the previous method. These are the most rigid molecules, for which no more low

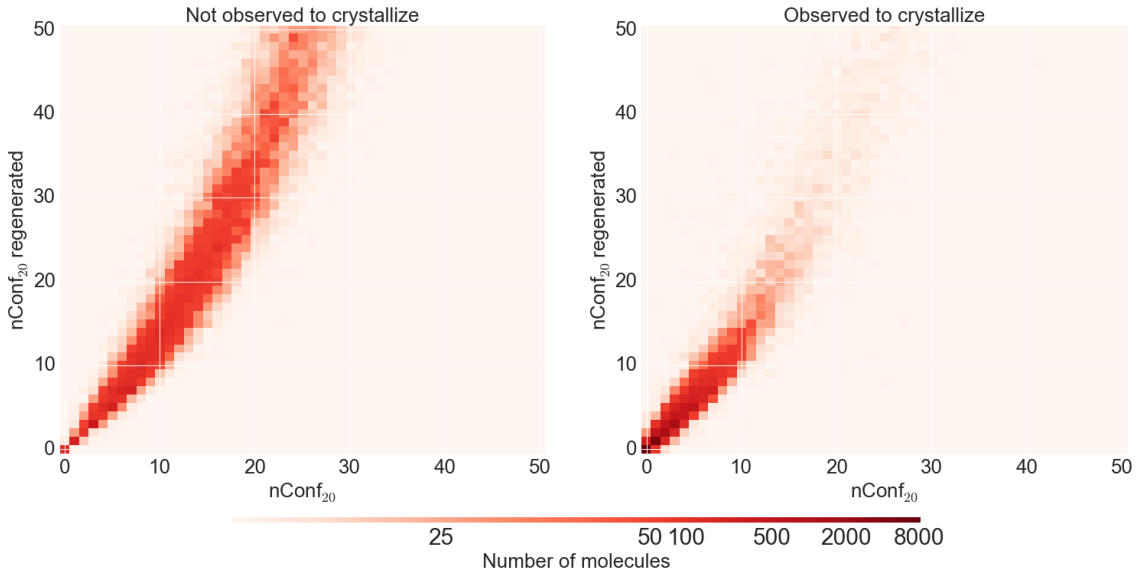


Figure 4.12 Histogram of the original $nConf_{20}$ values and $nConf_{20}$ upon regeneration of conformers using 200 initial conformers for a) non-crystallizable molecules b) crystallizable molecules, colour-coded by number of molecules.

energy conformers are found in the expanded set of initial conformers. For those crystallizable molecules which do show an increase in $nConf_{20}$, only 6224 show an increase of more than 1 in the value of the descriptor, and the average percentage change in $nConf_{20}$ is only 28.7%, showing that for most crystallizable molecules it is almost invariant to the number of initial conformers generated.

However, only 6% of the non-crystallizable molecules show no change in $nConf_{20}$. In these cases, the molecules are highly flexible and so the greater sampling is indeed finding more conformers, with an average increase of 20.9, corresponding to an average percentage difference of 53.6%, so the change is much more significant for the more flexible non-crystallizable molecules. The difference in distributions of $nConf_{20}$ for the crystallizable and non-crystallizable molecules means that this is not a confounding effect, since the greater sampling of the conformer space simply slightly improves the separation of the two classes. The $nConf_{20}$ values for the crystallizable molecules are generally lower and increase by a lesser amount than the non-crystallizable ones. This is borne out by the single variable classifier accuracy of a model built using $nConf_{20}$ from 200 initial conformers, which shows little change

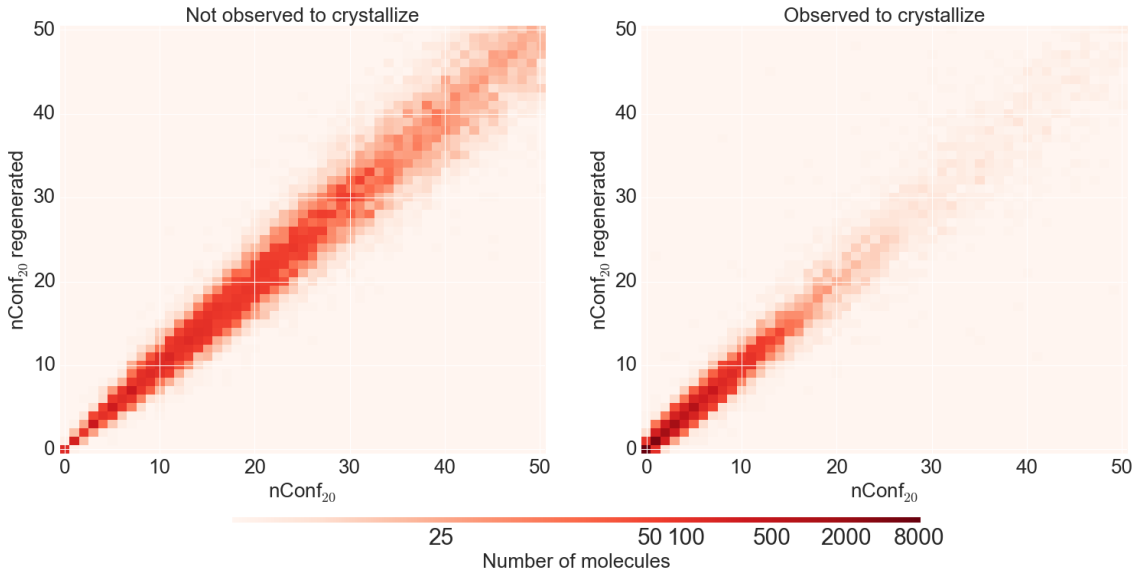


Figure 4.13 Histogram of the nConf$_{20}$ values generated from 200 initial conformers and nConf$_{20}$ upon regeneration of conformers using 200 regenerated initial conformers for a) non-crystallizable molecules b) crystallizable molecules, colour-coded by number of molecules.

compared to the original descriptor, with an identical mean cross-validation accuracy of 84.7(3)%, although the variance in the cross-validation accuracy has decreased slightly.

By using 200 initial conformers, the reproducibility of the value of nConf$_{20}$ improves slightly, as can be seen in Figure 4.13, with an increase in the value of the correlation coefficient to 0.99 upon regeneration of the conformers with a new set of 200 initial conformers. However, this increase is not significant, and using 50 initial conformers gives a sufficient level of reproducibility.

This is important when considering that increasing the number of initial conformers generated does, however, have a significant impact on the computational resources required. More time is required to not only generate and minimise the energies of the conformers, but also to calculate the RMS values of the conformer overlaps with each other for duplicate removal. As the increased computational cost yields little extra predictive accuracy or reproducibility, the original descriptor calculated from 50 conformers was chosen for the purposes of the crystallization predic-

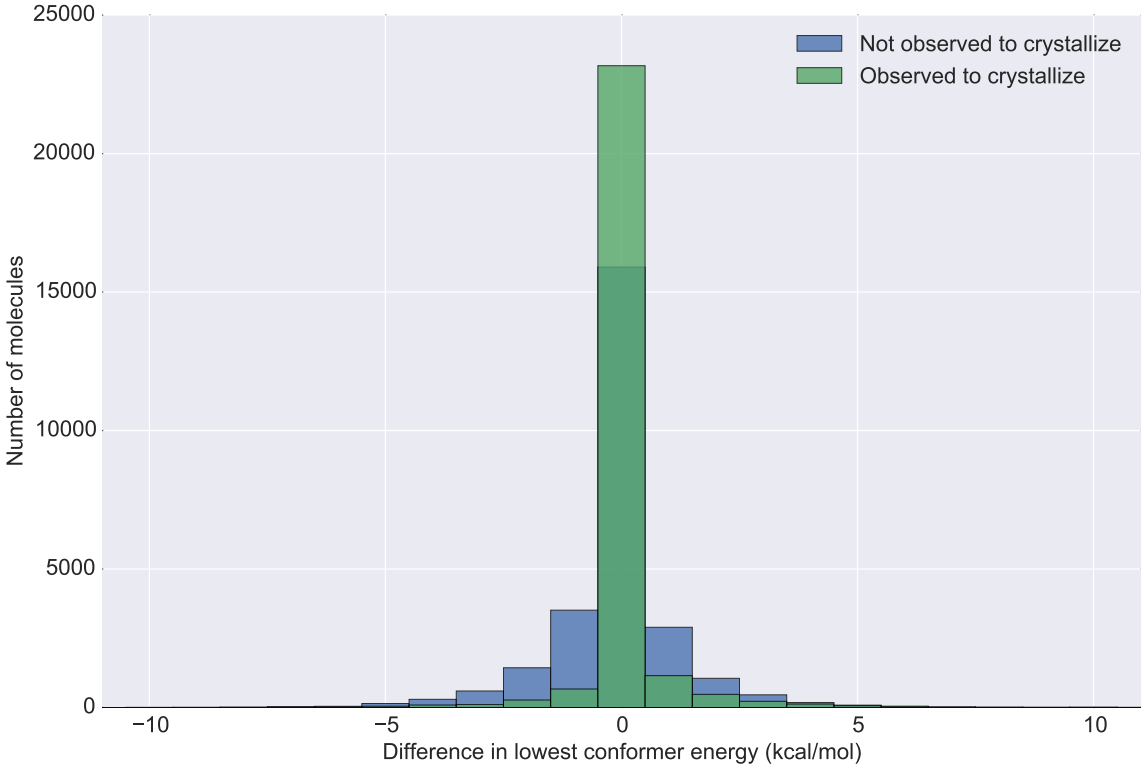


Figure 4.14 Change in lowest conformer energy upon regeneration of conformers using experimental torsion angles.

tion model. The increased number of starting conformers is only useful for applications where a more accurate lowest energy conformer is required, or where a more complete sampling of the conformer energy landscape is desired.

Recently, the RDKit conformer generation method has been updated to combine distance geometry with experimental torsion angles from small-molecule crystallographic data in a method known as ETKDG.[208] This approach was evaluated here using 50 new initial conformers for the $nConf_{20}$ calculation.

Figure 4.14 indicates that while many of the molecules still show little or no change in the energy of the lowest conformer, the distribution of increases and decreases is not as even as with the conformers generated from only distance geometry. The non-crystallizable molecules are more likely to show a decrease in energy, while the crystallizable molecules are more likely to show an increase in energy.

Furthermore, the effect on $nConf_{20}$ of using experimental torsion angles is to decrease the value of the descriptor, particularly for the non-crystallizable molecules, as

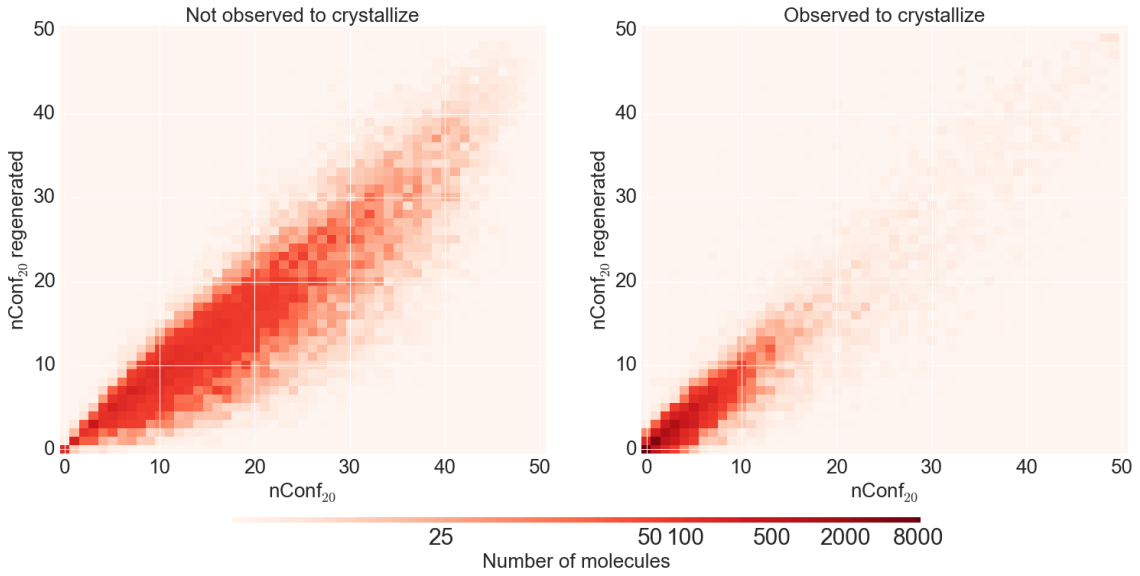


Figure 4.15 Histogram of the original nConf$_{20}$ values and nConf$_{20}$ upon regeneration of conformers using experimental torsion angles for a) non-crystallizable molecules b) crystallizable molecules, colour-coded by number of molecules.

illustrated in Figure 4.15. This is the result of the decrease in the size of the conformational space that is searched by favouring conformations with torsion angles that are found in the small-molecule crystal structures. This means that the 50 conformers which are generated are more likely to contain duplicates which will be removed.

The greater general decrease in nConf$_{20}$ values for the non-crystallizable molecules than the crystallizable ones by this method causes a decrease in the separation of the two classes, which manifests itself as a decrease in the single variable classifier accuracy of nearly 2 percentage points in Table 4.4. Most of the decrease in accuracy is due to poorer predictive capability on the non-crystallizable set, as the spread of some of these molecules extends into the region of chemical space occupied predominantly by crystallizable molecules.

The torsion angles used are taken from the small molecule crystal structures present within the CSD. This means that while they are applicable to the molecules in the crystallizable set, they may not be as suitable for application to the molecules in the non-crystallizable set if there are some differences in the torsion angles that the two sets are likely to adopt. This may introduce some accidental bias by treating the

Table 4.4 Confusion matrices for single variable SVM models trained on the unfiltered data using a) $nConf_{20}$ from 50 new initial conformers b) $nConf_{20}$ from 200 initial conformers c) $nConf_{20}$ from 50 conformers generated using experimental torsion angles.

Key		50 conformers repeated		200 conformers		50 conformers with experimental torsion angles	
T (NC)	F (NC)	**88.8%**	11.2%	**88.1%**	11.9%	**85.2%**	14.8%
F (C)	T (C)	20.9%	**79.1%**	19.8%	**80.2%**	20.3%	**79.7%**
Overall		84.0%		84.1%		82.5%	
Cross-validation		84.7(4)%		84.7(3)%		82.9(6)%	

two classes of molecules using information derived from only one of the sets. The torsion angles used are also taken from molecules in the solid state, whereas the conformations of interest in this case are the potential conformations in solution, which may be different to those observed in the crystal structures.

While use of the information present in the CSD is useful for molecules which are likely to be crystallizable, it is inappropriate to use such prior knowledge in this case, particularly as it actually creates a descriptor which is less able to discriminate between the two classes. The most appropriate method, which will be carried forward, is the method involving generation of 50 conformers using solely distance geometry followed by MMFF energy minimisation.

4.6 Conclusions

$nConf_{20}$ captures information about molecular flexibility more comprehensively than previous 2-dimensional descriptors, providing single-variable predictive models which are up to 7 percentage points better than any 2-dimensional descriptor. Inclusion with the overall descriptor set causes a negligible increase in predictive accuracy, showing that the other descriptors capture this information in combination, but $nConf_{20}$ simplifies the rule extraction process, so the final model incorporates this descriptor in addition to the RDKit descriptors. Descriptor reproducibility can be improved

by increasing the number of initial conformers generated, at a large computational cost, while using experimental torsion angles reduces model performance. This 3-dimensional flexibility descriptor was incorporated into the final model.

Chapter 5

Experimental Validation

5.1 Introduction

The percentage accuracies of the models as presented in the previous chapters were subject to possible biases due to the possibility that some molecules in the non-crystallizable group of molecules were in fact crystallizable, but simply had never been crystallized (or at least had never been analysed by SXRD and reported in the literature). The aim of the experiments described in this chapter was to experimentally validate the predictions of the chosen model (RBF-SVM with RDKit descriptors) using a blind microcrystallization screen and controlled cooling experiments. The model is found to be accurate at predicting whether a molecule will crystallize, and also provides information about the cooling rate required for crystals to grow to a size suitable for SXRD.

Contents

5.2 Blind test

In order to validate the theoretical accuracy of the predictive model, a blind microcrystallization screen was carried out as described in Section 2.4.1. By purchasing, analysing and attempting to recrystallize a subset of the test molecules, containing both molecules which the model predicted to be crystallizable and ones predicted

to be non-crystallizable, an experimental success rate was obtained which could be compared to the theoretical values.

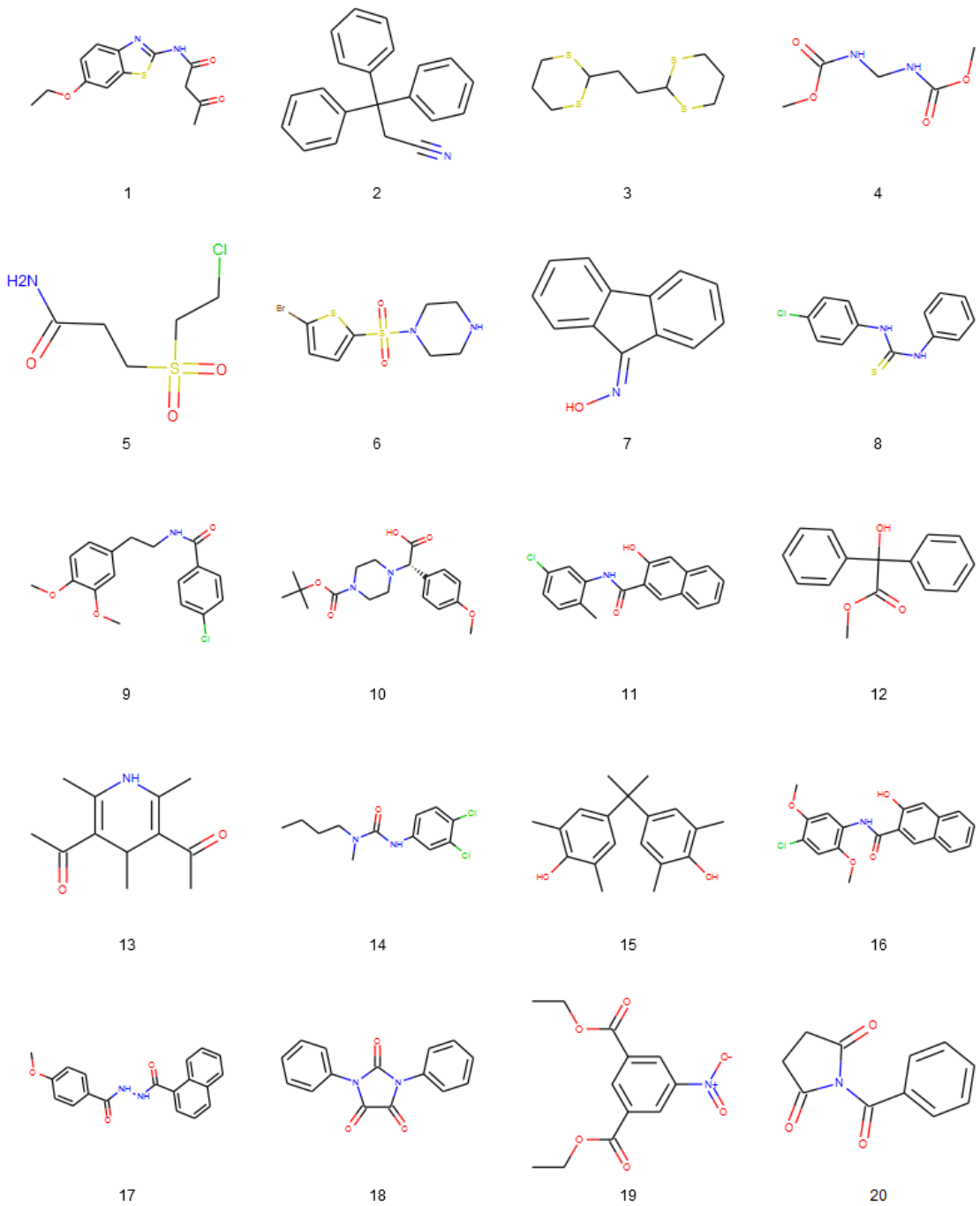


Figure 5.1 Molecules used for the blind test.

Of the 20 compounds which were ordered, shown in Figure 5.1, sample 18 was excluded as it was found not to be the compound that was ordered on determination

of the crystal structure, and this was verified by mass spectrometry. Table 5.1 shows the predicted crystallization probability for each compound, the solvent used for successful recrystallization where applicable, as well as the crystal structure and details about the unit cell for those materials which gave a successful structure solution.

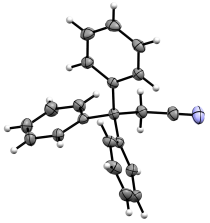
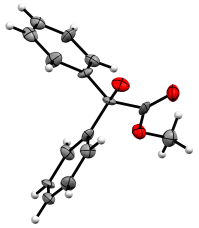
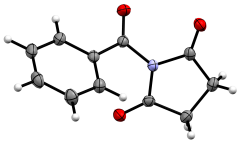
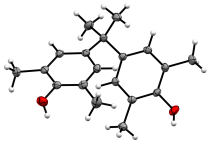
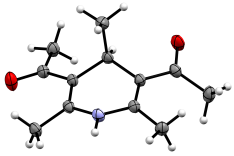
7 of the 10 molecules strongly predicted to be crystallizable by the model (greater than 0.75 probability of crystallization) were successfully recrystallized from one of the solvents with crystals large enough to be used for SXRD. Of these, 6 previously unreported crystal structures were obtained, although one of these has subsequently been added to the CSD (sample 20, refcode LONSEO).

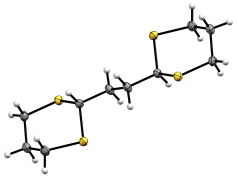
For sample 7, high quality diffraction patterns were obtained, but the structure solution failed due to potential twinning of that particular crystal. However, this material, fluoren-9-one oxime, has subsequently had a crystal structure determined and deposited in the CSD (refcode NIXWUO), confirming that this material is indeed crystallizable. The fact that this screen did not obtain single crystals from room-temperature dichloromethane, whereas single crystals were subsequently reported to have been obtained from cold dichloromethane,[209] illustrates the vast range of potential conditions that could be used for recrystallization screens.

Of the remainder, compound 4 did give crystals which were needle-like and unsuitable for SXRD, suggesting that although it did not crystallize to a sufficiently large size under the range of experimental conditions tested, slightly more careful selection of the crystallization conditions may have allowed crystals to grow to a more suitable size.

For the non-crystallizable group (with predicted probabilities below 0.75), no crystals of sufficient size or quality were obtained for any of the 9 compounds, although compounds 16 and 17 did give needle-like crystals which were unsuitable for SXRD. This was confirmed by attempting to obtain a diffraction pattern using a crystal of sample 16, which provided a low quality pattern indicating large amounts of strain or modulation of the structure, making the crystals unsuitable for use in structure de-

Table 5.1 Blind test results.

Name	Propensity	Cell parameters	Space group	Solvent	Result	RBC
2	1.000	9.01Å 9.05Å 11.08Å 101.05° 92.86° 117.601°	P-1	DMF		4
12	1.000	9.21Å 24.77Å 27.82Å 71.89° 89.69° 89.04°	P-1	EtOH		3
20	1.000	12.40Å 9.89Å 7.89Å 90° 103.89° 90°	$P2_1/c$	Ethyl acetate		1
7	1.000	–	–	Ethyl acetate	Crystals not suitable for X-ray diffraction, NIXWUO in CSD 2014	0
4	1.000	–	–	–	Needles	2
8	1.000	–	–	–	Remained as powder	2
19	1.000	–	–	–	Remained as powder	5
15	0.996	25.52Å 11.53Å 10.64Å 90° 90° 90°	Pbcn	EtOH		2
13	0.987	7.43Å 8.95Å 9.20Å 109.09° 103.98° 93.63°	P-1	DMF		2

Name	Propensity	Cell parameters	Space group	Solvent	Result	RBC
3	0.957	4.87Å 10.67Å 12.07Å 90° 92.69° 90°	$P2_1/n$	DMF		3
11	0.576	–	–	–	Remained as powder	2
17	0.570	–	–	–	Needles	3
5	0.562	–	–	–	Remained as powder	5
6	0.500	–	–	–	Remained as powder	2
1	0.370	–	–	–	Remained as powder	5
9	0.358	–	–	–	Remained as powder	6
14	0.267	–	–	–	Remained as powder	4
16	0.209	–	–	–	Needles	4
10	0.130	–	–	–	Remained as powder	4

termination. The remainder of the materials remained as powders. This is not to say that they may not crystallize with more careful selection of solvents and conditions, but they are clearly more difficult to crystallize under this range of simple conditions.

This gave an overall predictive accuracy of 84%, lower than the accuracy obtained from the computational testing, but promising considering the small sample size and the relatively small range of crystallization conditions. The model is able to correctly identify materials that crystallize but have not been added to the CSD and so would be incorrectly labelled as “non-crystallizable”. The “blind” nature of this test meant that no undue effort or experimenter bias was involved in attempting to obtain crystallizable or non-crystallizable results, ensuring the integrity of the test. The area under the ROC curve (Figure 5.2) for the list ranked by crystallization probability is 0.84, with all 7 successful crystals being found in the top 10 molecules in the list, indicating that the predictive model is effective at ranking crystallizable materials relative to non-crystallizable ones. This shows that the model is a useful tool for screening materials to find candidates to focus recrystallization resources on.

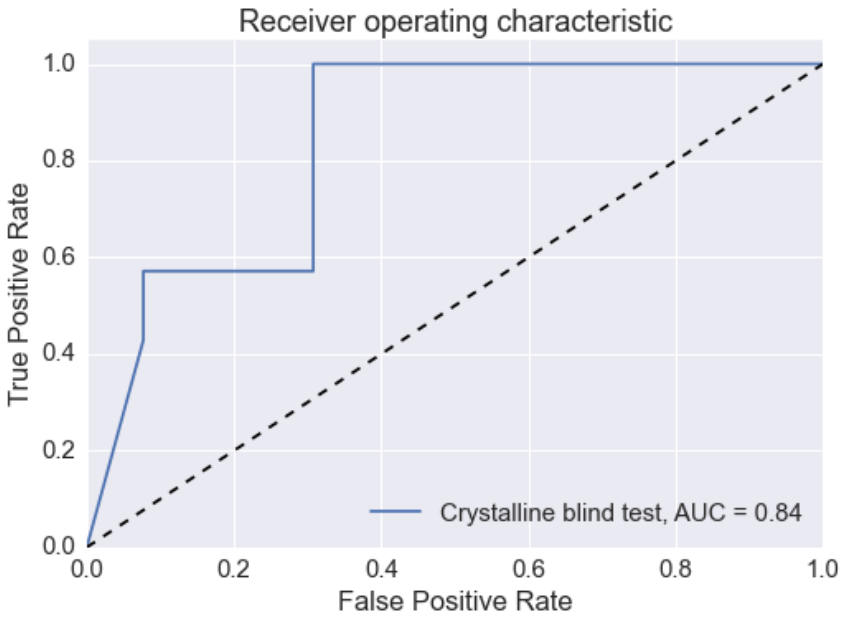


Figure 5.2 ROC curve for the blind test.

5.3 Controlled cooling experiments

Another consideration in the crystallization process is the time needed for each molecule to form crystals. As discussed earlier, some crystallites have slower nucleation and growth rates. For certain molecules this means that the quality of the crystals produced may critically depend on the speed of the increase in supersaturation of the solution. Some molecules may not produce crystals of good enough quality for structure determination by SXRD via slow evaporation crystallization. It is hypothesised that the molecules that are predicted to be harder to crystallize will have slower nucleation and growth rates and they will require a slower increase in supersaturation of the solution in order to produce large single crystals. Controlled cooling experiments were used to test this hypothesis and also give a quantifiable measure of how easy it is to produce single crystals. The method for this experiment was outlined in Section 2.4.2, using the materials shown in Figure 5.3.

Table 5.2 shows a general trend that materials with lower predicted crystallization propensity require a longer crystallization time to grow crystals. 7 of the 9 materials with the highest crystallization propensity formed single crystals after a cooling time

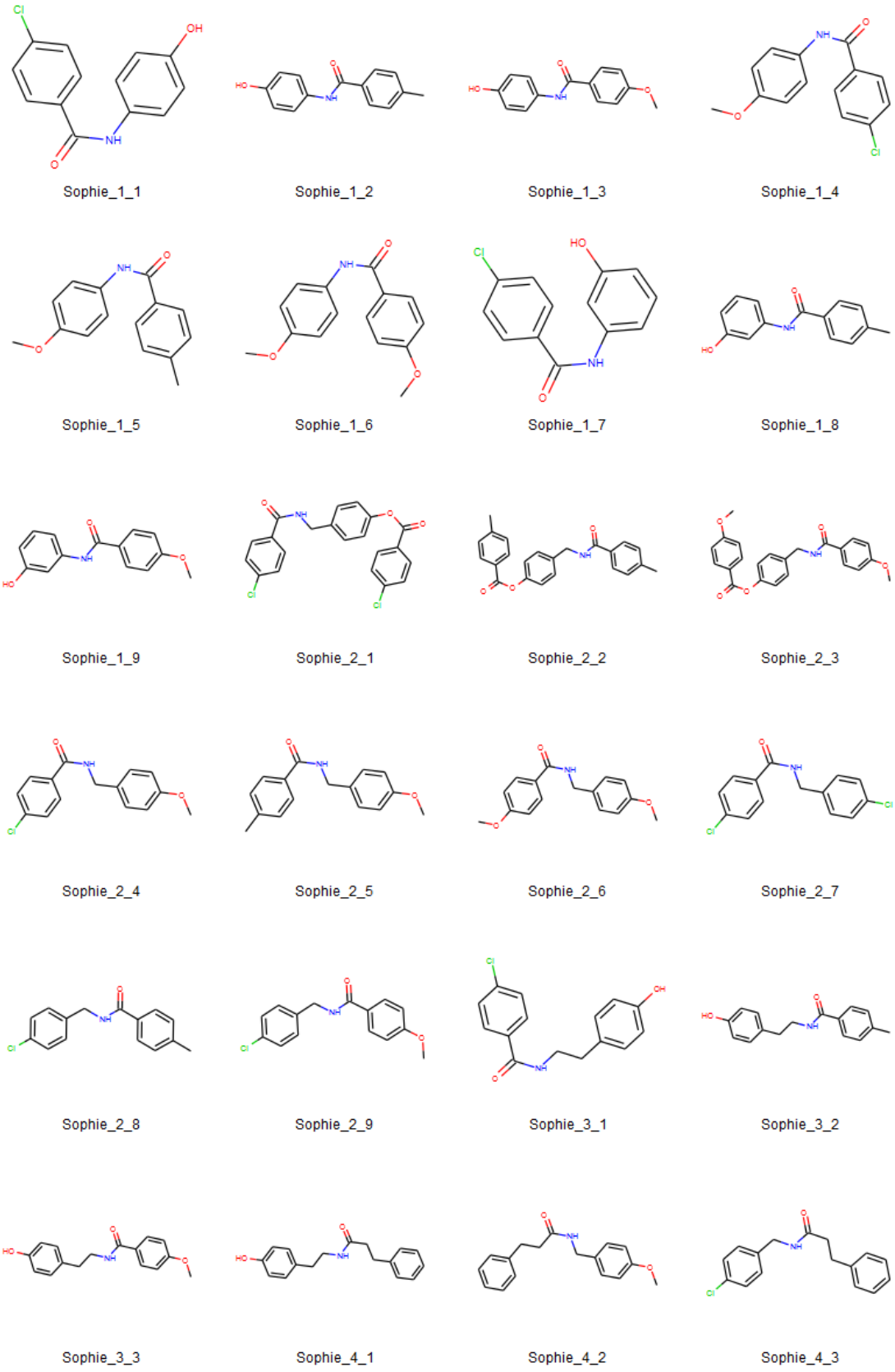


Figure 5.3 Materials used for the controlled cooling experiment.

Table 5.2 Results of controlled cooling experiments with acetone as the solvent, sorted by predicted crystallization propensity. • indicates that the crystals produced were of sufficient quality for SXRD to be successful. ○ indicates that crystals were produced, but they were of insufficient quality for successful SXRD.

Molecule	Crystallization propensity (%)	1 week	3 days	1 day
1.4	98.7	○	○	
1.8	98.6	•	○	○
1.1	98.5	○	○	○
1.7	98.4	•	•	•
1.2	98.3	•	•	○
1.6	98.0	•	•	•
2.7	97.7	•	•	
1.5	97.4	○	○	○
1.3	96.5	•	○	○
1.9	95.8	•	•	
2.8	91.0	•	•	
2.9	88.1	•	•	
3.2	82.8	•	○	
2.5	82.7	•	•	
2.6	82.2	•	•	
2.4	82.0	•	•	
3.1	79.9	•	•	
3.3	75.4	○	○	
2.1	70.5	○	○	
4.3	54.5	•		
4.2	47.8			
4.1	45.9			
2.2	40.9	○	○	
2.3	37.3	○	○	

of only a day, although only 2 of these were sufficiently high quality for SXRD to be successful. None of the rest of the materials formed any crystals at all with this cooling rate.

A cooling time of 3 days was sufficient for all but three of the materials to form single crystals, with 11 materials producing crystals suitable for SXRD on this timescale. One material that only produced crystals which were unsuitable for SXRD after one

day gave higher quality crystals by this slightly slower cooling rate. Of the three materials which did not form any crystals after 3 days, all three were ranked in the bottom 5 of the list.

With a cooling time of a week, one material which formed no crystals at all in the previous experiments gave crystals of SXRD quality, three materials which gave poor quality single crystals after 3 days gave SXRD quality crystals, while two materials still gave no crystals.

These experiments show that the cooling rate has less of an impact on the crystallizability of the material for which the predicted ease of crystallization is high. This would support the hypothesis that the metastable zone width of these crystallizable materials is wider, so a faster cooling rate would still allow sufficient time in the metastable zone for crystal growth to occur. However, the quality of the crystals produced is affected by the cooling rate, with higher quality crystals sometimes being achieved only with a slower crystallization rate. A 3 day cooling period appears to provide the optimum rate for successful crystallization for this set of materials, although the appearance of crystals of material 4.3 only after a cooling time of a week highlights the value of using slower cooling rates for materials with lower predicted crystallization probability.

Although the lack of crystals over any time period for materials 4.1 and 4.2 would strongly suggest that these materials are not crystallizable, there are a number of reasons why they may not have crystallized under these conditions. The solvent choice may not have been appropriate for these materials, a slower cooling rate may have been necessary for these materials, or a lower final temperature may have been beneficial for crystal growth. However, this does support the use of the crystallization predictions as indicators of the ease of crystal growth, seeing as these materials would require a wider range of potential conditions to be attempted in order to achieve crystallization.

Similarly, the formation of only low-quality crystals for 7 of the materials does not

necessarily mean that these materials cannot grow high-quality crystals, rather that the solvent choice was not the optimal one for these particular materials. However, the fact that they formed crystals at all suggests that with a more careful solvent choice, SXRD quality crystals may be obtained.

This experiment suggests that materials with a predicted crystallization propensity greater than 95% are highly likely to form crystals under most cooling rates, although the quality of the crystal is affected by the cooling rate. Those between 50% and 95% are likely to form crystals provided a slow enough cooling rate is used. Those below 50% are unlikely to form crystals, and would require more effort to crystallize.

5.4 Conclusions

The experiments described in this chapter show that the model obtains around 80% accuracy for prediction of crystallization tendency of molecules that were previously not known to crystallize, demonstrating that the CSD contains enough information to be able to identify materials which are incorrectly labelled as non-crystallizable. The cooling rate experiments show that crystallization tendency is related to cooling rate required, with the optimum cooling rate for this set of materials being a 55 °C to 30 °C cooling over 3 days. Slower cooling rates are necessary for materials which are harder to crystallize, and reduced cooling rates also improve crystal quality in most cases.

Chapter 6

Powder Diffraction Studies

This chapter describes the experiments carried out on a selected set of materials to assess the complexity of determining crystal structures from powder diffraction data and to evaluate the relationship between the structure and microstructure of these materials and their predicted crystallization propensity. Although no obvious relationship between the microstructure of the material and the crystallization propensity is found, there are indications of a link between the lattice energies of the materials and the model predictions.

Contents

6.1 Introduction

The aim of this experiment was to examine a subset of materials in detail using X-ray powder diffraction methods. By selecting materials from several different molecular families and attempting to determine the structures by high-resolution powder diffraction, the relationship between the crystallization propensity predictions from the model and the ease of structure determination can be assessed. Additionally, some insight can be gained into any structural reasons for the lack of single crystal growth.

6.2 Molecule selection

Materials were selected in collaboration with Max Pillong and Trixie Wagner at Novartis. Firstly, a set of available compounds needed to be clustered. A set of publicly available compounds which Novartis had in their possession was used as the starting dataset. Very small molecules with fewer than 10 heavy atoms were excluded, as were fatty acids, to leave a set of 16949 molecules. For each molecule, the Morgan fingerprint[175, 176] with a radius of 3 was calculated as a bit vector. These fingerprints were used to calculate the pairwise distance matrix between all the molecules based on the Tanimoto similarity of the fingerprints.

The Taylor-Butina clustering algorithm[179, 180] was used to separate the molecules into clusters based on this distance matrix, which generated 9641 clusters, of which 7123 contained only a single molecule. 146 of the clusters contained at least 10 molecules, for which class probabilities were calculated using the final model. These probabilities lie on a scale from 0 to 1, with a value of 1 indicating a strong propensity to crystallize and a value of 0 indicating a low propensity to crystallize. 12 families were selected from these to give a spread of crystallization propensity probabilities, as shown in Figure 6.1, with the centroid of each cluster being shown in Figure 6.2, a total of 195 materials.

Families 1–5, 9 and 10 are strongly predicted to be easy to crystallize, families 7, 11 and 12 are strongly predicted to be hard to crystallize, and family 6 and 8 show a spread of crystallization propensities.

6.3 Attempted crystal structure determinations

The powder diffraction patterns were collected as described in Section 2.5. Indexing was attempted by picking the first 20 peaks and running the indexing routine within TOPAS.[210] For 12 of the materials, no diffraction pattern was collected because a capillary could not be loaded either due to the electrostatic nature of the sample, or

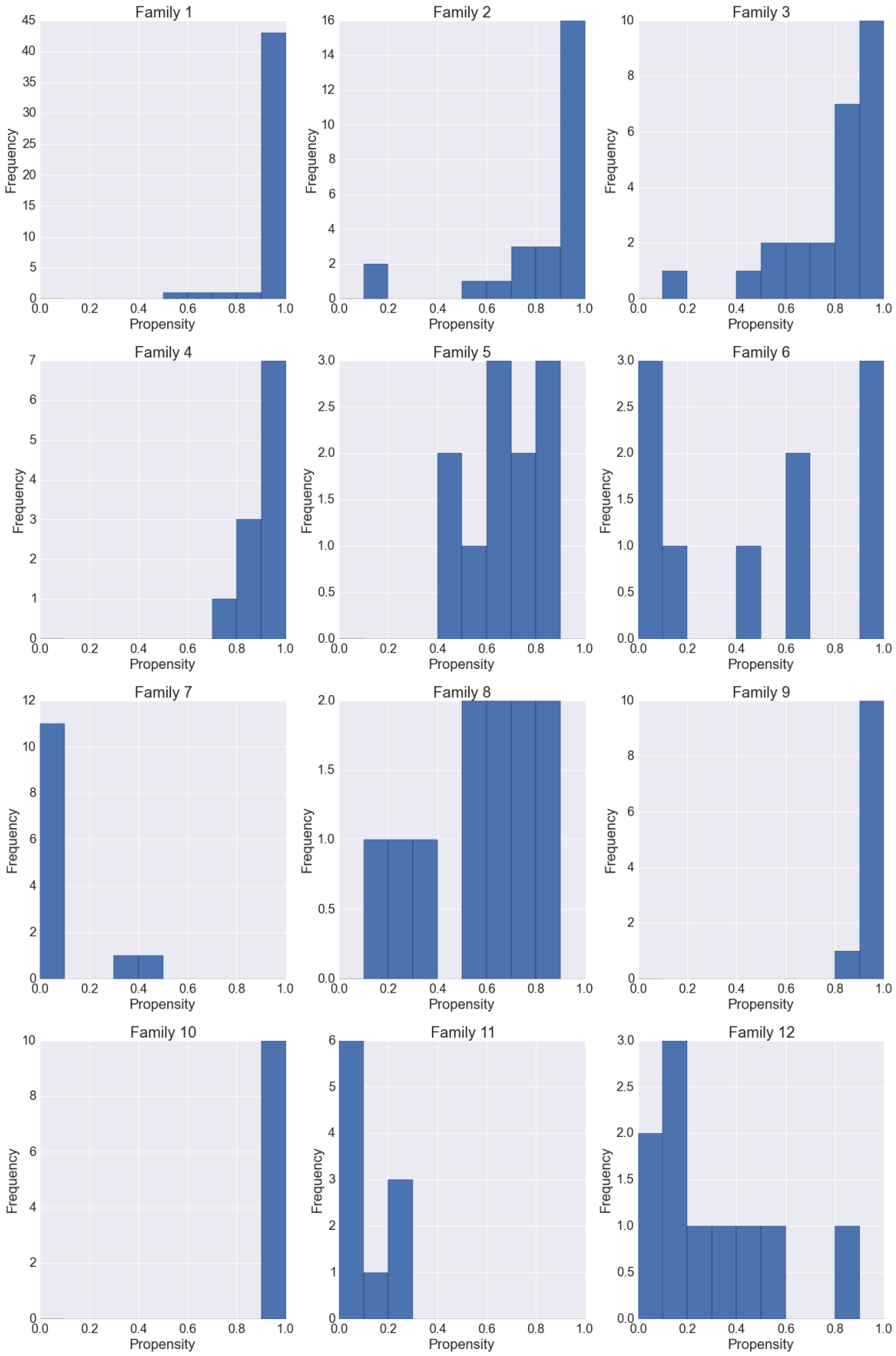


Figure 6.1 Probability distribution of likelihood to crystallize for each family, with 0 being unlikely to crystallize and 1 being likely to crystallize.

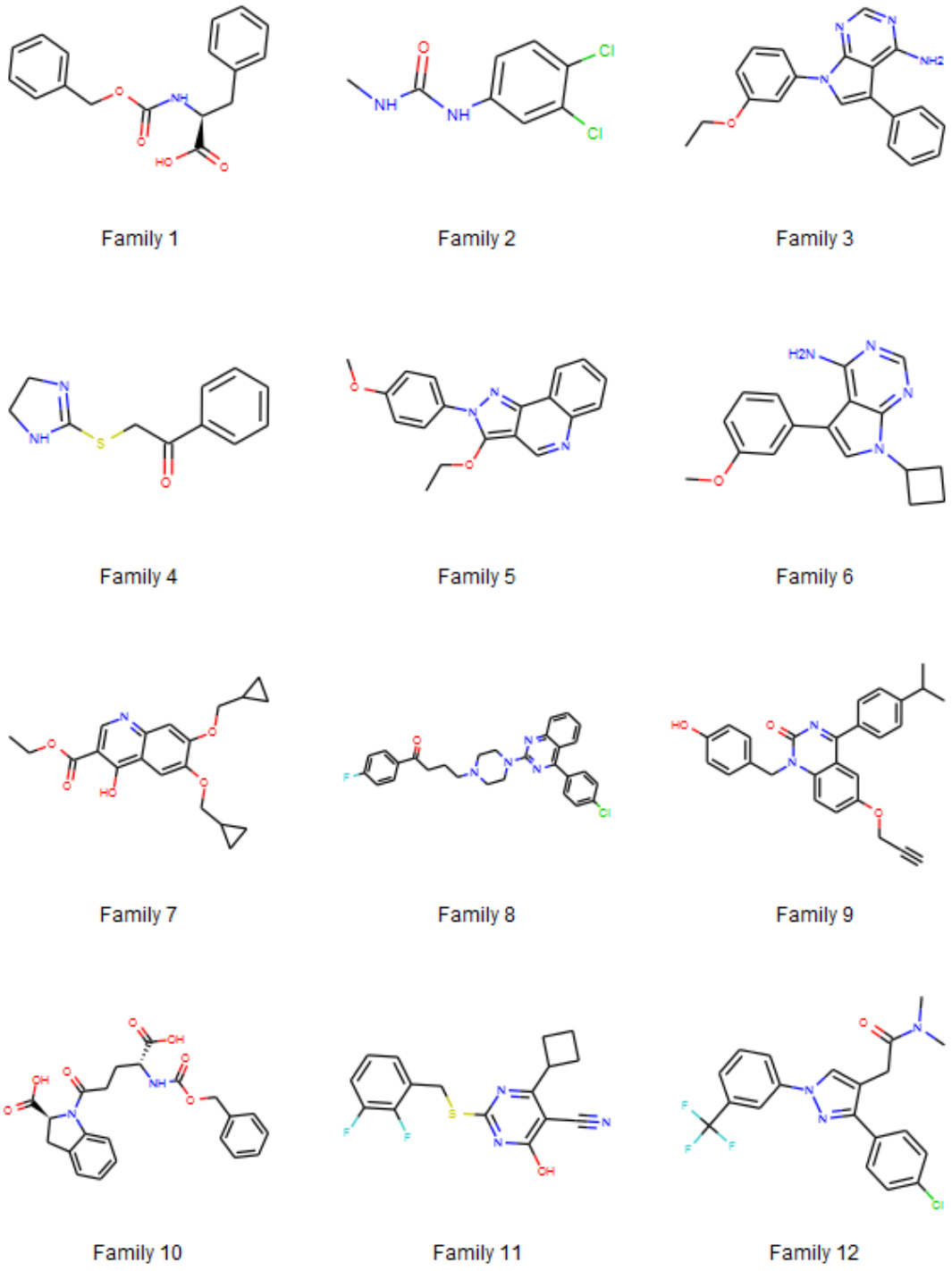


Figure 6.2 Centroids of the chosen clusters.

because the sample was an oil.

Where multiple possible space groups were suggested by the TOPAS indexing program, each was tried. Structure solution was attempted by the simulated annealing procedure within DASH, with torsion angle probability distributions set using data imported from Mogul.[211]

Of the 183 materials that were able to be loaded into capillaries and therefore successfully had a powder diffraction pattern collected, lattice parameters were identified for 96 materials. From these, 74 crystal structures were solved by this method and the details of the lattice parameters and crystal structures are displayed in Tables 6.1–6.21. Five crystal structures were found to have previously been determined and added to the CSD (1-32, 1-38, 1-45, 2-13, 2-16), while one structure was a pure form of a material previously only crystallized as a solvate (1-44). Two materials which are present in the CSD produced powder diffraction patterns from which the structure could not be determined (2-7, 2-19).

Crystallite size and strain parameters were extracted from the powder diffraction patterns by Rietveld analysis as implemented in TOPAS, using input files generated by DASH. In the cases where lattice parameters were determined but no structure was solved, Pawley analyses were performed using TOPAS. This approach gave reliable size and strain parameters for 21 of these materials, with one further material showing *hkl*–dependent asymmetric line-broadening from which no microstructure parameters could be reliably extracted. For both the Pawley and Rietveld methods, the Voigt FWHM of the peaks was used to calculate the size and strain.

Table 6.1 Crystal structure solutions 1

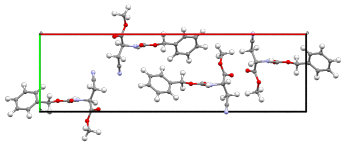
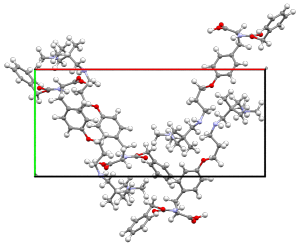
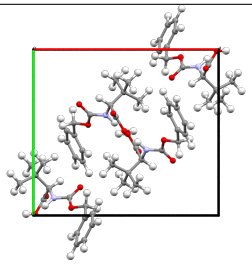
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-1	1.00	13.59Å 5.00Å 22.09Å 90° 96.22° 90°	$P2_1$, Z'=2	–	No solution
1-2	0.99	10.58Å 10.86Å 12.25Å 94.08° 96.65° 90.53°	P-1, Z'=2	–	No solution
1-3	0.80	30.39Å 8.84Å 4.93Å 90° 90° 90°	$P2_12_12_1$	12.46	
1-4	1.00	–	–	–	Unindexed, no solution
1-5	0.53	25.83Å 12.04Å 8.41Å 90° 90° 90°	$P2_12_12_1$	9.25	
1-6	1.00	15.23Å 13.31Å 7.14Å 90° 102.2° 90°	$P2_1/a$	11.57	
1-7	1.00	–	–	–	Unindexed, no solution
1-8	0.72	10.31Å 13.79Å 17.97Å 92.71° 105.08° 108.98°	P1	–	No solution

Table 6.2 Crystal structure solutions 2

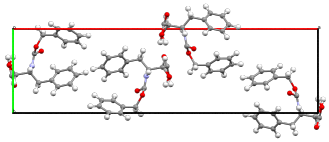
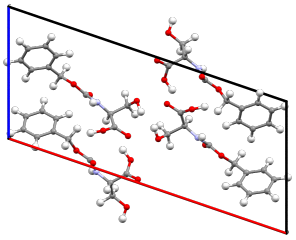
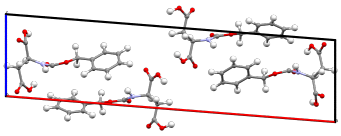
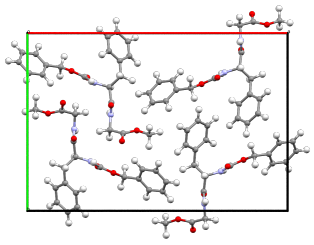
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-9	0.99	33.63Å 9.26Å 5.20Å 90° 90° 90°	$P2_12_12_1$	15.11	
1-10	0.99	–	–	–	Oil, capillary could not be loaded
1-11	0.97	–	–	–	Capillary could not be loaded
1-12	1.00	–	–	–	Capillary could not be loaded
1-13	1.00	22.97Å 4.99Å 10.31Å 90° 108.82° 90°	$P2_1$, Z'=2	13.21	
1-14	1.00	–	–	–	Unindexed, no solution
1-15	1.00	30.58Å 5.17Å 7.58Å 90° 94.50° 90°	$P2_1$, Z'=2	13.8	
1-16	0.97	23.41Å 15.98Å 4.91Å 90° 90° 90°	$P2_12_12_1$	11.04	
1-17	0.99	–	–	–	Unindexed, no solution

Table 6.3 Crystal structure solutions 3

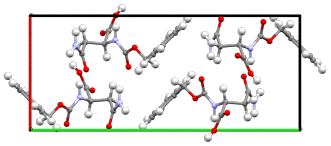
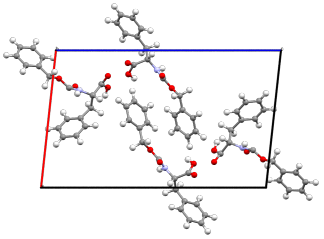
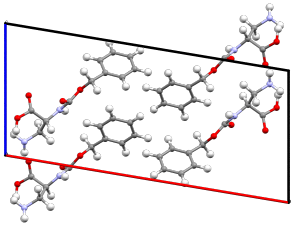
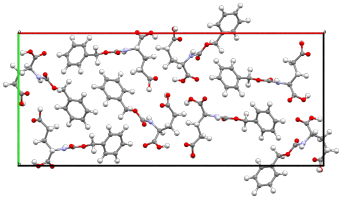
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-18	1.00	9.28Å 20.68Å 7.13Å 90° 108.8° 90°	$P2_1/a$	12.51	
1-19	0.99	–	–	–	Unindexed, no solution, stacking faults
1-20	0.99	–	–	–	Capillary could not be loaded
1-21	1.00	13.58Å 5.00Å 22.10Å 90° 96.21° 90°	$P2_1$, Z'=2	16.21	
1-22	1.00	22.39Å 4.89Å 10.31Å 90° 99.51° 90°	$P2_1/c$	12.78	
1-23	1.00	–	–	–	Unindexed, no solution, two phases
1-24	0.99	–	–	–	Unindexed, no solution, two phases
1-25	1.00	35.58Å 15.42Å 4.82Å 90° 90° 90°	$P2_12_12_1$ Z'=2	10.72	
1-26	0.99	–	–	–	Unindexed, no solution, two phases

Table 6.4 Crystal structure solutions 4

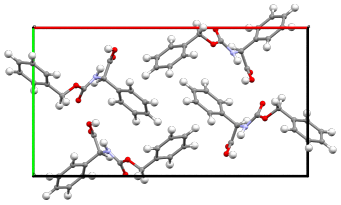
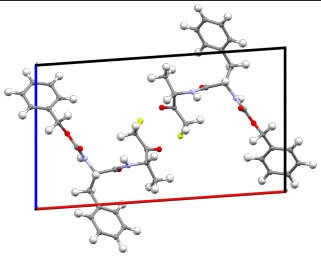
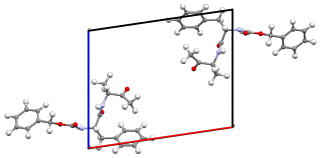
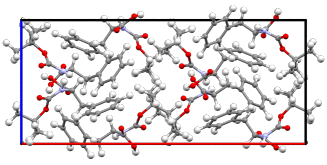
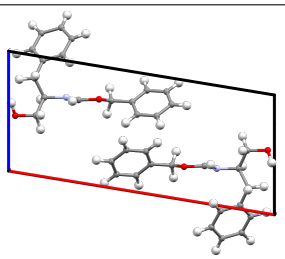
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-27	1.00	21.45Å 11.59Å 5.54Å 90° 90° 90°	$Pna2_1$	11.16	
1-28	0.99	–	–	–	Unindexed, no solution, two phases
1-29	1.00	16.43Å 4.90Å 17.05Å 90° 78.14° 90°	$P2_1$, Z'=2	–	No solution
1-30	0.92	18.83Å 4.91Å 10.80Å 90° 86.00° 90°	$P2_1$	12.43	
1-31	0.93	15.37Å 5.02Å 12.40Å 90° 81.62° 90°	$P2_1$	10.98	
1-32	1.00	24.26Å 11.40Å 10.55Å 90° 90° 90°	$P2_12_12_1$, Z'=2, BXCPAL	12.57	
1-33	0.93	18.23Å 5.16Å 8.08Å 90° 99.32° 90°	$P2_1$	13.54	

Table 6.5 Crystal structure solutions 5

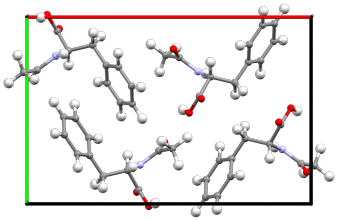
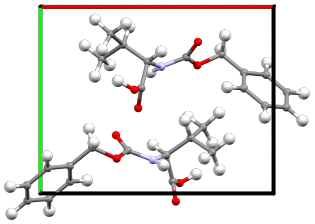
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-34	1.00	–	–	–	Unindexed, no solution, two phases
1-35	1.00	–	–	–	Unindexed, no solution, two phases
1-36	1.00	–	–	–	Unindexed, no solution
1-37	1.00	29.53Å 5.04Å 9.34Å 90° 94.28° 90°	P21, Z'=2	–	No solution
1-38	0.99	16.94Å 11.11Å 5.64Å 90° 90° 90°	$P2_12_12_1$, COQHAR	11.10	
1-39	1.00	47.51Å 4.85Å 11.05Å 90° 113.53° 90°	C2, Z'=2	–	No solution
1-40	0.97	–	–	–	Unindexed, no solution
1-41	1.00	35.53Å 15.40Å 4.82Å 90° 90° 90°	$P2_12_12_1$, Z'=2	–	No solution
1-42	1.00	11.82Å 9.47Å 5.65Å 90° 93.10° 90°	$P2_1$	10.39	

Table 6.6 Crystal structure solutions 6

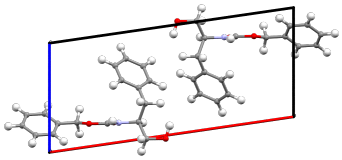
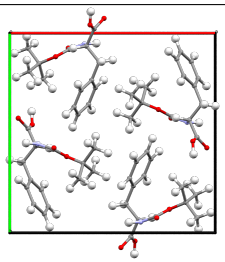
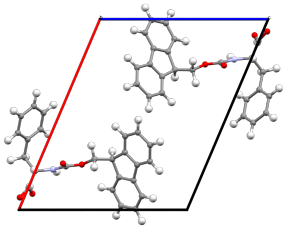
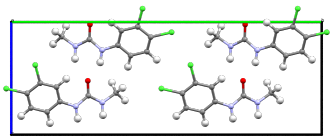
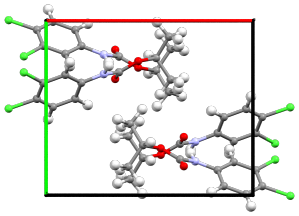
Name	Propensity	Cell parameters	Space group	Rwp	Result
1-43	0.94	18.15Å 5.15Å 8.02Å 90° 81.73° 90°	$P2_1$	13.65	
1-44	1.00	16.24Å 14.67Å 6.26Å 90° 68.6° 90°	$P2_1/a$	11.82	
1-45	0.94	16.14Å 4.91Å 13.15Å 90° 113.15° 90°	$P2_1$, OGIZOT	11.48	
1-46	0.61	–	–	–	Capillary could not be loaded
1-47	1.00	–	–	–	Unindexed, no solution
2-1	0.99	3.92Å 25.23Å 9.32Å 90° 101.42° 90°	Cc	10.90	
2-2	0.98	–	–	–	Unindexed, no solution
2-3	0.97	12.13Å 10.20Å 9.18Å 90° 92.99° 90°	$P2_1/c$	9.98	

Table 6.7 Crystal structure solutions 7

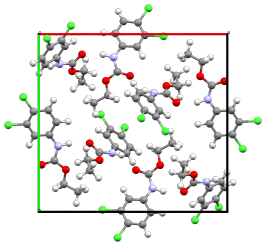
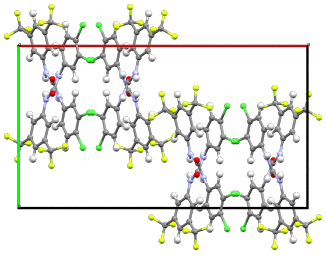
Name	Propensity	Cell parameters	Space group	Rwp	Result
2-4	0.96	17.45Å 16.38Å 7.34Å 90° 96.38° 90°	$P2_1/a$, Z'=2	15.98	
2-5	1.00	–	–	–	Unindexed, no solution
2-6	0.99	–	–	–	Capillary could not be loaded
2-7	0.76	–	–	–	Unindexed, no solution, KAMBEF
2-8	0.80	4.55Å 11.46Å 18.22Å 94.29° 85.53° 98.05°	P-1	–	No solution
2-9	0.57	7.83Å 15.29Å 12.10Å 90° 93.71° 90°	$P2_1/c$	–	No solution, asymmetric peaks
2-10	0.98	25.06Å 13.95Å 9.12Å 90° 98.07° 90°	I2/a	11.12	
2-11	0.98	24.85Å 4.62Å 5.86Å 90° 96.28° 90°	–	–	No solution

Table 6.8 Crystal structure solutions 8

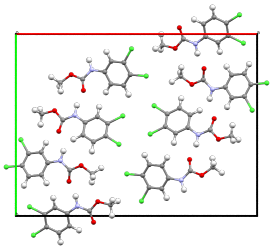
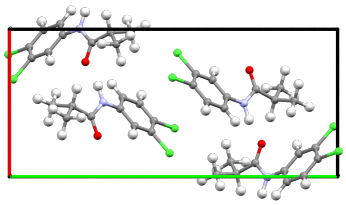
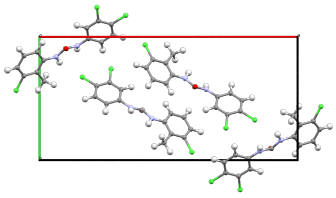
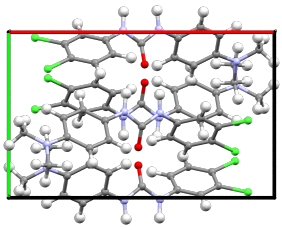
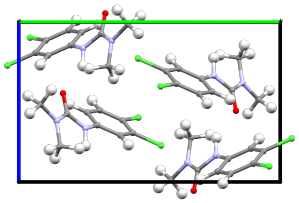
Name	Propensity	Cell parameters	Space group	Rwp	Result
2-12	0.98	25.20Å 18.64Å 3.84Å 90° 79.10° 90°	$P2_1/n$, Z'=2	10.20	
2-13	0.91	9.24Å 19.88Å 5.94Å 90° 105.09° 90°	$P2_1/n$, DCIBAN	12.59	
2-14	0.90	35.02Å 12.00Å 4.59Å 90° 133.89° 90°	$P2_1/a$	9.15	
2-15	0.78	35.04Å 12.01Å 4.60Å 90° 133.88° 90°	$P2_1/c$	11.14	
2-16	0.95	7.68Å 14.56Å 9.09Å 90° 101.70° 90°	$P2_1/c$, CLPHUR	15.76	
2-17	0.98	29.54Å 9.73Å 4.53Å 90° 90° 90°	–	–	No solution

Table 6.9 Crystal structure solutions 9

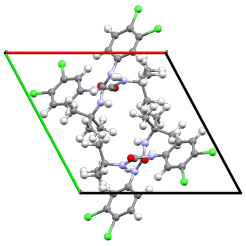
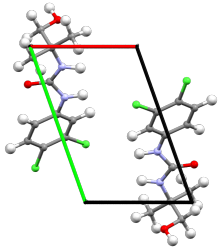
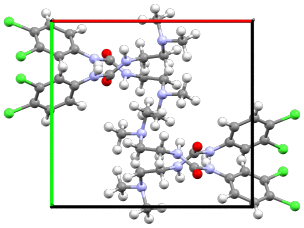
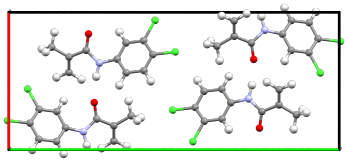
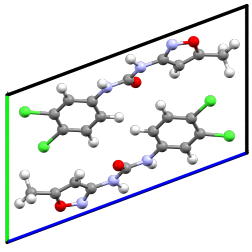
Name	Propensity	Cell parameters	Space group	Rwp	Result
2-18	0.90	13.01Å 13.23Å 9.133Å 69.85° 73.98° 58.61°	P-1, Z'=2	9.92	
2-19	0.99	–	–	–	Unindexed, no solution, YEHHUQ
2-20	0.16	6.32Å 9.87Å 11.573Å 108.07° 106.47° 66.852°	P-1	9.72	
2-21	0.15	12.80Å 11.83Å 8.73Å 90° 93.98° 90°	$P2_1/c$	17.74	
2-22	0.97	10.14Å 24.34Å 4.01Å 90° 88.46° 90°	$P2_1/a$	12.63	
2-23	0.67	4.87Å 8.63Å 15.24Å 70.57° 101.29° 87.05°	P-1	11.98	

Table 6.10 Crystal structure solutions 10

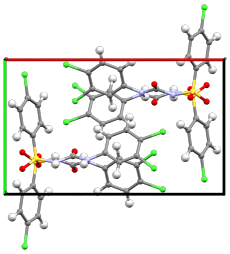
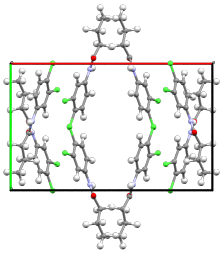
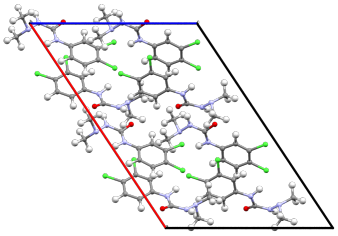
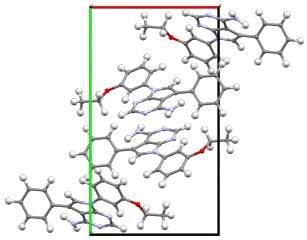
Name	Propensity	Cell parameters	Space group	Rwp	Result
2-24	0.96	16.54Å 9.82Å 9.40Å 90° 104.70° 90°	$P2_1/c$	12.80	
2-25	0.90	18.74Å 11.71Å 9.81Å 90° 94.86° 90°	C2/c	13.78	
2-26	0.82	21.71Å 8.34Å 14.83Å 90° 56.28° 90°	Cc	11.17	
3-1	0.91	10.92Å 19.30Å 7.66Å 90° 94.56° 90°	$P2_1/a$	10.21	
3-2	0.83	10.24Å 13.11Å 17.94Å 104.0° 90.55° 74.58°	P-1, Z'=2	–	No solution
3-3	0.65	–	–	–	Unindexed, no solution
3-4	0.86	–	–	–	Unindexed, no solution
3-5	0.82	–	–	–	Unindexed, no solution
3-6	0.93	–	–	–	Unindexed, no solution
3-7	0.72	–	–	–	Capillary could not be loaded
3-8	0.90	–	–	–	Unindexed, no solution

Table 6.11 Crystal structure solutions 11

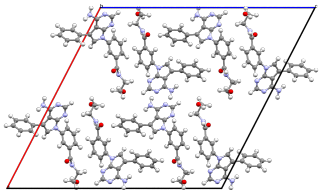
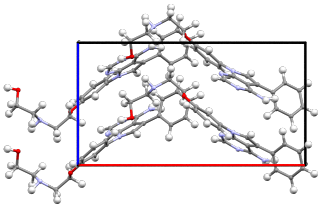
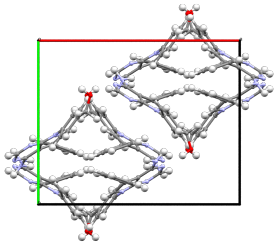
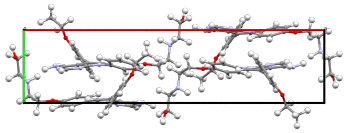
Name	Propensity	Cell parameters	Space group	Rwp	Result
3-9	0.13	28.43Å 4.98Å 30.42Å 90° 118.92° 90°	C2/c	15.08	
3-10	0.62	–	–	–	Unindexed, no solution
3-11	0.96	–	–	–	Unindexed, no solution
3-12	0.76	18.40Å 10.23Å 9.93Å 90° 90° 90°	$P2_1cn$	9.64	
3-13	0.97	17.17Å 13.64Å 13.40Å 90° 101.71° 90°	I2/a	9.30	
3-14	0.89	–	–	–	Unindexed, no solution
3-15	0.94	25.46Å 10.00Å 15.29Å 90° 86.73° 90°	$P2_1/n$, Z'=2	–	No solution
3-16	0.92	–	–	–	Unindexed, no solution
3-17	0.89	–	–	–	Unindexed, no solution
3-18	0.95	29.51Å 7.08Å 10.89Å 90° 97.70° 90°	$P2_1/a$	12.00	

Table 6.12 Crystal structure solutions 12

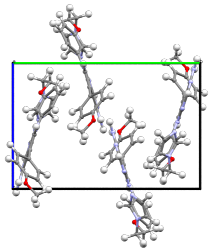
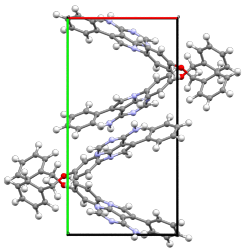
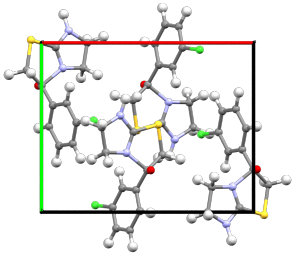
Name	Propensity	Cell parameters	Space group	Rwp	Result
3-19	0.60	11.97Å 16.14Å 11.52Å 90° 109.99° 90°	$P2_1/n$	12.56	
3-20	0.94	–	–	–	Unindexed, no solution
3-21	0.96	–	–	–	Unindexed, no solution, two phases
3-22	0.96	10.49Å 19.10Å 10.69Å 90° 67.94° 90°	$P2_1/c$	10.92	
3-23	0.58	–	–	–	Unindexed, no solution
3-24	0.47	–	–	–	Unindexed, no solution
3-25	0.89	–	–	–	Unindexed, no solution
4-1	0.99	–	–	–	Unindexed, no solution
4-2	0.98	–	–	–	Unindexed, no solution
4-3	0.87	12.62Å 9.97Å 8.84Å 90° 96.19° 90°	$P2_1/a$	22.04	
4-4	0.94	–	–	–	Unindexed, no solution
4-5	0.82	–	–	–	Unindexed, no solution, two phases
4-6	0.96	–	–	–	Unindexed, no solution

Table 6.13 Crystal structure solutions 13

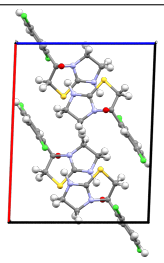
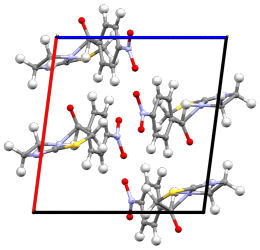
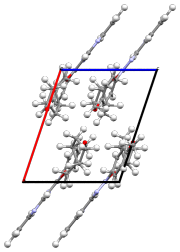
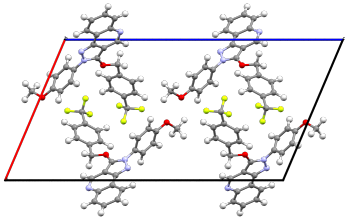
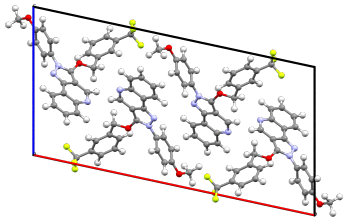
Name	Propensity	Cell parameters	Space group	Rwp	Result
4-7	0.72	14.59Å 7.29Å 11.34Å 90° 92.17° 90°	$P2_1/a$	18.02	
4-8	0.97	–	–	–	Unindexed, no solution
4-9	0.94	–	–	–	Unindexed, no solution
4-10	0.86	11.03Å 9.73Å 10.61Å 90° 97.44° 90°	$P2_1/a$	10.09	
4-11	0.93	–	–	–	Unindexed, no solution
5-1	0.89	–	–	–	Capillary could not be loaded
5-2	0.63	10.84Å 20.62Å 8.80Å 90° 108.50° 90°	$P2_1/c$	10.44	
5-3	0.72	–	–	–	Unindexed, no solution
5-4	0.42	–	–	–	Unindexed, no solution
5-5	0.66	14.40Å 5.80Å 26.16Å 90° 113.01° 90°	$P2_1/c$	11.76	
5-6	0.55	26.53Å 5.73Å 13.71Å 90° 102.14° 90°	$P2_1/a$	11.02	

Table 6.14 Crystal structure solutions 14

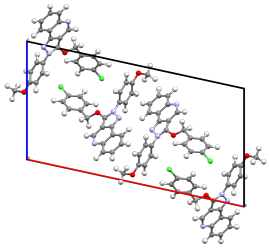
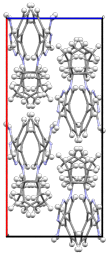
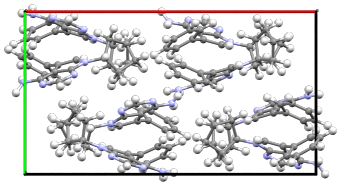
Name	Propensity	Cell parameters	Space group	Rwp	Result
5-7	0.68	–	–	–	Unindexed, no solution
5-8	0.85	–	–	–	Unindexed, no solution
5-9	0.85	–	–	–	Unindexed, no solution
5-10	0.47	–	–	–	Unindexed, no solution
5-11	0.79	25.55Å 5.65Å 13.57Å 90° 102.24° 90°	$P2_1/n$	12.88	
6-1	0.47	–	–	–	Capillary could not be loaded
6-2	0.95	24.90Å 11.88Å 10.96Å 90° 90° 90°	Pnab	17.78	
6-3	0.69	–	–	–	Unindexed, no solution
6-4	0.18	–	–	–	Unindexed, no solution
6-5	0.98	8.28Å 10.70Å 16.80Å 62.4° 92.3° 82.3°	P1	–	No solution
6-6	0.92	21.17Å 11.81Å 11.45Å 90° 90° 90°	$P2_12_12_1$, Z'=2	12.40	

Table 6.15 Crystal structure solutions 15

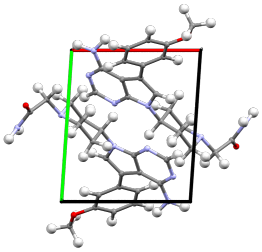
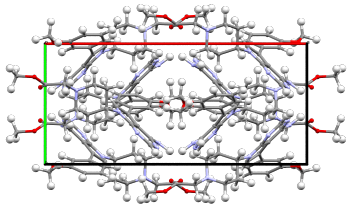
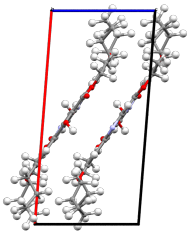
Name	Propensity	Cell parameters	Space group	Rwp	Result
6-7	0.69	9.01Å 11.96Å 14.53Å 90° 106.89° 90°	$P2_1/c$	–	No solution
6-8	0.05	7.75Å 9.24Å 13.48Å 101.32° 97.46° 92.28°	P-1	10.00	
6-9	0.08	23.79Å 9.24Å 20.80Å 90° 122.71° 90°	C2/c	11.28	
6-10	0.06	37.63Å 5.92Å 10.35Å 90° 67.22° 90°	$P2_1/n$	–	No solution
7-1	0.01	–	–	–	Unindexed, no solution
7-2	0.35	–	–	–	Unindexed, no solution
7-3	0.05	16.66Å 12.80Å 8.08Å 90° 94.48° 90°	$P2_1/c$	10.65	
7-4	0.06	–	–	–	Unindexed, no solution
7-5	0.09	–	–	–	Unindexed, no solution
7-6	0.01	–	–	–	Unindexed, no solution

Table 6.16 Crystal structure solutions 16

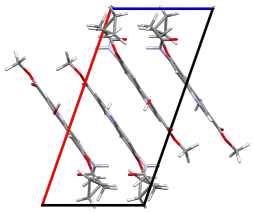
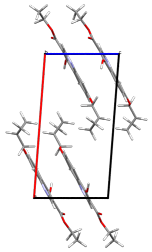
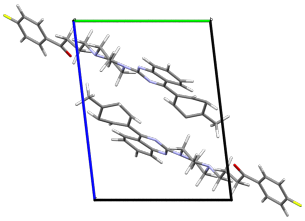
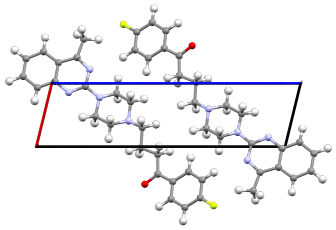
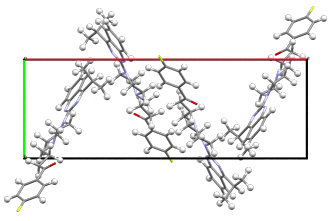
Name	Propensity	Cell parameters	Space group	Rwp	Result
7-7	0.02	16.27Å 12.65Å 7.96Å 90° 109.33° 90°	$P2_1/c$	12.91	
7-8	0.03	–	–	–	Unindexed, no solution
7-9	0.03	–	–	–	Unindexed, no solution
7-10	0.03	–	–	–	Unindexed, no solution
7-11	0.08	14.67Å 12.64Å 7.52Å 90° 94.37° 90°	$P2_1/c$	12.51	
7-12	0.50	–	–	–	Unindexed, no solution, two phases
7-13	0.04	–	–	–	Unindexed, no solution
8-1	0.58	8.10Å 10.69Å 14.71Å 83.71° 107.55° 89.42°	P-1	16.89	
8-2	0.21	5.34Å 11.21Å 20.43Å 65.11° 91.06° 65.10°	P-1	13.28	
8-3	0.84	–	–	–	Unindexed, no solution
8-4	0.35	27.09Å 9.37Å 8.64Å 90° 98.19° 90°	$P2_1/a$	13.79	

Table 6.17 Crystal structure solutions 17

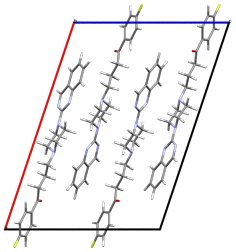
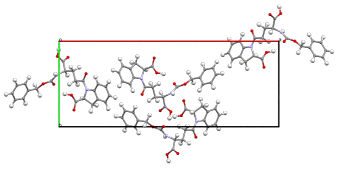
Name	Propensity	Cell parameters	Space group	Rwp	Result
8-5	0.56	–	–	–	Unindexed, no solution
8-6	0.77	–	–	–	Unindexed, no solution
8-7	0.17	21.46Å 6.03Å 15.22Å 90° 108.43° 90°	$P2_1/c$	16.53	
8-8	0.68	–	–	–	Unindexed, no solution
8-9	0.73	–	–	–	Unindexed, no solution
8-10	0.62	–	–	–	Unindexed, no solution
8-11	0.81	–	–	–	Unindexed, no solution
9-1	0.97	–	–	–	Unindexed, no solution
9-2	0.98	–	–	–	Capillary could not be loaded
9-3	0.87	–	–	–	Unindexed, no solution
9-4	0.99	–	–	–	Capillary could not be loaded
9-5	0.97	–	–	–	Unindexed, no solution
9-6	0.94	–	–	–	Unindexed, no solution
9-7	0.97	–	–	–	Capillary could not be loaded
9-8	1.00	–	–	–	Unindexed, no solution
9-9	0.98	–	–	–	Unindexed, no solution, two phases
9-10	0.97	–	–	–	Unindexed, no solution
9-11	0.99	–	–	–	Capillary could not be loaded
10-1	1.00	31.96Å 12.47Å 4.93Å 90° 90° 90°	$P2_12_12_1$	9.89	

Table 6.18 Crystal structure solutions 18

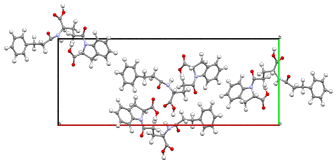
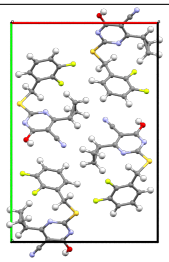
Name	Propensity	Cell parameters	Space group	Rwp	Result
10-2	0.94	–	–	–	Unindexed, no solution
10-3	1.00	–	–	–	Unindexed, no solution, two phases
10-4	0.99	–	–	–	Unindexed, no solution
10-5	0.97	13.33Å 21.23Å 5.02Å 90° 88.11° 90°	$P2_1$, Z'=2	–	No solution
10-6	0.99	26.60Å 18.06Å 4.58Å 90° 111.79° 90°	–	–	No solution
10-7	0.98	–	–	–	Unindexed, no solution
10-8	0.99	32.24Å 12.60Å 4.89Å 90° 90° 90°	$P2_12_12_1$	11.99	
10-9	0.98	–	–	–	Unindexed, no solution
10-10	0.99	–	–	–	Unindexed, no solution
11-1	0.01	13.70Å 20.36Å 4.86Å 90° 93.41° 90°	$P2_1/n$	14.03	
11-2	0.01	–	–	–	Unindexed, no solution, two phases

Table 6.19 Crystal structure solutions 19

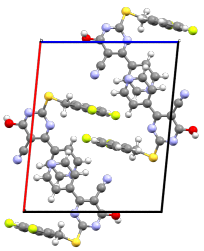
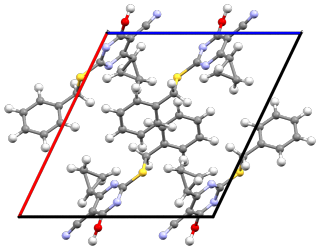
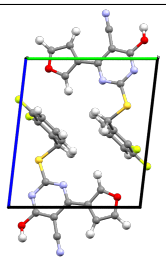
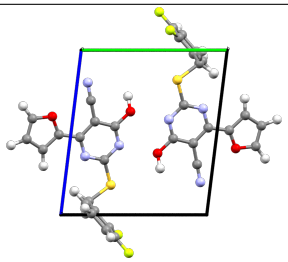
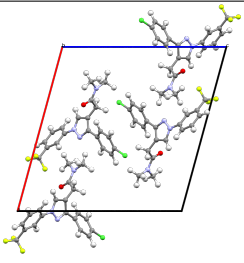
Name	Propensity	Cell parameters	Space group	Rwp	Result
11-3	0.11	–	–	–	Unindexed, no solution
11-4	0.05	15.84Å 7.38Å 12.74Å 90° 95.40° 90°	$P2_1/n$	25.26, salt	
11-5	0.24	–	–	–	Unindexed, no solution
11-6	0.24	13.75Å 8.55Å 13.08Å 90° 115.5° 90°	$P2_1/c$	16.54, salt	
11-7	0.04	–	–	–	Unindexed, no solution
11-8	0.03	7.44Å 9.25Å 10.79Å 97.79° 103.64° 84.33°	P-1	19.10, salt	
11-9	0.25	–	–	–	Unindexed, no solution
11-10	0.04	7.44Å 9.29Å 10.77Å 97.53° 77.67° 92.88°	P-1	24.22, salt	
12-1	0.40	18.28Å 5.96Å 17.60Å 90° 105.22° 90°	$P2_1/a$	10.19	

Table 6.20 Crystal structure solutions 20

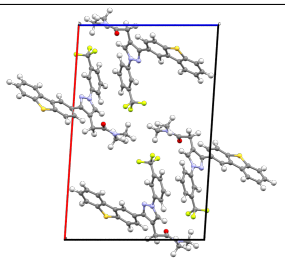
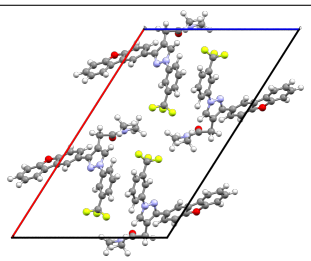
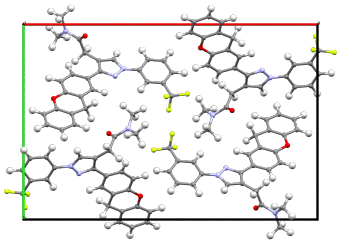
Name	Propensity	Cell parameters	Space group	Rwp	Result
12-2	0.12	8.88Å 12.19Å 18.82Å 80.08° 78.64° 73.84°	P-1, Z'=2	–	No solution
12-3	0.07	36.22Å 12.67Å 26.05Å 90° 134.02° 90°	C2/c, Z'=2	–	No solution
12-4	0.23	–	–	–	Capillary could not be loaded
12-5	0.40	23.53Å 6.07Å 15.34Å 90° 93.76° 90°	$P2_1/n$	11.46	
12-6	0.20	–	–	–	Unindexed, no solution
12-7	0.06	25.76Å 6.11Å 16.27Å 90° 122.48° 90°	C2	12.01	
12-8	0.12	25.66Å 16.70Å 5.20Å 90° 79.08° 90°	$P2_1/a$	9.64	

Table 6.21 Crystal structure solutions 21

Name	Propensity	Cell parameters	Space group	Rwp	Result
12-9	0.80	9.50Å 31.54Å 7.58Å 90° 99.06° 90°	$P2_1$, Z'=2	–	No solution
12-10	0.50	10.18Å 11.76Å 22.35Å 103.09° 103.41° 65.86°	P-1, Z'=2	–	No solution

6.3.1 Discussion

Of the 74 materials for which crystal structure solution was successful, 43 (58%) of them were very strongly predicted to be easy to crystallize by the model, with a score of over 0.8. The relatively high proportion of materials not strongly predicted to be crystallizable for which a successful structure solution was achieved shows that there is little relationship between how easy it is to grow large crystals of a material and the ease with which the structure can be solved by powder diffraction. In some cases the structure was even successfully solved for patterns which contained salt peaks, as was the case for Family 11 (albeit with a higher Rwp). As a result, powder diffraction is a useful tool for characterising structures of materials which cannot be solved by SXRD because a large enough crystal cannot be grown.

There are some exceptions to this, most notably for Family 7, for which only 3 of the 11 compounds produced a diffraction pattern of sufficient quality for the structure to be solved, so in these cases the material is so poorly crystalline that even powder diffraction methods are unsuitable for determining the crystal structure. Conversely, some materials which were strongly predicted to be crystallizable, such as

all of those in Family 9, gave a diffraction pattern from which not even the lattice parameters can be extracted. In some of these cases, the materials were loaded into the capillary with great difficulty or not at all, and the poor packing of the material into the capillary may have contributed to the poor diffraction data.

Figure 6.3 and Figure 6.4 show the distributions of crystallite size and strain respectively with crystallization probability from the predictive model for those materials with a successful crystal structure solution. They show that there is no correlation between the microstructure of the material and the predicted ease of crystallization, with correlation coefficients of 0.12 and -0.05 with the crystallite size and strain respectively. This evidence suggests that there is little relationship between the factors that allow formation of a crystalline powder with very small crystallites, and those that aid formation of a large single crystal suitable for SXRD. The thermodynamic stability and imperfections of the crystal structure would not appear to be the factor which determines the extent to which the crystal grows, and in fact some materials which are strongly predicted to be hard to crystallize have high crystallite size and low strain.

In these cases, perhaps the material forms crystallites too easily, leading to a crystalline powder with few imperfections, which crashes out of solution quickly before crystal growth can occur to form a large single crystal. However, it also seems to support the conclusions from Chapter 3, that the kinetic effects of conformations in solution have a more direct impact on the ability to form a large crystal than the thermodynamic stability of the crystal. Even in cases where tertiary amide groups, which were found to be detrimental to formation of large single crystals, are present in the molecule, these groups appear to have little effect on the microstructure of the material, as is the case for Family 12.

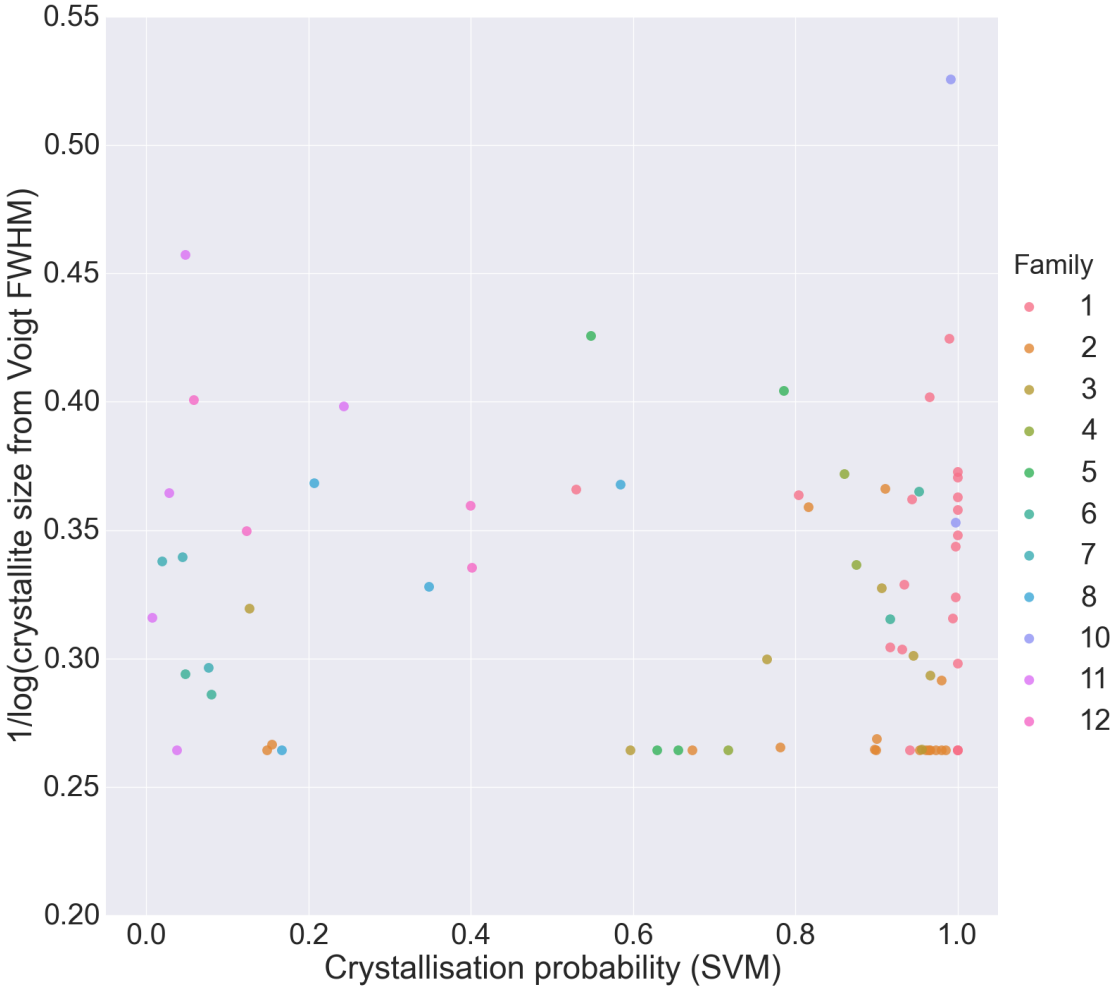


Figure 6.3 Log of crystallite size (Å) by Rietveld analysis against crystallization probability for successful crystal structure solutions, colour-coded by family.

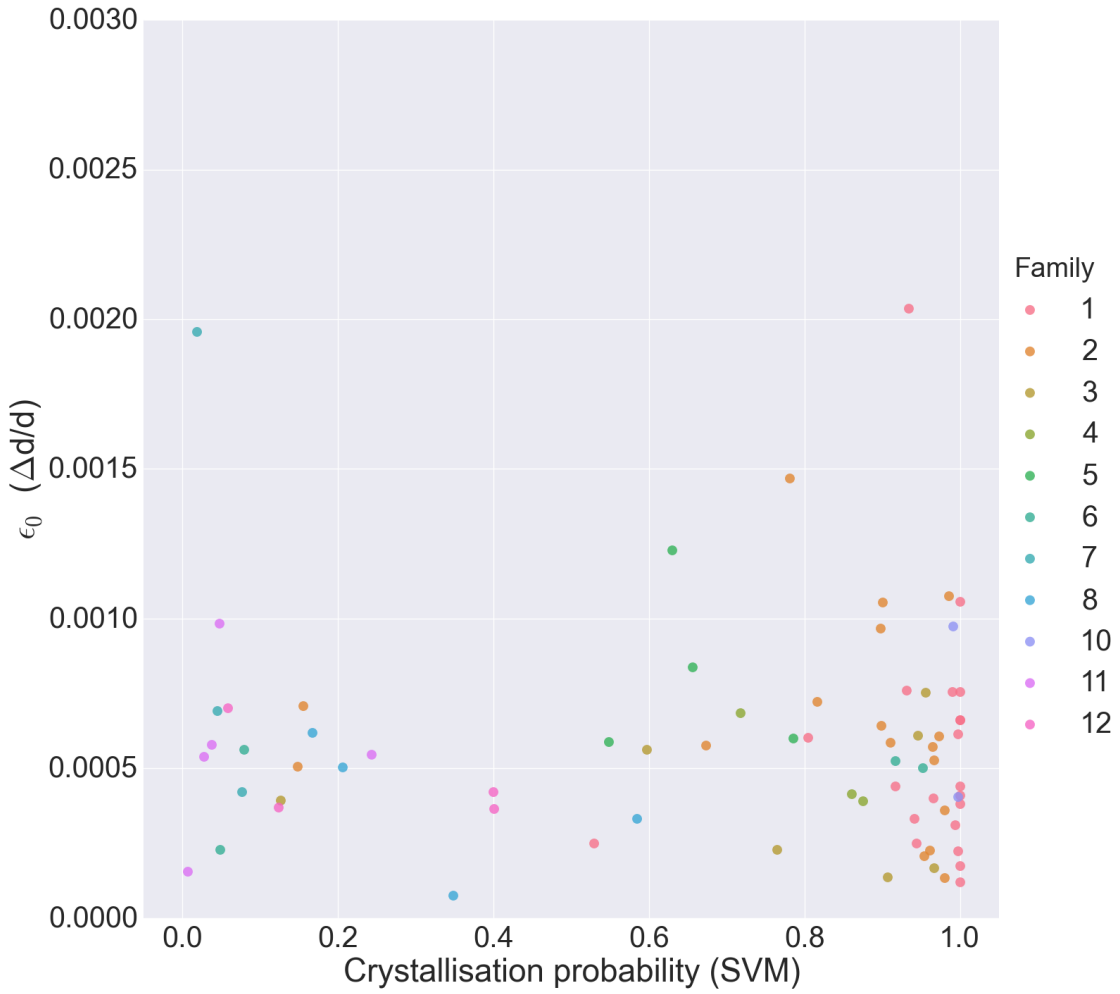


Figure 6.4 Crystallite strain by Rietveld analysis against crystallization probability for successful crystal structure solutions, colour-coded by family.

There are two key points in the structure solution process at which failure is most likely to occur; the indexing stage, and the simulated annealing structure solution stage.

Table 6.22 gives the details of those materials for which crystal structure solution failed at the simulated annealing stage, including calculations of the ratio of unit cell volume to molecular volume, calculated using the standard volume of 17 Å^3 per atom. These 22 structures were successfully indexed to obtain lattice parameters with subsequent Pawley analysis being used to verify the lattice parameters.

Table 6.22 Details of materials with lattice parameters successfully determined but no crystal structure solution. *poor pattern **ambiguous space group ***asymmetric peaks

Name	Unit cell volume (Å^3)	Mol. volume (Å^3)	Ratio	Difference to nearest integer	Z'>1	Rwp	RBC
12_2	1901	476	3.99	0.01	Y	12.84	4
10_6	2033	510	3.99	0.01	N*	9.16	7
10_5	1420	357	3.98	0.02	Y	12.04	5
1_1	1486	374	3.97	0.03	Y	11.92	6
2_17	1302	323	4.03	0.03	N**	9.27	2
2_11	669	340	1.97	0.03	N**	10.87	2
1_29	1344	340	3.95	0.05	Y	10.49	7
6_5	1295	629	2.06	0.06	N**	10.11	6
1_37	1388	340	4.08	0.08	Y	12.6	7
12_3	8487	527	16.10	0.10	Y	9.27	4
2_9	1447	374	3.87	0.13	N	18.64***	3
6_10	2120	510	4.16	0.16	N	8.25	3
12_10	2350	561	4.19	0.19	Y	12.07	4
6_7	1499	357	4.20	0.20	N	12.5	3
1_41	2638	340	7.76	0.24	Y	12.55	7
12_9	2244	527	4.26	0.26	Y	9.58	4
3_2	2250	527	4.27	0.27	Y	9.51	9
1_2	1397	323	4.33	0.33	N	12.38	6
3_15	3887	510	7.62	0.38	Y	10.03	5
1_8	2310	527	4.38	0.38	Y	9.00	11
2_8	936	374	2.50	0.50	N	9.45	3
1_39	2334	272	8.58	0.58	Y	12.03	4

9 of the materials were found to have a unit cell with a volume which was inconsistent with an integer multiple of the molecular volume calculated from the 17 Å rule, suggesting that they had crystallized as a solvate. Such structures are difficult to solve because they require either knowledge of the solvent of crystallization, which can be input to the simulated annealing process, or use of an alternative method such as the maximum likelihood method[212] which is useful when the structural model being

optimized is not a complete description of the crystal structure under study. This adds an extra level of complexity to the process.

The most common cause of failure at the simulated annealing stage, however, is in cases where there is more than one molecule in the asymmetric unit. This also adds extra complexity to the problem, since not only do the translations, rotations and torsions of each independent molecule need to be optimised, but also their positions relative to each other. Of the 22 materials which were successfully indexed, 13 of them failed at the simulated annealing stage due to having two or molecules in the asymmetric unit. 4 of these also had a cell volume consistent with the presence of a solvate, with the remaining 9 giving a cell volume consistent with a pure material.

It is worth noting that 7 of the 74 successful crystal structure solutions (9%) contained more than one molecule in the asymmetric unit. This is a much smaller proportion than the roughly 50% of molecules which failed the simulated annealing process, but is much closer to the overall prevalence of crystal structures with Z'>1 in the CSD, which is around 10%.[213] This highlights the extra complexity involved in solving more complex structures by powder diffraction, which simply require more time to achieve. In practice, increasing the length and number of simulated annealing runs facilitates the discovery of a solution for these materials, but this is computationally more expensive and ideally requires parallel computing to be achieved within a sensible timeframe. For particularly complex problems, cloud computing can be a helpful tool to solve the structure,[214] but this is costly. In addition, less certainty can be attached to the final solution for such materials, since there are a great many more possibilities for arranging several molecules in an asymmetric unit, in contrast to the situation with Z'=1, when often the same structure will be discovered by several, if not all, of the simulated annealing runs. DFT validation of the structure or neutron diffraction studies could be used to increase the confidence in the final structure solution.

The remaining three materials had potential unit cell volumes consistent with an

integer multiple of molecules, suggesting that they had not crystallized as a solvate. In these cases the space group was ambiguous, either because there were several possible options (as for 2-17), the pattern was too poor (10-6) or the cell size was ambiguous (2-11), which prevented structure solution.

Figure 6.5 and Figure 6.6 show the distributions of crystallite size and strain respectively with crystallization probability from the predictive model for those materials with no successful crystal structure solution, but successful lattice parameter determination. The distributions are similar to those for the materials with successful structure solutions, and again there appears to be no correlation between the ease of crystallization and the crystallite size or strain, with correlation coefficients of 0.16 and 0.09 respectively.

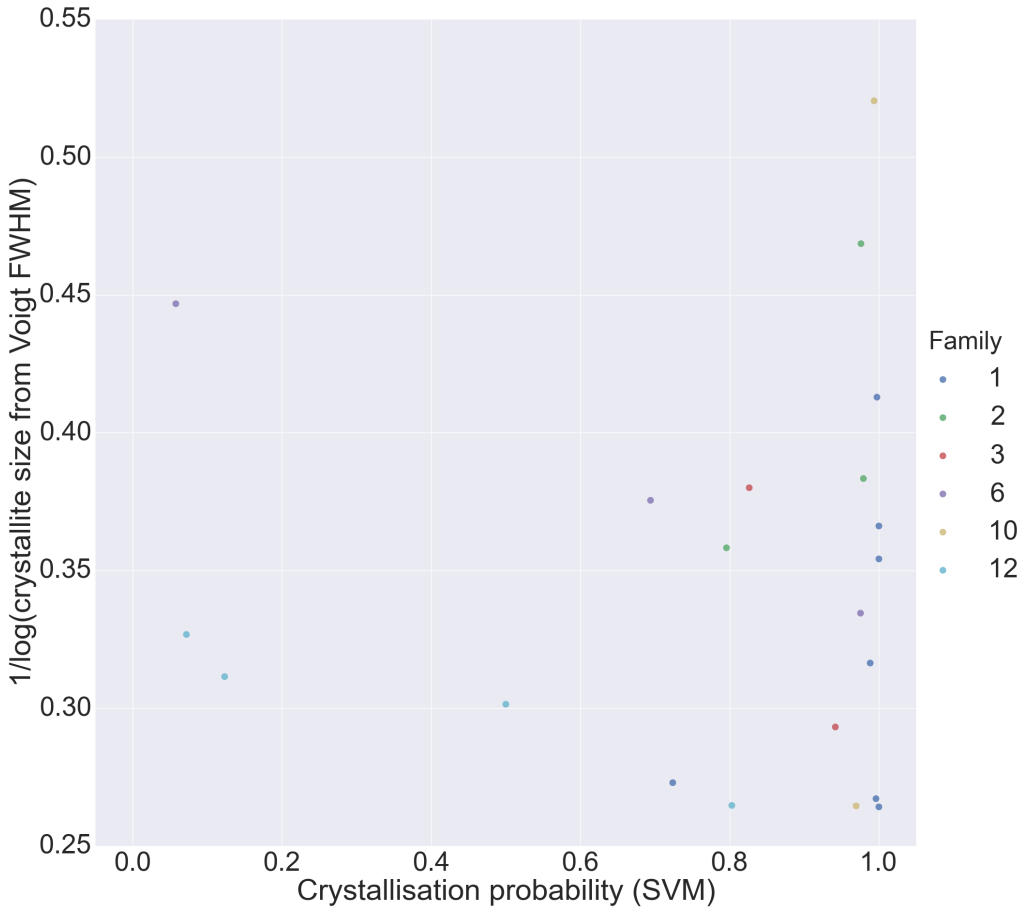


Figure 6.5 Log of crystallite size by Pawley analysis against crystallization probability for unsuccessful crystal structure solutions, colour-coded by family.

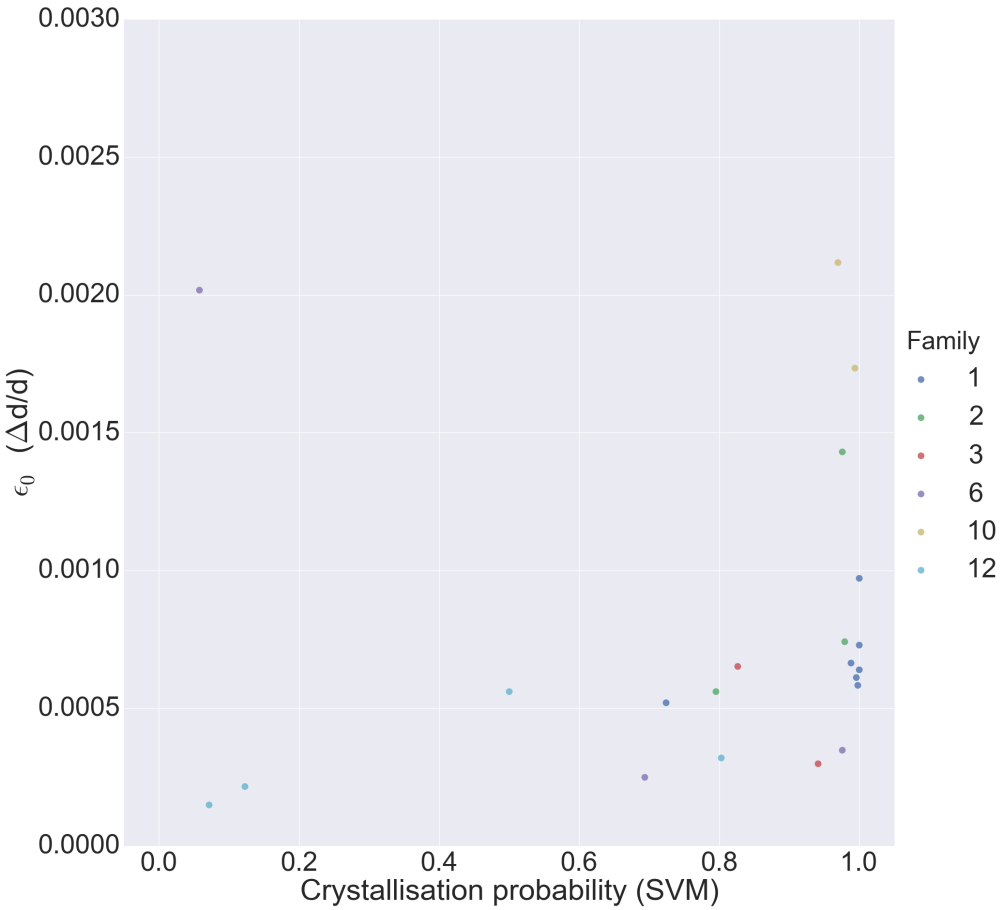


Figure 6.6 Crystallite strain by Pawley analysis against crystallization probability for unsuccessful crystal structure solutions, colour-coded by family.

87 structures could not be indexed at all. There are two main reasons why this was the case; either there were multiple phases present in the powder, or the material exhibited significant structural imperfections such as stacking faults.

Figure 6.7 shows the diffraction pattern of a material which could not be successfully indexed because there was more than one phase in the powder. The indexing output from TOPAS identified that the most likely potential phase had a space group of $P2_1/c$ and a volume of 1312 Å^3, with lattice parameters which would give the peaks identified in red. This seems to fit reasonably well with many of the peaks in the pattern. However, of the 20 peaks which were used by the indexing algorithm, 10 of them remain unindexed by this unit cell.

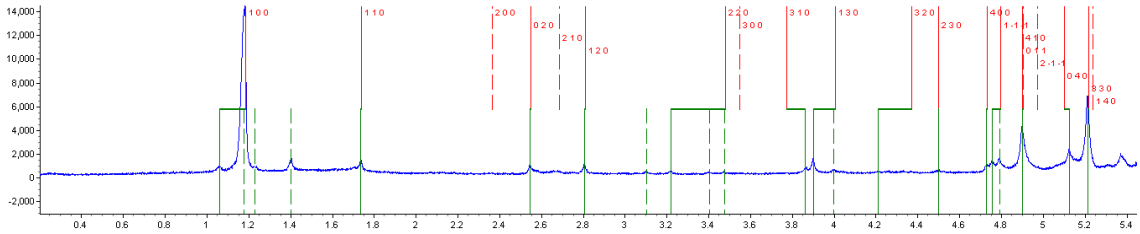


Figure 6.7 Diffraction pattern of compound 24 from family 1, which appears to contain two phases.

Such a large number of unindexed peaks suggests that the remaining peaks belong to a different phase. This could either be another pure polymorph of the same crystal, a solvated polymorph, or an impurity. In practice, the remaining peaks can be input to another indexing run to identify the second phase, but phase identification was unsuccessful in this case. The presence of multiple phases greatly increases the complexity of the indexing process, since the possibility for peak overlap is increased, meaning that some peaks may be missed. The assignment of peaks to a particular phase can be aided if the microstructural broadening is significantly different for the two phases. However, the phases may be present in different proportions, meaning that it may not be possible to identify many of the peaks in the second phase, especially if they overlap with peaks from the dominant phase. It would require a much more detailed analysis and significant effort to succeed with indexing in such cases.

An extreme example of structural imperfections preventing successful indexing is presented in Figure 6.8. The powder diffraction pattern shows a "saw-toothed" peak at a 2θ value of approximately 4.7°, corresponding to a d-spacing of 4.94 Å, a peak shape which was first identified by Warren as being indicative of stacking faults in the structure.[215] This structure consists of equidistant layers of parallel molecules which have random translations parallel to the layer, giving rise to a long tail for some peaks in the diffraction pattern. This tail obscures the remainder of the peaks, making it difficult to select the peaks to use as input for the indexing algorithm.

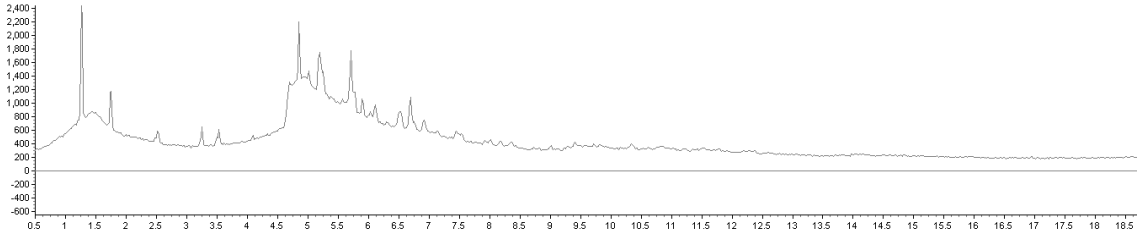


Figure 6.8 Diffraction pattern of compound 19 from family 1, which exhibits stacking faults.

A more common microstructural reason for failed indexing is small crystallite size, leading to large peak broadening throughout the sample as a result of the $\cos\theta$ dependence of the line-broadening from Equation 1.20. Such an example is shown in Figure 6.9, and a simple estimate of the crystallite size can be obtained by using Equation 1.20 with one of the low angle peaks. For the peak at 1.5°, the FWHM is approximately 0.4°, giving a particle size of approximately 50 Å. The log of this is 1.7, which makes it smaller than any of the materials that have had their crystallite size extracted by Pawley or Rietveld analysis shown in Figures 6.3 and 6.5. Since the approximate molecular volume of this molecule is 357 $Å^3$, the cube root of which is 7 Å, approximately 7 of the molecules will fit along a crystallite side length of 50 Å and so the crystallite roughly consists of $7x7x7$ molecules, which is indeed very small. The broad peaks make it difficult to identify the peak position, and there are insufficient distinguishable peaks to be able to select 20 peaks for an indexing run.

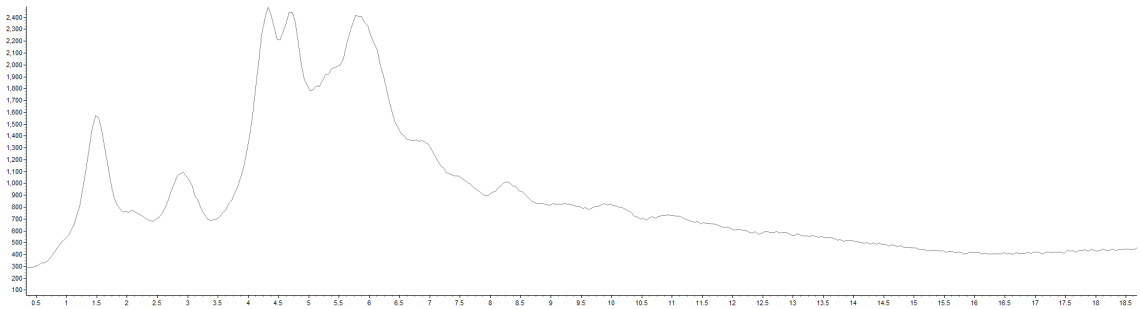


Figure 6.9 Diffraction pattern of compound 5 from family 10, which exhibits a small crystallite size.

By contrast, the pattern shown in Figure 6.10 shows sharp peaks at low 2θ, but the

peaks quickly become much broader as 2θ increases. This is indicative of relatively large crystallite size, giving little broadening at low angle, but also large strain, as a result of the $1/\tan\theta$ dependence of the line-broadening on the crystallite size from Equation 1.21. Taking a low angle peak to calculate the crystallite size, the peak at 0.7° has a FWHM of approximately 0.07°, giving a particle size of around 300Å. This is still relatively small, but is of the same order of magnitude as some of the successful structure solutions, suggesting that such a particle size alone is not necessarily sufficient to prevent structure solution in this case. However, taking a higher angle peak to calculate the strain from Equation 1.21, the peak at 5.6° has a FWHM of 0.22°, giving an ϵ_0 value of 0.02. This is an order of magnitude greater than the strain in structures that have been solved as shown in Figures 6.4 and 6.6. The large strain broadens the peaks quickly as 2θ increases, so again it is difficult to identify the peak position unless the peaks are at low angle, and there are insufficient distinguishable peaks to be able to select 20 peaks for an indexing run.

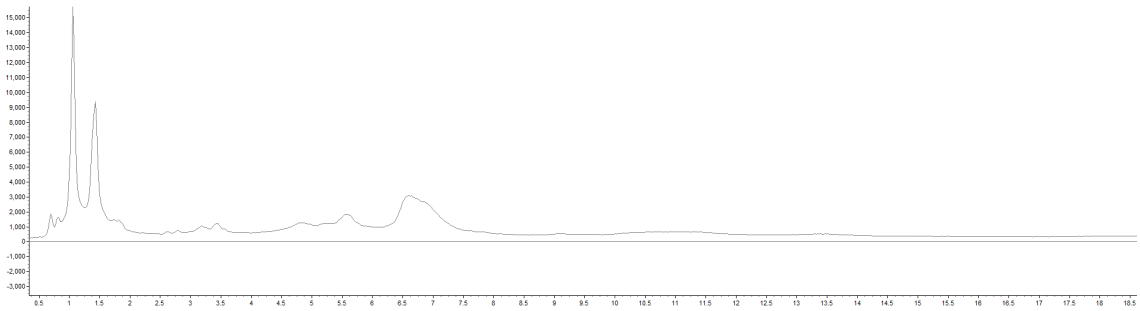


Figure 6.10 Diffraction pattern of compound 5 from family 7, which exhibits significant strain in the structure.

The lack of correlation between the crystallization propensity and the strain in the crystal structure for these materials indicates that the factors affecting these two properties do not appear to be related. This is reinforced by attempting to fit a regression model to the data. The outcome is usually a situation where the coefficients of the model are zero for each descriptor, indicating that attempting to use the descriptors to predict the strain of a material is no better than predicting a single value of strain for every material. The descriptors which are effective at distinguishing the

ease of crystallization of materials are therefore ineffective at predicting crystallite strain, so either different descriptors must be the cause of this strain, or there is no information in the 2-dimensional representation of the molecule that accounts for the strain in the structure.

Finally, energy calculations were attempted using the successful crystal structure determinations, to identify any potential relationship between the predicted crystallization propensity and the thermodynamic stability of the crystal structure. The structure obtained from the powder diffraction structure solution was optimised using CASTEP if necessary before calculation of the lattice energy by the AA-CLP method, as explained in Section 2.6.

Of the 74 structures, 5 could not have their energies calculated by the AA-CLP method due to the intermolecular interactions being deemed to be too strong by the program even after geometry optimisation, and so were excluded from the analysis. 22 of the remainder required geometry optimisation by CASTEP before successful lattice energy calculations could be attempted, with the remainder being successful using the structure from powder diffraction.

The sparse nature of the data due to the relatively small number of points in the set lends itself to kernel density estimation, which was used to estimate the probability density function of the lattice energy for each class of molecules, essentially smoothing the data. Figure 6.11 shows that there is significant overlap between the two lattice energy distributions, but the molecules which are predicted to be harder to crystallize tend to have a slightly more negative lattice energy than the ones which are predicted to be easy to crystallize. The maximum density for the less crystallizable molecules occurs at around -210 kcal mol^{-1}, compared to around -130 kcal mol^{-1} for the more crystallizable molecules. The least stable molecule with high predicted crystallizability has a lattice energy of -103.2 kcal mol^{-1}, compared to -138.0 kcal mol^{-1} for the molecules with low predicted crystallizability. However, the significant overlap of the two distributions means that there is little predictive capability on a molecule

by molecule basis, as shown by the fact that a handful of crystallizable molecules actually have stronger lattice energies than the less crystallizable molecules.

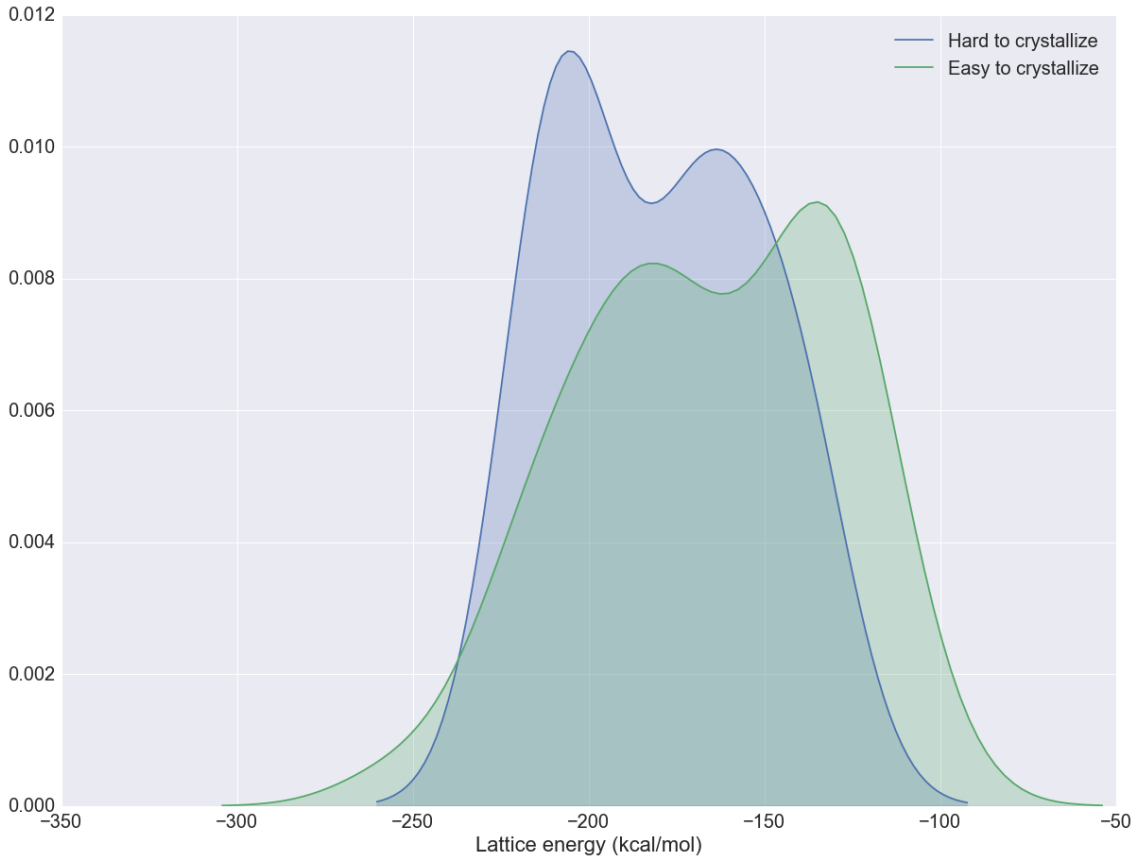


Figure 6.11 Lattice energy distributions for molecules which are predicted to be easy to crystallize (green) and hard to crystallize (blue).

In general, the structures of the molecules which are predicted to be hard to crystallize appear to be slightly more stable than those of the molecules which are predicted to be easy to crystallize. One possible explanation for this is that once the molecules find the correct conformation and orientation to attach to the nucleus of the crystal, they attach strongly and so the crystal grows quickly. This fast crystal growth results in small crystals that crash out of solution quickly as a powder, leading to a narrower metastable zone (Section 1.1.2), and increasing the difficulty of maintaining the crystal in the metastable region of the solubility diagram. For materials with a less stable crystal structure, crystal attachment is weaker and so growth occurs more slowly, leading to a wider metastable zone and greater ease of crystallization.

Materials with a high lattice energy may still be easy to crystallize if there are kinetic reasons for a large metastable zone width, such as low flexibility.

6.3.2 Conclusions

Overall, there appears to be little correlation between the powder diffraction pattern and the ease of crystallization, which gives hope for solving the structures of materials which cannot grow into a high quality single crystal. Lattice parameters were successfully determined for approximately half of the materials of varying predicted crystallization propensity, including a significant number which were predicted not to be crystallizable, and structure solutions were obtained for the majority of these. Most failures were due to the existence of solvent molecules or multiple molecules in the asymmetric unit, but even in these cases structure solution should be possible given greater time and computational resources. Although there are many cases where indexing and therefore structure solution is unachievable due to the extent of the peak broadening, a significant number of unindexed materials where two phases are present in the powder have the potential for a successful structure solution to be obtained if sufficient care and effort is expended on identifying the peaks which belong to each phase. Consequently, a non-crystallizable material should not be considered a hopeless case in terms of structure solution. Instead, for such a case, efforts should be focussed on obtaining a powder diffraction pattern to solve the structure, rather than using time and resources attempting to grow a single crystal.

Molecules which are predicted to be easier to crystallize have in general been found to have slightly less stable crystal structures than those which are hard to crystallize, a surprising discovery that could be attributed to the speed with which such materials crystallize as a result of the strong intermolecular interactions in the crystal.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This work shows that machine learning algorithms can be used to train predictive models for assessing the ease of crystallization of molecular materials using 2D molecular descriptors with an error rate of approximately 10% on drug-like materials. Extending the training dataset to include molecules outside the drug-like filter extends the domain of applicability of the model and improves the overall predictive accuracy, without compromising the predictive capability on the drug-like dataset. A similar approach can be used to predict the likelihood of co-crystal formation with significant enrichment in a ranked list of coformer-API pairs.

The best two variable classifier of crystallization propensity is obtained by using SMR VSA3 (a descriptor encoding information about the environment of the nitrogen atoms in the molecule) with ${}^{1}\kappa$ (a shape descriptor correlated with the size of the molecule). Large molecules with hindered nitrogen atoms are hard to crystallize, although smaller molecules with similar nitrogen atom environments can still crystallize due to reduced hindrance and great ease of attaining the correct orientation.

Flexibility is also a key feature for predicting crystallization, as encoded by rotatable bond count – more flexible molecules are harder to crystallize. This flexibility information is captured more directly by a new 3-dimensional descriptor, nConf_{20}, calculated by generating and counting conformers for each molecule. When nConf_{20} is used to make a single variable classifier of molecules observed and not observed to crystallize, the mean predictive accuracy from cross-validation is 84.9%, 7 percentage points better than any other single variable, so nConf_{20} captures more information than any other single 2D descriptor about the likelihood of a molecule being observed

to crystallize.

The descriptor performance is reduced when conformers are generated using torsion angle distributions from the CSD, as accidental bias is introduced by including information which has been derived from the crystalline set of molecules, which is not necessarily applicable to the non-crystalline set.

The blind test provided a method of validating the model in an unbiased fashion, identifying materials incorrectly labelled as non-crystalline that do crystallize, and giving an accuracy of 84%. This reinforced the computational validation, which was carried out using an update of the CSD and which gave an accuracy of 88%. Controlled cooling experiments show that the model also provides information about cooling rates required for high quality crystals to grow, and the optimal cooling rate for the chosen set of molecules was found to be 55°C to 30°C over a three-day period. Slower cooling rates are required for molecules which are harder to crystallize, and improve crystal quality in most cases.

There appears to be no relationship between the ease of crystallization and the microstructure of the material, and powder diffraction has been shown to be a crucial method for determining the structure of materials which are difficult to crystallize as well as those which are easy to crystallize. The main causes of failure of the structure solution process were an inability to index the powder pattern due to the presence of multiple phases, and the difficulty of obtaining a successful solution by simulated annealing in cases where there was more than one molecule in the asymmetric unit. Lattice energy calculations suggest that materials with a more stable structure are more difficult to crystallize, which has been rationalised in terms of the effect on the metastable zone width and the speed of crystallization.

7.2 Future work

This work focuses on the prediction of crystallization propensity for organic molecular materials, but with the inclusion of the correct descriptors, this approach could

potentially be applied to inorganic materials such as metal-organic frameworks. The success of the method for co-crystal prediction opens the door to other multi–component crystallization predictions, either by extending the test set of data to allow prediction of other co-crystal systems, or by using the information contained within the CSD to identify molecules which readily form solvates.

The new descriptor could be developed further by using different force-fields (such as the Universal Force Field) and other conformer generators like BALLOON, FROG2 or CONFAB. Solvent effects could be taken into account within the force field in order to produce more realistic solution-phase conformations. The distribution of conformer energies within the given energy range for a particular molecule could be taken into account to distinguish between molecules with identical $nConf_{20}$ values.

The blind test and controlled cooling experiments could be extended to include a wider set of materials with varying chemistry and a broader range of crystallization conditions, to further validate the model. The effect of the metastable zone width on the crystallization could be assessed by measuring solubility curves.

There are still improvements to be made in the powder diffraction structure solution process. Developing routines to improve the indexing process in cases where there are two or more phases present would be invaluable. For Z'>1 structures, more simulated annealing runs could be used with a greater number of moves to allow determination of the correct structure, but other factors could be considered such as improved methods for preventing unreasonable overlap of the separate molecules in the cell.

The developments in machine learning and the increasing amount of available data which can be used to train such algorithms offer great opportunities for not only predicting the properties of materials, but also guiding the discovery of new materials, using currently available information to generate new knowledge.

References

[1] J. Hulliger, *Angew. Chem. Int. Ed. Engl.* **1994**, *33*, 143–162.

[2] D. J. Watkin, *Crystallogr. Rev.* **2010**, *16*, 197–230.

[3] M. E. Bunnage, *Nat. Chem. Biol.* **2011**, *7*, 335–339.

[4] F. Pammolli, L. Magazzini, M. Riccaboni, *Nat. Rev. Drug Discovery* **2011**, *10*, 428–438.

[5] M. J. Waring, J. Arrowsmith, A. R. Leach, P. D. Leeson, S. Mandrell, R. M. Owen, G. Pairaudeau, W. D. Pennie, S. D. Pickett, J. Wang, O. Wallace, A. Weir, *Nat. Rev. Drug Discovery* **2015**, *14*, 475–486.

[6] B. C. Hancock, *J. Pharm. Sci.* **2017**, *106*, 28–30.

[7] M. Descamps, *Adv. Drug Delivery Rev.* **2016**, *100*, 1–2.

[8] J. K. R. Weber, C. J. Benmore, K. J. Suthar, A. J. Tamalonis, O. L. G. Alderman, S. Sendelbach, V. Kondev, J. Yarger, C. A. Rey, S. R. Byrn, *Biochim. Biophys. Acta Gen. Subj.* **2017**, *1861*, 3686–3692.

[9] R. Laitinen, K. Löbmann, C. J. Strachan, H. Grohganz, T. Rades, *Int. J. Pharm.* **2013**, *453*, 65–79.

[10] N. Stieger, W. Lienenberg in *Crystallization - Science and Technology*, InTech, **2012**, Chapter 7, pp. 183–204.

[11] J. Chen, B. Sarma, J. M. B. Evans, A. S. Myerson, *Cryst. Growth Des.* **2011**, *11*, 887–895.

[12] N. Variankaval, A. S. Cote, M. F. Doherty, *AIChE J.* **2008**, *54*, 1682–1688.

[13] S. L. Morissette, Ö. Almarsson, M. L. Peterson, J. F. Remenar, M. J. Read, A. V. Lemmo, S. Ellis, M. J. Cima, C. R. Gardner, *Adv. Drug Delivery Rev.* **2004**, *56*, 275–300.

[14] B. Y. Shekunov, P. York, *J. Cryst. Growth* **2000**, *211*, 122–136.

[15] S. Datta, D. J. W. Grant, *Nat. Rev. Drug Discovery* **2004**, *3*, 42–57.

[16] J. A. Dirksen, T. A. Ring, *Chem. Eng. Sci.* **1991**, *46*, 2389–2427.

[17] D. Kashchiev, G. M. van Rosmalen, *Cryst. Res. Technol.* **2003**, *38*, 555–574.

[18] S. A. Kulkarni, S. S. Kadam, H. Meekes, A. I. Stankiewicz, J. H. ter Horst, *Cryst. Growth Des.* **2013**, *13*, 2435–2440.

[19] G. Coquerel, *Chem. Soc. Rev.* **2014**, *43*, 2286–2300.

[20] R.-Q. Song, H. Cölfen, *CrystEngComm* **2011**, *13*, 1249–1276.

[21] D. Erdemir, A. Y. Lee, A. S. Myerson, *Acc. Chem. Res.* **2009**, *42*, 621–629.

[22] P. G. Vekilov, *Cryst. Growth Des.* **2010**, *10*, 5007–5019.

[23] D. Gebauer, M. Kellermeier, J. D. Gale, L. Bergström, H. Cölfen, *Chem. Soc. Rev.* **2014**, *43*, 2348–2371.

[24] P. E. Bonnett, K. J. Carpenter, S. Dawson, R. J. Davey, *ChemComm* **2003**, *120*, 698–699.

[25] R. J. Davey, S. L. M. Schroeder, J. H. Ter Horst, *Angew. Chem. Int. Ed.* **2013**, *52*, 2167–2179.

[26] L. Yu, S. M. Reutzel-Edens, C. A. Mitchell, *Org. Process Res. Dev.* **2000**, *4*, 396–402.

[27] L. Derdour, D. Skliar, *Cryst. Growth Des.* **2012**, *12*, 5180–5187.

[28] L. Derdour, C. Sivakumar, D. Skliar, S. K. Pack, C. J. Lai, J. P. Vernille, S. Kiang, *Cryst. Growth Des.* **2012**, *12*, 5188–5196.

[29] J. Ulrich, C. Strege, *J. Cryst. Growth* **2002**, *237-239*, 2130–2135.

[30] G. Hofmann in Die Praxis de Kristallisation, Essen, **1991**.

[31] S. S. Kadam, S. A. Kulkarni, R. Coloma Ribera, A. I. Stankiewicz, J. H. ter Horst, H. J. Kramer, *Chem. Eng. Sci.* **2012**, *72*, 10–19.

[32] B. Spingler, S. Schnidrig, T. Todorova, F. Wild, *CrystEngComm* **2012**, *14*, 751–757.

[33] C. B. Aakeröy, D. J. Salmon, *CrystEngComm* **2005**, *7*, 439–448.

[34] G. R. Desiraju, *J. Am. Chem. Soc.* **2013**, *135*, 9952–9967.

[35] T. Grecu, C. A. Hunter, E. J. Gardiner, J. F. McCabe, *Cryst. Growth Des.* **2014**, *14*, 165–171.

[36] N. Issa, P. G. Karamertzanis, A. V. Kazantsev, G. W. A. Welch, S. L. Price, *Cryst. Growth Des.* **2009**, *9*, 442–453.

[37] P. G. Karamertzanis, A. V. Kazantsev, N. Issa, W. A. Gareth, C. S. Adjiman, C. C. Pantelides, S. L. Price, *J. Chem. Theory Comput.* **2009**, *5*, 1432–1448.

[38] G. He, P. S. Chow, R. B. H. Tan, *Cryst. Growth Des.* **2009**, *9*, 4529–4532.

[39] Y. A. Abramov, C. Loschen, A. Klamt, *Journal of Pharmaceutical Sciences* **2012**, *101*, 3687–3697.

[40] C. A. Hunter, *Angew. Chem. Int. Ed.* **2004**, *43*, 5310–5324.

[41] M. C. Etter, *J. Phys. Chem.* **1991**, *95*, 4601–4610.

[42] T. Grecu, H. Adams, C. A. Hunter, J. F. McCabe, A. Portell, R. Prohens, *Cryst. Growth Des.* **2014**, *14*, 1749–1755.

[43] P. A. Wood, N. Feeder, M. Furlow, P. T. A. Galek, C. R. Groom, E. Pidcock, *CrystEngComm* **2014**, *16*, 5839–5848.

[44] L. Fábián, *Cryst. Growth Des.* **2009**, *9*, 1436–1443.

[45] A. Gavezzotti, *Acc. Chem. Res* **1994**, *27*, 309–314.

[46] J. Nyman, O. S. Pundyke, G. M. Day, *Phys. Chem. Chem. Phys.* **2016**, *18*, 15828–15837.

[47] G. M. Day, *Crystallogr. Rev.* **2011**, *17*, 3–52.

[48] J. Bernstein, J. D. Dunitz, A. Gavezzotti, *Cryst. Growth Des.* **2011**, *8*, 2011–2018.

[49] S. L. Price, *Chem. Soc. Rev.* **2014**, *43*, 2098–2111.

[50] S. L. Price, *Acta Cryst. B* **2013**, *69*, 313–328.

[51] D. S. Coombes, C. R. A. Catlow, J. D. Gale, A. L. Rohl, S. L. Price, *Cryst. Growth Des.* **2005**, *5*, 879–885.

[52] S. L. Price, D. E. Braun, S. M. Reutzel-Edens, *ChemComm* **2016**, *52*, 7065–7077.

[53] S. L. Price, *Acc. Chem. Res.* **2009**, *42*, 117–26.

[54] J. P. M. Lommerse, W. D. Motherwell, H. L. Ammon, J. D. Dunitz, A. Gavezzotti, D. W. M. Hofmann, F. J. J. Leusen, W. T. M. Mooij, S. L. Price, B. Schweizer, M. U. Schmidt, B. P. van Eijck, P. Verwer, D. E. Williams, *Acta Cryst. B* **2000**, *56*, 697–714.

[55] W. D. S. Motherwell, H. L. Ammon, J. D. Dunitz, A. Dzyabchenko, P. Erk, A. Gavezzotti, D. W. M. Hofmann, F. J. J. Leusen, J. P. M. Lommerse, W. T. M. Mooij, S. L. Price, H. Scheraga, B. Schweizer, M. U. Schmidt, B. P. van Eijck, P. Verwer, D. E. Williams, *Acta Cryst. B* **2002**, *58*, 647–661.

[56] G. M. Day, W. D. S. Motherwell, H. L. Ammon, S. X. M. Boerrigter, R. G. Della Valle, E Venuti, A Dzyabchenko, J. D. Dunitz, B Schweizer, B. P. van Eijck, P Erk, J. C. Facelli, V. E. Bazterra, M. B. Ferraro, D. W. M. Hofmann, F. J. J. Leusen, C Liang, C. C. Pantelides, P. G. Karamertzanis, S. L. Price, T. C. Lewis, H. Nowell, A. Torrisi, H. A. Scheraga, Y. A. Arnautova, M. U. Schmidt, P. Verwer, *Acta Cryst. B* **2005**, *61*, 511–527.

[57] G. M. Day, T. G. Cooper, A. J. Cruz-Cabeza, K. E. Hejczyk, H. L. Ammon, S. X. M. Boerrigter, J. S. Tan, R. G. Della Valle, E. Venuti, J. Jose, S. R. Gadre, G. R. Desiraju, T. S. Thakur, B. P. van Eijck, J. C. Facelli, V. E. Bazterra, M. B. Ferraro, D. W. M. Hofmann, M. A. Neumann, F. J. J. Leusen, J. Kendrick, S. L. Price, A. J. Misquitta, P. G. Karamertzanis, G. W. A. Welch, H. A. Scheraga, Y. A. Arnautova, M. U. Schmidt, J. van de Streek, A. K. Wolf, B. Schweizer, *Acta Cryst. B* **2009**, *65*, 107–125.

[58] D. A. Bardwell, C. S. Adjiman, Y. A. Arnautova, E. Bartashevich, S. X. M. Boerrigter, D. E. Braun, A. J. Cruz-Cabeza, G. M. Day, R. G. Della Valle, G. R. Desiraju, B. P. van Eijck, J. C. Facelli, M. B. Ferraro, D. Grillo, M. Habgood, D. W. M. Hofmann, F. Hofmann, K. V. J. Jose, P. G. Karamertzanis, A. V. Kazantsev, J. Kendrick, L. N. Kuleshova, F. J. J. Leusen, A. V. Maleev, A. J. Misquitta, S. Mohamed, R. J. Needs, M. A. Neumann, D. Nikylov, A. M. Orendt, R. Pal, C. C. Pantelides, C. J. Pickard, L. S. Price, S. L. Price, H. A. Scheraga, J. van de Streek, T. S. Thakur, S. Tiwari, E. Venuti, I. K. Zhitkov, *Acta Cryst. B* **2011**, *67*, 535–551.

[59] A. M. Reilly, R. I. Cooper, C. S. Adjiman, S. Bhattacharya, A. D. Boese, J. G. Brandenburg, P. J. Bygrave, R. Bylsma, J. E. Campbell, R. Car, D. H. Case, R. Chadha, J. C. Cole, K. Cosburn, H. M. Cuppen, F. Curtis, G. M. Day, R. A. DiStasio Jr, A. Dzyabchenko, B. P. van Eijck, D. M. Elking, J. A. van den Ende, J. C. Facelli, M. B. Ferraro, L. Fusti-Molnar, C.-A. Gatsiou, T. S. Gee, R. de Gelder, L. M. Ghiringhelli, H. Goto, S. Grimme, R. Guo, D. W. M. Hofmann, J. Hoja, R. K. Hylton, L. Iuzzolino, W. Jankiewicz, D. T. de Jong, J. Kendrick, N. J. J. de Klerk, H.-Y. Ko, L. N. Kuleshova, X. Li, S. Lohani, F. J. J. Leusen, A. M. Lund, J. Lv, Y. Ma, N. Marom, A. E. Masunov, P. McCabe, D. P. McMahon, H. Meekes, M. P. Metz, A. J. Misquitta, S. Mohamed, B. Monserrat, R. J. Needs, M. A. Neumann, J. Nyman, S. Obata, H. Oberhofer, A. R. Oganov, A. M. Orendt, G. I. Pagola, C. C. Pantelides, C. J. Pickard, R. Podeszwa, L. S. Price, S. L. Price, A. Pulido, M. G. Read, K. Reuter, E. Schneider, C. Schober, G. P. Shields, P. Singh, I. J. Sugden, K. Szalewicz, C. R. Taylor, A. Tkatchenko, M. E. Tuckerman, F. Vacarro, M. Vasileiadis, A. Vazquez-Mayagoitia, L. Vogt, Y. Wang, R. E. Watson, G. A. de Wijs, J. Yang, Q. Zhu, C. R. Groom, *Acta Cryst. B* **2016**, *72*, 439–459.

[60] B. Rupp, J. Wang, *Methods* **2004**, *34*, 390–407.

[61] P. Smialowski, T. Schmidt, J. Cox, A. Kirschner, D. Frishman, *Proteins: Struct. Funct. Bioinf.* **2006**, *62*, 343–355.

[62] G. Babnigg, A. Joachimiak, *J. Struct. Funct. Genomics* **2010**, *11*, 71–80.

[63] M. J. Mizianty, L. Kurgan, *Bioinformatics* **2011**, *27*, 24–33.

[64] S. Jahandideh, A. Mahdavi, *J. Theor. Biol.* **2012**, *306*, 115–119.

[65] I. M. Overton, C. A. J. van Niekerk, G. J. Barton, *Proteins* **2011**, *79*, 1027–1033.

[66] P. Strohriegl, J. V. Grazulevicius, *Adv. Mater.* **2002**, *14*, 1439–1452.

[67] J. D. Wuest, O. Lebel, *Tetrahedron* **2009**, *65*, 7393–7402.

[68] Y. Shirota, *J. Mater. Chem.* **2005**, *15*, 75–93.

[69] E. Gagnon, T. Maris, J. D. Wuest, *Org. Lett.* **2010**, *12*, 404–407.

[70] E. Gagnon, T. Maris, P.-M. Arseneault, K. E. Maly, J. D. Wuest, *Cryst. Growth Des.* **2010**, *10*, 648–657.

[71] J. A. Baird, B. Van Eerdenbrugh, L. S. Taylor, *J. Pharm. Sci.* **2010**, *99*, 3787–3806.

[72] A. Alhalaweh, A. Alzghoul, W. Kaialy, D. Mahlin, C. A. S. Bergström, *Mol. Pharmaceutics* **2014**, *11*, 3123–3132.

[73] K. Nurzyńska, J. Booth, C. J. Roberts, J. McCabe, I. Dryden, P. M. Fischer, *Mol. Pharmaceutics* **2015**, *12*, 3389–3398.

[74] M. B. Hursthouse, L. S. Huth, T. L. Threlfall, *Org. Process Res. Dev.* **2009**, *13*, 1231–1240.

[75] R. M. Bhardwaj, A. Johnston, B. F. Johnston, A. J. Florence, *CrystEngComm* **2015**, *17*, 4272–4275.

[76] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland, J. Laufer, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 244–255.

[77] N. M. O'Boyle, *J. Cheminf.* **2012**, *4*, 22.

[78] D. Weininger, *J. Chem. Inf. Comput. Sci* **1988**, *28*, 31–36.

[79] E. Anderson, G. D. Veith, D. Weininger, SMILES: A line notation and computerized interpreter for chemical structures, tech. rep., **1987**, pp. 1–4.

[80] D. Weininger, A. Weininger, J. L. Weininger, *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101.

[81] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, D. Tchekhovskoi, *J. Cheminf.* **2015**, *7*, 23.

[82] G. Landrum, RDKit: Open-Source Cheminformatics.

[83] Y.-S. Wong in *Chemical Genomics and Proteomics: Reviews and Protocols*, (Ed.: E. D. Zanders), Methods in Molecular Biology, Humana Press, Totowa, NJ, **2012**, pp. 11–23.

[84] J. Bajorath, *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 233–245.

[85] J. Leszczynski, M. Shukla, *Practical Aspects of Computational Chemistry: Methods, Concepts and Applications*, Springer, Heidelberg, **2009**, p. 203.

[86] D. C. Kombo, K. Tallapragada, R. Jain, J. Chewning, A. A. Mazurov, J. D. Speake, T. A. Hauser, S. Toler, *J. Chem. Inf. Model.* **2013**, *53*, 327–342.

[87] S. A. Wildman, G. M. Crippen, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 868–873.

[88] P. Labute, *J. Mol. Graphics Modell.* **2000**, *18*, 464–477.

[89] J. Gasteiger, M. Marsili, *Tetrahedron* **1980**, *36*, 3219–3228.

[90] K. T. Nguyen, L. C. Blum, R. van Deursen, J.-L. Reymond, *ChemMedChem* **2009**, *4*, 1803–1805.

[91] R. van Deursen, L. C. Blum, J.-L. Reymond, *J. Chem. Inf. Model.* **2010**, *50*, 1924–1934.

[92] M. Awale, J.-L. Reymond, *Bioorg. Med. Chem.* **2012**, *20*, 5372–5378.

[93] J.-L. Reymond, L. C. Blum, R. van Deursen, *Chimia* **2011**, *65*, 863–867.

[94] J.-L. Reymond, M. Awale, *ACS Chem. Neurosci.* **2012**, *3*, 649–657.

[95] J.-L. Reymond, R. van Deursen, L. C. Blum, L. Ruddigkeit, *MedChemComm* **2010**, *1*, 30–38.

[96] L. B. Kier, L. H. Hall, *J. Pharm. Sci.* **1981**, *70*, 583–589.

[97] L. B. Kier, *Quant. Struct.-Act. Relat.* **1985**, *4*, 109–116.

[98] L. B. Kier, *Quant. Struct.-Act. Relat.* **1986**, *5*, 1–7.

[99] L. H. Hall, L. B. Kier in *Reviews in Computational Chemistry, Volume 2*, (Eds.: K. B. Lipkowitz, D. B. Boyd), John Wiley & Sons, Inc., Hoboken, **1991**, pp. 367–390.

[100] M. Randic, *J. Am. Chem. Soc.* **1975**, *97*, 6609–6615.

[101] L. B. Kier, L. H. Hall, *J. Pharm. Sci.* **1976**, *65*, 1806–1809.

[102] M. Protic, A. Sabljic, *Aquat. Toxicol.* **1989**, *14*, 47–64.

[103] L. B. Kier, *Quant. Struct.-Act. Relat.* **1986**, *5*, 7–12.

[104] P. L. Luisi, *Naturwissenschaften* **1977**, *64*, 569–574.

[105] M. Dervarics, F. Ötvös, T. A. Martinek, *J. Chem. Inf. Model.* **2006**, *46*, 1431–1438.

[106] R. Todeschini, V. Consonni, *Handbook of Molecular Descriptors*, Wiley-VCH Verlag GmbH, Weinheim, **2000**, pp. 178–179.

[107] L. B. Kier, *Quant. Struct.-Act. Relat.* **1989**, *8*, 221–224.

[108] C.-W. von der Lieth, K. Stumpf-Nothof, U. Prior, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 711–716.

[109] A. Samuel, *IBM Journal* **1959**, *3*, 211–229.

[110] S. B. Kotsiantis, *Informatica* **2007**, *31*, 249–268.

[111] A. Varnek, N. Kireeva, I. V. Tetko, I. I. Baskin, V. P. Solov'ev, *J. Chem. Inf. Model.* **2007**, *47*, 1111–1122.

[112] C. X. Xue, R. S. Zhang, H. X. Liu, M. C. Liu, Z. D. Hu, B. T. Fan, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1267–1274.

[113] B. Louis, J. Singh, B. Shaik, V. K. Agrawal, P. V. Khadikar, *Chem. Biol. Drug Des.* **2009**, *74*, 190–195.

[114] Y. Pan, J. Jiang, R. Wang, H. Cao, Y. Cui, *J. Hazard. Mater.* **2009**, *168*, 962–9.

[115] A. K. Jain, M. N. Murty, P. J. Flynn, *ACM Computing Surveys* **1999**, *31*, 264–323.

[116] S. K. Murthy, *Data Mining and Knowledge Discovery* **1998**, *2*, 345–389.

[117] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, *Classification and regression trees*, CRC Press, **1984**.

[118] S. B. Kotsiantis, I. D. Zaharakis, P. E. Pintelas, *Artif. Intell. Rev.* **2006**, *26*, 159–190.

[119] L. Breiman, *Machine Learning* **2001**, *45*, 5–32.

[120] T. G. Dietterich in *Multiple Classifier Systems*, Springer-Verlag, Berlin Heidelberg, **2000**, pp. 1–15.

[121] Y. Amit, D. Geman, *Neural Computation* **1997**, *9*, 1545–1558.

[122] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, **1995**.

[123] S. R. Gunn, Support vector machines for classification and regression. Tech. rep., **1998**.

[124] C. Cortes, V. Vapnik, *Machine Learning* **1995**, *20*, 273–297.

[125] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, *IEEE Transactions on Signal Processing* **1997**, *45*, 2758–2765.

[126] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A Practical Guide to Support Vector Classification, **2003**.

[127] B. Sarojini, N. Ramaraj, S. Nickolas in CCIS 40, Springer-Verlag, Berlin Heidelberg, **2009**, pp. 533–543.

[128] X.-Y. Liu, J. Wu, Z.-H. Zhou, *IEEE Transactions on Systems Man and Cybernetics - Part B: Cybernetics* **2009**, *39*, 539–550.

[129] J. A. Hanley, B. J. McNeil, *Radiology* **1982**, *143*, 29–36.

[130] A. P. Bradley, *Pattern Recognition* **1997**, *30*, 1145–1159.

[131] T. Fawcett, *Pattern Recogn. Lett.* **2006**, *27*, 861–874.

[132] M. Tremayne, *Phil. Trans. R. Soc. Lond. A* **2004**, *362*, 2691–2707.

[133] W. I. F. David, K. Shankland, N. Shankland, *ChemComm* **1998**, 931–932.

[134] C. R. Groom, F. H. Allen, *Angew. Chem. Int. Ed.* **2014**, *53*, 662–671.

[135] W. I. F. David, K. Shankland, J. van de Streek, E. Pidcock, W. D. S. Motherwell, J. C. Cole, *J. Appl. Cryst.* **2006**, *39*, 910–915.

[136] A. Boultif, D. Louër, *J. Appl. Cryst.* **1991**, *24*, 987–993.

[137] G. S. Pawley, *J. Appl. Cryst.* **1981**, *14*, 357–361.

[138] A. J. Markvardsen, K. Shankland, W. I. F. David, J. C. Johnston, R. M. Ibberson, M. Tucker, H. Nowell, T. Griffin, *J. Appl. Cryst.* **2008**, *41*, 1177–1181.

[139] P. Fernandes, K. Shankland, W. I. F. David, A. J. Markvardsen, A. J. Florence, N. Shankland, C. K. Leech, *J. Appl. Cryst.* **2008**, *41*, 1089–1094.

[140] K. Shankland, L. McBride, W. I. F. David, N. Shankland, G. Steele, *J. Appl. Cryst.* **2002**, *35*, 443–454.

[141] K. Shankland, W. David, T. Csoka, L. McBride, *Int. J. Pharm.* **1998**, *165*, 117–126.

[142] A. J. Florence, N. Shankland, K. Shankland, W. I. F. David, E. Pidcock, X. Xu, A. Johnston, A. R. Kennedy, P. J. Cox, J. S. O. Evans, G. Steele, S. D. Cosgrove, C. S. Frampton, *J. Appl. Cryst.* **2005**, *38*, 249–259.

[143] I. J. Bruno, J. C. Cole, M. Kessler, J. Luo, W. D. S. Motherwell, L. H. Purkis, B. R. Smith, R. Taylor, R. I. Cooper, S. E. Harris, A. G. Orpen, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2133–2144.

[144] H. M. Rietveld, *Acta Cryst.* **1967**, *22*, 151–152.

[145] H. M. Rietveld, *J. Appl. Cryst.* **1969**, *2*, 65–71.

[146] W. I. F. David, *J. Appl. Cryst.* **2001**, *34*, 691–698.

[147] K. H. Stone, S. H. Lapidus, P. W. Stephens, *J. Appl. Cryst.* **2009**, *42*, 385–391.

[148] J. A. Kaduk, J. Reid, *Powder Diffr.* **2011**, *26*, 88–93.

[149] P. Scherrer, *Nachr. Ges. Wiss. Gottingen* **1918**, 98–100.

[150] W. David, M. Leoni, P. Scardi, *Materials Science Forum* **2010**, *651*, 187–200.

[151] P. W. Stephens, *J. Appl. Cryst.* **1999**, *32*, 281–289.

[152] D. Balzar in *Microstructure Analysis from Diffraction*, (Eds.: R. L. Snyder, H. J. Bunge, J. Fiala), International Union of Crystallography, **1999**.

[153] P. Jurecka, J. Sponer, J. Ernyá, P. Hobza, *Phys. Chem. Chem. Phys.* **2006**, *8*, 1985–1993.

[154] J. Van de Streek, M. A. Neumann, *Acta Cryst. B* **2010**, *66*, 544–558.

[155] J. D. Dunitz, A. Gavezzotti, *Angew. Chem. Int. Ed.* **2005**, *44*, 1766–1787.

[156] A Gavezzotti, *J. Phys. Chem. B* **2002**, *106*, 4145–4154.

[157] A Gavezzotti, *J. Phys. Chem. B* **2003**, *107*, 2344–2353.

[158] A. Gavezzotti, *New J. Chem.* **2011**, *35*, 1360–1368.

[159] S. Weaver, M. P. Gleeson, *J. Mol. Graphics Modell.* **2008**, *26*, 1315–1326.

[160] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, S. H. Bryant, *Nucleic Acids Research* **2016**, *44*, D1202–13.

[161] L. C. Blum, J.-L. Reymond, *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733.

[162] J. J. Irwin, B. K. Shoichet, *J. Chem. Inf. Model.* **2005**, *45*, 177–182.

[163] T. Sterling, J. J. Irwin, *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337.

[164] C. R. Groom, I. J. Bruno, M. P. Lightfoot, S. C. Ward, *Acta Cryst. B* **2016**, *72*, 171–179.

[165] Q. Wei, R. L. Dunbrack, *PloS ONE* **2013**, *8*, e67863.

[166] T. M. Martin, P. Harten, D. M. Young, E. N. Muratov, A. Golbraikh, H. Zhu, A. Tropsha, *J. Chem. Inf. Model.* **2012**, *52*, 2570–2578.

[167] I. D. H. Oswald, D. R. Allan, P. A. McGregor, W. D. S. Motherwell, S. Parsons, C. R. Pulham, *Acta Cryst. B* **2002**, *58*, 1057–1066.

[168] S. L. Childs, G. P. Stahly, A. Park, *Mol. Pharmaceutics* **2007**, *4*, 323–338.

[169] S. Karki, T. Friščić, L. Fabián, P. R. Laity, G. M. Day, W. Jones, *Adv. Mater.* **2009**, *21*, 3905–3909.

[170] M. A. Elbagerma, H. G. M. Edwards, T. Munshi, I. J. Scowen, *CrystEngComm* **2011**, *13*, 1877–1884.

[171] V. K. Srirambhatla, A. Kraft, S. Watt, A. V. Powell, *Cryst. Growth Des.* **2012**, *12*, 4870–4879.

[172] A. C. Müller, S. Guido in *Introduction to Machine Learning with Python*, **2016**, pp. 251–303.

[173] Q. Liu, C. Chen, Y. Zhang, Z. Hu, *Artif. Intell. Rev.* **2011**, *36*, 99–115.

[174] I. Guyon, A. Elisseeff, *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.

[175] H. Morgan, *J. Chem. Doc.* **1965**, *5*, 107–112.

[176] D. Rogers, M. Hahn, *J. Chem. Inf. Model.* **2010**, *50*, 742–54.

[177] T Tanimoto, *An elementary mathematical theory of classification and prediction*, International Business Machines Corporation, New York, **1958**.

[178] J. H. Ward, *J. Am. Stat. Assoc.* **1963**, *58*, 236–244.

[179] R. Taylor, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 59–67.

[180] D. Butina, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747–750.

[181] J. P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.

[182] S. Grimme, *J. Comput. Chem.* **2006**, *27*, 1787–1799.

[183] J. Binns, M. R. Healy, S. Parsons, C. A. Morrison, *Acta Cryst. B* **2014**, *70*, 259–267.

[184] J. van de Streek, M. A. Neumann, *Acta Cryst. B* **2014**, *70*, 1020–1032.

[185] J. G. P. Wicker, R. I. Cooper, *CrystEngComm* **2015**, *17*, 1927–1934.

[186] C. A. Lipinski, F. Lombardo, B. W. Dominy, P. J. Feeney, *Adv. Drug Delivery Rev.* **1997**, *23*, 3–25.

[187] C. A. Lipinski, *Drug Discov. Today: Technologies* **2004**, *1*, 337–341.

[188] J. J. De Yoreo, P. G. Vekilov, *Rev. Mineral. Geochem.* **2003**, *54*, 57–93.

[189] P. A. Wood, T. S. G. Olsson, J. C. Cole, S. J. Cottrell, N. Feeder, P. T. A. Galek, C. R. Groom, E. Pidcock, *CrystEngComm* **2013**, *15*, 65–72.

[190] L. Leiserowitz, G. M. J. Schmidt, *J. Chem. Soc. A.* **1969**, 2372–2382.

[191] P. Dauber, A. T. Hagler, *Acc. Chem. Res.* **1980**, *13*, 105–112.

[192] M. Seo, J. Park, S. Y. Kim, *Org. Biomol. Chem* **2012**, *10*, 5332.

[193] Z. Berkovitch-Yellin, L. Leiserowitz, *J. Am. Chem. Soc.* **1980**, *102*, 7677–7690.

[194] L. Leiserowitz, M. Tuval, *Acta Cryst. B* **1978**, *34*, 1230–1247.

[195] P. W. Cains in *Polymorphism in Pharmaceutical Solids*, New York, **2009**, pp. 76–138.

[196] P. Panini, D. Chopra, *CrystEngComm* **2012**, *14*, 1972–1989.

[197] P. Panini, D. Chopra, M. B. Hursthouse, E. D'Oria, J. J. Novoa, C. M. R. Low, J. G. Vinter, K. R. Lawson, C. J. Urch, C. A. Hunter, H. G. Kjaeergaard, A. C. Legon, B. Mennucci, D. J. Nesbitt, *New J. Chem.* **2015**, *39*, 8720–8738.

[198] N. Huang, B. K. Shoichet, J. J. Irwin, *J. Med. Chem.* **2006**, *49*, 6789–6801.

[199] M. L. Verdonk, V. Berdini, M. J. Hartshorn, W. T. M. Mooij, C. W. Murray, R. D. Taylor, P. Watson, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 793–806.

[200] M. M. Mysinger, M. Carchia, J. J. Irwin, B. K. Shoichet, *J. Med. Chem.* **2012**, *55*, 6582–6594.

[201] A. J. Cruz-Cabeza, *CrystEngComm* **2012**, *14*, 6362–6365.

[202] J. G. P. Wicker, R. I. Cooper, *J. Chem. Inf. Model.* **2016**, *56*, 2347–2352.

[203] J.-P. Ebejer, G. M. Morris, C. M. Deane, *J. Chem. Inf. Model.* **2012**, *52*, 1146–1158.

[204] T. Halgren, *J. Comput. Chem.* **1996**, *17*, 490–519.

[205] P. Tosco, N. Stiefl, G. Landrum, *J. Cheminf.* **2014**, *6*, 37.

[206] M. A. González, *Collection SFN* **2011**, *12*, 169–200.

[207] T. A. Halgren, *J. Comput. Chem.* **1999**, *20*, 730–748.

[208] S. Riniker, G. A. Landrum, *J. Chem. Inf. Model.* **2015**, *55*, 2562–2574.

[209] B. Bugenhagen, Y. Al Jasem, M. Al-Azani, T. Thiemann, *Acta Cryst. E* **2014**, *70*, o265.

[210] A. A. Coelho, TOPAS-Academic, version 4.1, Coelho Software, Brisbane, **2007**.

[211] J. C. Cole, E. A. Kabova, K. Shankland, *Powder Diffr.* **2014**, *29*, S19–S30.

[212] A. J. Markvardsen, W. I. F. David, K. Shankland, *Acta Cryst. A* **2002**, *58*, 316–326.

[213] D. J. Watkin, R. I. Cooper, A. Collins, P. A. Wood in American Crystallographic Association Annual Meeting: Transformations and Structural Oddities in Molecular Crystals: In Honor of Bruce M. Foxman, **2012**, pp. 1–19.

[214] M. J. Spillman, K. Shankland, A. C. Williams, J. C. Cole, *Journal of Applied Crystallography* **2015**, *48*, 2033–2039.

[215] B. E. Warren, *A Physical Review* **1941**, *59*, 693–698.

Appendices

Appendix A

Model Information

A.1 Descriptor definitions

Table A.1 Descriptor definitions

RDKit Descriptors	Paper
MolWt, HeavyAtomMolWt, NumRadicalElectrons, NumValenceElectrons, HeavyAtomCount, NumHeteroatoms, NumRotatableBonds, RingCount	Self-explanatory; the implementation can be found in the open source RDKit version 2016.09.1 descriptor module
Chi0v, Chi1v, Chi2v, Chi3v, Chi4v, ChiNv, HallKierAlpha , Kappa1, Kappa2, Kappa3	Rev. Comp. Chem. vol 2, 367-422, (1991)
Chi0n, Chi1n, Chi2n, Chi3n, Chi4n, ChiNn	Similar to Hall Kier ChiXv, but uses nVal instead of valence
BalabanJ	Chem. Phys. Lett. vol 89, 399-404, (1982)
BertzCT	J. Am. Chem. Soc., vol 103, 3599-601 (1981)
Ipc	J. Chem. Phys., vol 67, 4517-33 (1977)
LabuteASA PEOE-VSA1 – PEOE-VSA14 SMR-VSA1 – SMR-VSA10 SlogP-VSA1 – SlogP-VSA12	J. Mol. Graph. Mod., vol 18, 464-77 (2000)
TPSA	J. Med. Chem., vol 43, 3714-7, (2000)
MolLogP, MolMR	J. Chem. Inform. Comput. Sci., vol 39, 868-73 (1999)
EState-VSA1 – EState-VSA11 VSA-EState1 – VSA-EState10	MOE-type descriptors using electrotopological state indices and surface area contributions developed at RD from J. Chem. Inform. Comput. Sci., vol 31, 76-81 (1991)

A.2 Fragment definitions

Table A.2 Fragment definitions

Fragment name	Definition
NHOHCount	Number of NHs and OHs
NOCount	Number of nitrogen and oxygen atoms
NumHAcceptors	Number of Hydrogen Bond Acceptors
NumHDonors	Number of Hydrogen Bond Donors
fr-Al-COO	Number of aliphatic carboxylic acids
fr-Al-OH	Number of aliphatic hydroxyl groups
fr-Al-OH-noTert	Number of aliphatic hydroxyl groups excluding tert-OH
fr-ArN	Number of N functional groups attached to aromatics
fr-Ar-COO	Number of aromatic carboxylic acids
fr-Ar-N	Number of aromatic nitrogens
fr-Ar-NH	Number of aromatic amines
fr-Ar-OH	Number of aromatic hydroxyl groups
fr-COO	Number of carboxylic acids
fr-COO2	Number of carboxylic acids
fr-C-O	Number of carbonyl
fr-C-O-noCOO	Number of carbonyl O, excluding COOH
fr-C-S	Number of thiocarbonyl
fr-HOCCN	Number of C(OH)CCN-Ctert-alkyl or C(OH)CCNcyclic
fr-Imine	Number of Imines
fr-NH0	Number of Tertiary amines
fr-NH1	Number of Secondary amines
fr-NH2	Number of Primary amines
fr-N-O	Number of hydroxylamine groups
fr-Ndealkylation1	Number of XCCNR groups
fr-Ndealkylation2	Number of tert-alicyclic amines (no heteroatoms, not quinine-like bridged N)

Fragment name	Definition
fr-Nhpyrrole	Number of H-pyrrole nitrogens
fr-SH	Number of thiol groups
fr-aldehyde	Number of aldehydes
fr-alkyl-carbamate	Number of alkyl carbamates
fr-alkyl-halide	Number of alkyl halides
fr-allylic-oxid	Number of allylic oxidation sites excluding steroid dienone
fr-amide	Number of amides
fr-amidine	Number of amidine groups
fr-aniline	Number of anilines
fr-aryl-methyl	Number of aryl methyl sites for hydroxylation
fr-azide	Number of azide groups
fr-azo	Number of azo groups
fr-barbitur	Number of barbiturate groups
fr-benzene	Number of benzene rings
fr-benzodiazepine	Number of benzodiazepines with no additional fused rings
fr-bicyclic	Number of bicyclic rings
fr-diazo	Number of diazo groups
fr-dihydropyridine	Number of dihydropyridines
fr-epoxide	Number of epoxide rings
fr-ester	Number of esters
fr-ether	Number of ether oxygens (including phenoxy)
fr-furan	Number of furan rings
fr-guanido	Number of guanidine groups
fr-halogen	Number of halogens
fr-hdrzine	Number of hydrazine groups
fr-hdrzone	Number of hydrazone groups
fr-imidazole	Number of imidazole rings
fr-imide	Number of imide groups
fr-isocyan	Number of isocyanates
fr-isothiocyan	Number of isothiocyanates
fr-ketone	Number of ketones
fr-ketone-Topliss	Number of ketones excluding diaryl, a,b-unsat.
fr-lactam	Number of beta lactams

Fragment name	Definition
fr-lactone	Number of cyclic esters (lactones)
fr-methoxy	Number of methoxy groups -OCH3
fr-morpholine	Number of morpholine rings
fr-nitrile	Number of nitriles
fr-nitro	Number of nitro groups
fr-nitro-arom	Number of nitro benzene ring substituents
fr-nitro-arom-nonortho	Number of non-ortho nitro benzene ring substituents
fr-nitroso	Number of nitroso groups, excluding NO2
fr-oxazole	Number of oxazole rings
fr-oxime	Number of oxime groups
fr-para-hydroxylation	Number of para-hydroxylation sites
fr-phenol	Number of phenols
fr-phenol-noOrthoHbond	Number of phenolic OH excluding ortho intramolecular Hbond substituents
fr-phos-acid	Number of phosphoric acid groups
fr-phos-ester	Number of phosphoric ester groups
fr-piperdine	Number of piperdine rings
fr-piperzine	Number of piperzine rings
fr-priamide	Number of primary amides
fr-prisulfonamd	Number of primary sulfonamides
fr-pyridine	Number of pyridine rings
fr-quatN	Number of quarternary nitrogens
fr-sulfide	Number of thioether
fr-sulfonamd	Number of sulfonamides
fr-sulfone	Number of sulfone groups
fr-term-acetylene	Number of terminal acetylenes
fr-tetrazole	Number of tetrazole rings
fr-thiazole	Number of thiazole rings
fr-thiocyan	Number of thiocyanates
fr-thiophene	Number of thiophene rings
fr-unbrch-alkane	Number of unbranched alkanes of at least 4 members (excludes halogenated alkanes)
fr-urea	Number of urea groups

A.3 Common solvent SMILES

Table A.3 SMILES for common solvents

Smiles
S=C=S
O
Cc1ccccn1
CO
ClC(Cl)Cl
ClCCl
CCO
O=C(O)C(F)(F)F
CC(C)=O
CS(C)=O
CCCCCC
C1CCOC1
Cc1ccccc1
CN(C)C=O
c1ccccc1
CCOC(C)=O
CCOCC
N=C(N)N
CC#N
CC(C)O
O=c1cccc[nH]1
ClC(Cl)C(Cl)Cl
CC(=O)O
Cl
ClCCCl
c1cnccn1
c1ccncc1
C1COCCO1
HH
CC(C)OC(C)C
O=C(O)CC(=O)O
OO
COC(C)(C)C
C[N+](=O)[O-]
CC(N)=O
CN1CCCC1=O

Smiles
Oc1ccccc1
ClC(Cl)(Cl)Cl
CCC(=O)CC
O=C(O)C(=O)O
COC
O=CO
CCCC
CCNCC
CC(=O)N(C)C
Nc1ccccc1
CCCCC
CCCCO
OCCO
CCC(C)O
CCCO
BrC(Br)(Br)Br
CCC(C)=O
COCCOC
O=C1CCCCC1
CCC(=O)O
CC(C)(C)O
CCC
OCC(F)(F)F
OCO
Clc1ccccc1
O=S(=O)(O)O
OCCCCO
N
Cc1ccncc1
C
CCCCCCC
C1CCCC1
C1COCCN1
Brc1ccccc1
F
O=[N+]([O-])O
CC(O)C(C)O
Br
NC(N)=O

A.4 Co-crystal experimental results.

	2-Nitrobenzoic Acid	3-Nitrobenzoic Acid	4-Nitrobenzoic Acid	2-Hydroxybenzoic Acid	3-Hydroxybenzoic Acid	4-Hydroxybenzoic Acid	2-Fluorobenzoic Acid	3-Fluorobenzoic Acid	4-Fluorobenzoic Acid	2-Aminobenzoic Acid	3-Aminobenzoic Acid	4-Aminobenzoic Acid	2-Methoxybenzoic Acid	3-Methoxybenzoic Acid	4-Methoxybenzoic Acid	2-Methylbenzoic Acid	3-Methylbenzoic Acid	4-Methylbenzoic Acid
Nicotinamide	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1
Isonicotinamide	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1
4,4'-Bipyridine	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1
Fumaric Acid	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
Salicylic Acid	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	1	0	0
Urea	1	1	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
Benzamide	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Oxalic Acid	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0
Hydroquinone	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	0	1
2-Pyrrolidinone	1	1	1	1	1	1	1	0	1	1	1	0	2	2	0	0	1	0
Caffeine	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1
Caprylic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-Glutamic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nicotinic Acid	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Pyridoxine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,3'-Thiodipropionic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Malic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trans-Aconitic Acid	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
L-Ascorbic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Riboflavin	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

(a) Acids data matrix.

	2-Nitrobenzamide	3-Nitrobenzamide	4-Nitrobenzamide	2-Hydroxybenzamide	4-Hydroxybenzamide	3-Fluorobenzamide	2-Aminobenzamide	3-Aminobenzamide	4-Aminobenzamide	2-Methoxybenzamide	3-Methoxybenzamide	4-Methoxybenzamide	2-Methylbenzamide	3-Methylbenzamide	4-Methylbenzamide
Nicotinamide	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0
Isonicotinamide	2	2	2	2	2	2	0	2	2	2	2	2	0	2	2
4,4'-Bipyridine	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0
Fumaric Acid	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1
Salicylic Acid	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1
Urea	0	1	0	1	1	0	0	0	1	1	0	1	0	1	1
Benzamide	0	1	0	1	1	0	0	0	0	1	0	0	1	0	0
Oxalic Acid	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1
Hydroquinone	0	0	0	1	1	1	0	0	1	1	1	1	1	0	1
2-Pyrrolidinone	2	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Caffeine	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0
Caprylic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-Glutamic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nicotinic Acid	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Pyridoxine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,3'-Thiodipropionic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Malic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trans-Aconitic Acid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-Ascorbic Acid	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
Riboflavin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) Amides data matrix.

A.5 Python scripts

Listing A.1 Script calculating MQN descriptors for an example molecule. The code runs in Python 2.7 and uses the following packages: RDKit version 2016.03.1; SciKit Learn version 0.17.1; and NumPy 1.11.1

```
1
2  from rdkit import Chem
3  from rdkit.Chem import Descriptors
4  import collections
5  import math
6  import numpy
7
8  def GenerateMQNs(m):
9
10     Chem.Kekulize(m)
11     mqnarray = []
12     counts = collections.defaultdict(int)
13     sssr = Chem.GetSymmSSSR(m)
14     ri = m.GetRingInfo()
15
16     #ringinfo
17     for ring in sssr:
18         if len(ring)<10:
19             counts['R{}'.format(len(ring))]+=1
20         else:
21             counts['RG10']+=1
22
23     #atominfo
24     for atom in m.GetAtoms():
```

```
25
26          symb = atom.GetSymbol()
27
28          if symb == 'N' or symb == 'O':
29              if atom.IsInRing():
30                  counts['C'+symb]+=1
31              else:
32                  counts['A'+symb]+=1
33
34          else:
35              counts[symb]+=1
36
37          num_hydrogens = atom.GetTotalNumHs()
38          counts['H']+=num_hydrogens
39
40          bdcount = len(atom.GetNeighbors())
41
42          if atom.IsInRing():
43              counts['C{}V'.format(bdcount)]+=1
44          else:
45              counts['A{}V'.format(bdcount)]+=1
46
47          if ri.NumAtomRings(atom.GetIdx())>1:
48              counts['AFRC']+=1
49
50          charge = atom.GetFormalCharge()
51
52          if charge>0:
```

```
53                counts['POSC']+=charge
54            elif charge<0:
55                counts['NEGC']+=math.fabs(charge)
56
57        #bondinfo
58        rbc = Descriptors.NumRotatableBonds(m)
59
60        for bond in m.GetBonds():
61            if bond.IsInRing():
62                counts['C{}'.format(bond.GetBondType())]+=1
63            else:
64                counts['A{}'.format(bond.GetBondType())]+=1
65
66            if ri.NumBondRings(bond.GetIdx())>1:
67                counts['BFRC']+=1
68
69        if rbc < 0:
70            rbc = 0
71
72        counts['RBC']+= rbc
73
74        #hbondinfo
75        for atom in m.GetAtoms():
76
77            symb = atom.GetSymbol()
78
79            if symb == 'N' or symb == 'O':
80                counts['HBDM']+=atom.GetTotalNumHs()
```

```
81
82      HAcceptorSmarts =
            ↪ Chem.MolFromSmarts('[$([O,S;H1;v2]-[!$(*=[O,N,P,S])]),$([O,S;H0;v2]),$([O,S;-]),$
83
84      matches = m.GetSubstructMatches(HAcceptorSmarts)
85
86      symbols = [m.GetAtomWithIdx(element).GetAtomicNum() for tupl in matches for
            ↪ element in tupl]
87
88      for atomicNum in symbols:
89          if atomicNum == 6 or 18:
90              counts['HBAM']+=2
91          else:
92              counts['HBAM']+=1
93
94      counts['HBA']+=Descriptors.NumHAcceptors(m)
95      counts['HBD']+=Descriptors.NumHDonors(m)
96
97      #collect counts
98
99      mqnarray.append(counts['C'])
100     mqnarray.append(counts['F'])
101     mqnarray.append(counts['Cl'])
102     mqnarray.append(counts['Br'])
103     mqnarray.append(counts['I'])
104     mqnarray.append(counts['S'])
105     mqnarray.append(counts['P'])
106     mqnarray.append(counts['AN'])
```

```
107         mqnarray.append(counts['CN'])
108         mqnarray.append(counts['AO'])
109         mqnarray.append(counts['CO'])
110         mqnarray.append(m.GetNumHeavyAtoms())
111         mqnarray.append(counts['ASINGLE'])
112         mqnarray.append(counts['ADOUBLE'])
113         mqnarray.append(counts['ATRIPLE'])
114         mqnarray.append(counts['CSINGLE'])
115         mqnarray.append(counts['CDOUBLE'])
116         mqnarray.append(counts['CTRIPLE'])
117         mqnarray.append(counts['RBC'])
118         mqnarray.append(counts['HBAM'])
119         mqnarray.append(counts['HBA'])
120         mqnarray.append(counts['HBDM'])
121         mqnarray.append(counts['HBD'])
122         mqnarray.append(counts['NBGC'])
123         mqnarray.append(counts['POSC'])
124         mqnarray.append(counts['A1V'])
125         mqnarray.append(counts['A2V'])
126         mqnarray.append(counts['A3V'])
127         mqnarray.append(counts['A4V'])
128         mqnarray.append(counts['C2V'])
129         mqnarray.append(counts['C3V'])
130         mqnarray.append(counts['C4V'])
131         mqnarray.append(counts['R3'])
132         mqnarray.append(counts['R4'])
133         mqnarray.append(counts['R5'])
134         mqnarray.append(counts['R6'])
```

```
135         mqnarray.append(counts['R7'])
136         mqnarray.append(counts['R8'])
137         mqnarray.append(counts['R9'])
138         mqnarray.append(counts['RG10'])
139         mqnarray.append(counts['AFRC'])
140         mqnarray.append(counts['BFRC'])
141
142         return mqnarray
143
144 example_molecule = Chem.MolFromSmiles('c1ccccc1CCCCC')
145 example_MQNs = GenerateMQNs(example_molecule,50)
```

Listing A.2 Script calculating $nConf_{20}$ for an example molecule. The code runs in Python 2.7 and uses the following packages: RDKit version 2016.03.1; SciKit Learn version 0.17.1; and NumPy 1.11.1

```
1  from rdkit import Chem
2  from rdkit.Chem import AllChem
3  from collections import OrderedDict
4  import numpy as np
5
6  def GenerateConformers(mol, numConfs):
7
8      #Add H atoms to skeleton
9      molecule = Chem.AddHs(mol)
10
11     conformerIntegers = []
12
13     #Embed and optimise the conformers
14     conformers = AllChem.EmbedMultipleConfs(molecule, numConfs,
```

```
        ↪ pruneRmsThresh=0.5, numThreads=3)
    optimised_and_energies = AllChem.MMFFOptimizeMoleculeConfs(molecule,
        ↪ maxIters=600, numThreads=3, nonBondedThresh=100.0)

    EnergyDictionaryWithIDAsKey = {}
    FinalConformersToUse = {}

    #Only keep the conformers which were successfully fully optimised
    for conformer in conformers:
        optimised, energy = optimised_and_energies[conformer]
        if optimised == 0:
            EnergyDictionaryWithIDAsKey[conformer] = energy
            conformerIntegers.append(conformer)

    #Keep the lowest energy conformer
    lowestenergy = min(EnergyDictionaryWithIDAsKey.values())

    for k, v in EnergyDictionaryWithIDAsKey.iteritems():
        if v == lowestenergy:
            lowestEnergyConformerID = k

    FinalConformersToUse[lowestEnergyConformerID] = lowestenergy

    #Remove H atoms to speed up substructure matching
    molecule = AllChem.RemoveHs(molecule)

    #Find all substructure matches of the molecule with itself, to account for
        ↪ symmetry
```

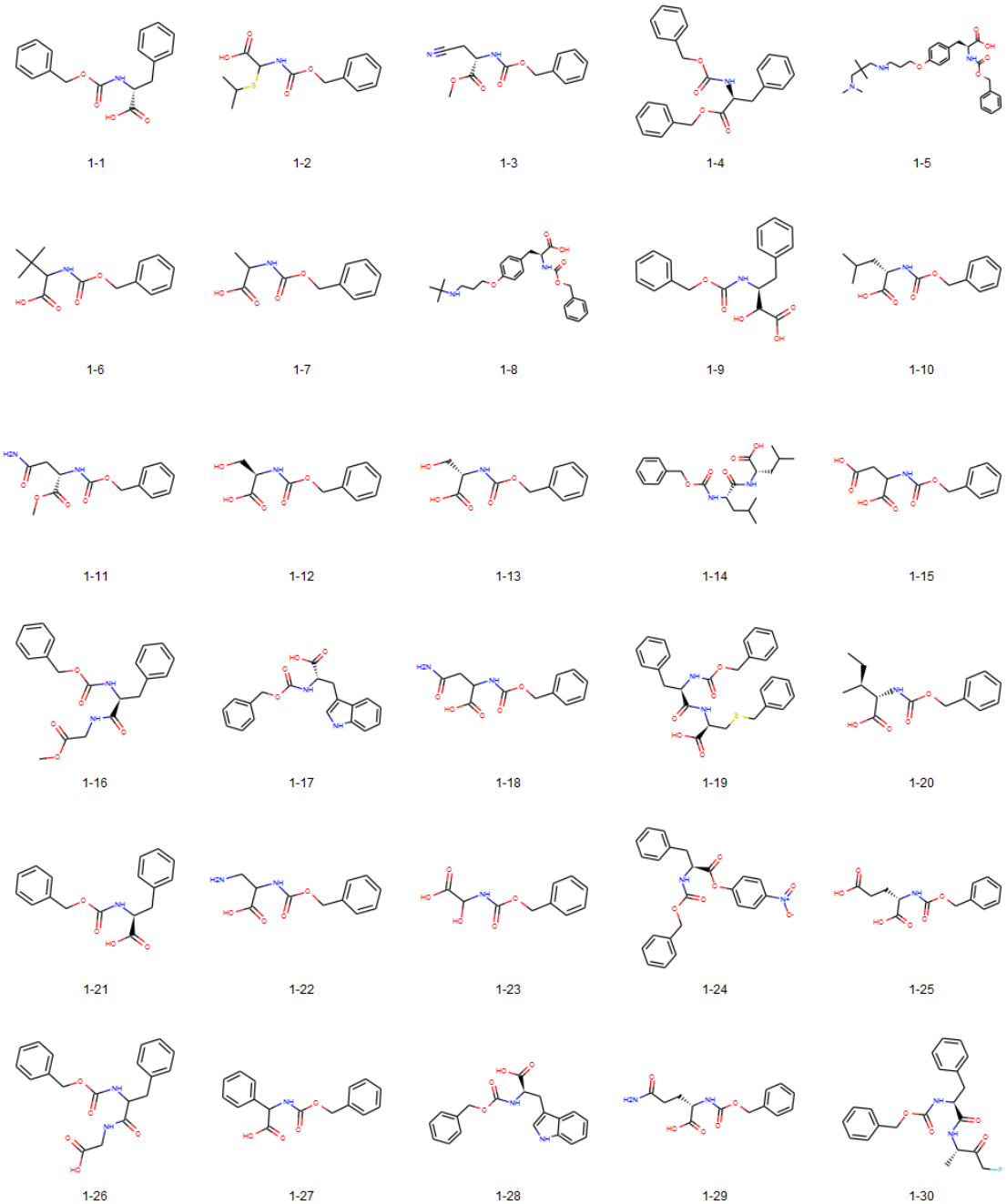
```
40      matches = molecule.GetSubstructMatches(molecule,uniquify=False)
41      maps = [list(enumerate(match)) for match in matches]
42
43      #Loop over conformers other than the lowest energy one
44      for conformerID in EnergyDictionaryWithIDAsKey.keys():
45
46          okayToAdd = True
47
48          #Loop over reference conformers already added to list
49          for finalconformerID in FinalConformersToUse.keys():
50
51              #Calculate the best RMS of this conformer with the reference
                    ↪ conformer in the list
52              RMS = AllChem.GetBestRMS(molecule, molecule,finalconformerID,
                    ↪ conformerID, maps)
53
54              #Do not add if a match is found with a reference conformer
55              if RMS < 1.0:
56                  okayToAdd = False
57                  break
58
59          #Add the conformer if the RMS is greater than 1.0 for every reference
                ↪ conformer
60          if okayToAdd:
61              FinalConformersToUse[conformerID] =
                    ↪ EnergyDictionaryWithIDAsKey[conformerID]
62
63      #Sort the conformers by energy
```

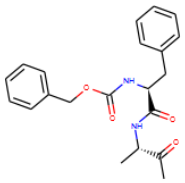
```
64      sortedDictionary = OrderedDict(sorted(FinalConformersToUse.iteritems(),
            ↪ key=lambda t: t[1]))
65
66      energies = [val for val in sortedDictionary.itervalues()]
67
68      return energies
69
70  def Calc_nConf20(energylist):
71      energy_descriptor = 0
72
73      relativeenergies = np.array(energylist) - energylist[0]
74
75      #Only look at the energies of conformers other than the global minimum
76      for energy in relativeenergies[1:]:
77
78          #Optimized lower and upper energy limits for conformer energy
79          if 0 <= energy < 20:
80              energy_descriptor += 1
81
82      return energy_descriptor
83
84  example_molecule = Chem.MolFromSmiles('c1ccccc1CCCCC')
85  example_energylist = GenerateConformers(example_molecule,50)
86  example_energy_descriptor = Calc_nConf20(example_energylist)
```

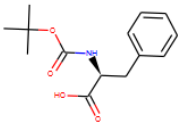
Appendix B

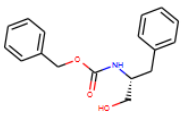
Powder Diffraction Information

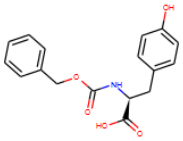
B.1 Molecules chosen for powder diffraction studies.

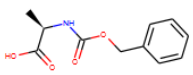


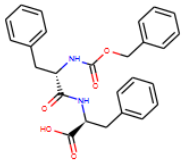

1-31

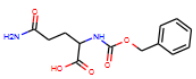

1-32

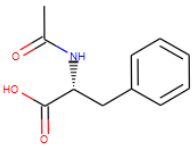

1-33

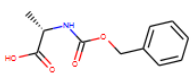

1-34

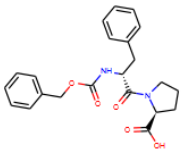

1-35

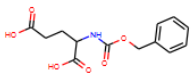

1-36

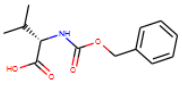

1-37

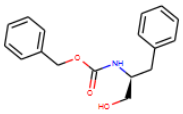

1-38

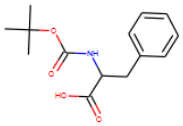

1-39

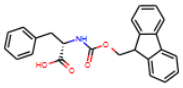

1-40

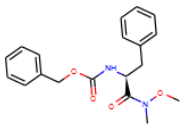

1-41

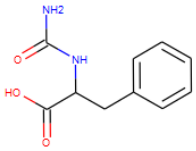

1-42

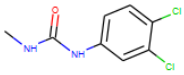

1-43

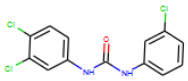

1-44

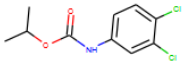

1-45

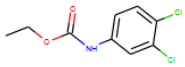

1-46

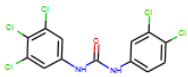

1-47

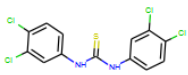

2-1

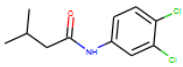

2-2

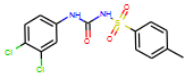

2-3

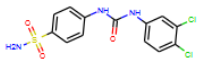

2-4

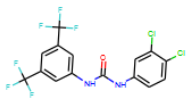

2-5

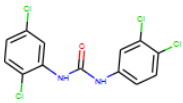

2-6

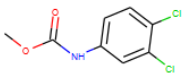

2-7

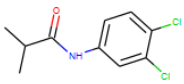

2-8

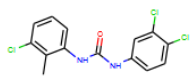

2-9


2-10

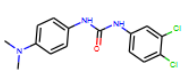

2-11


2-12

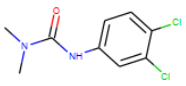

2-13



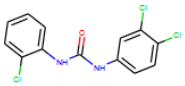
2-14



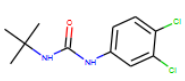
2-15



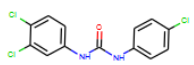
2-16



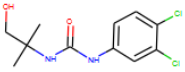
2-17



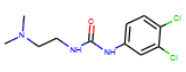
2-18



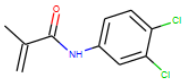
2-19



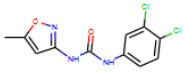
2-20



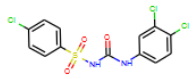
2-21



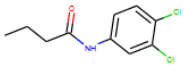
2-22



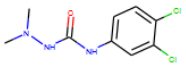
2-23



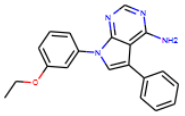
2-24



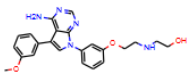
2-25



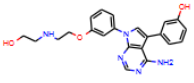
2-26



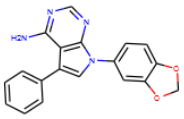
3-1



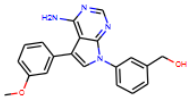
3-2



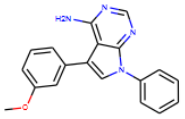
3-3



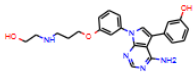
3-4



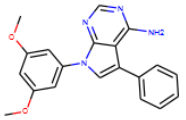
3-5



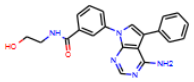
3-6



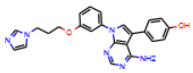
3-7



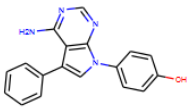
3-8



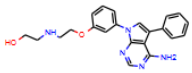
3-9



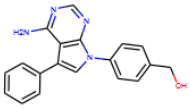
3-10



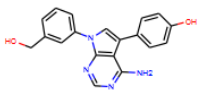
3-11



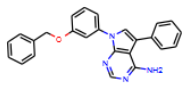
3-12



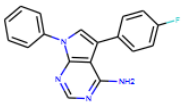
3-13



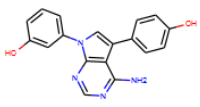
3-14



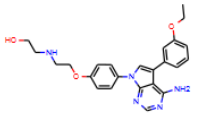
3-15



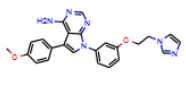
3-16



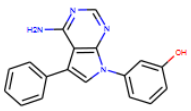
3-17



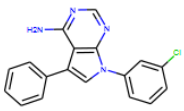
3-18



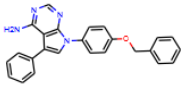
3-19



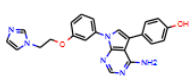
3-20



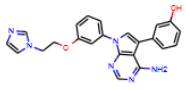
3-21



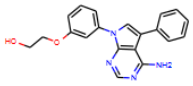
3-22



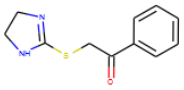
3-23



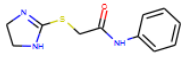
3-24



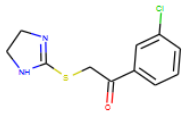
3-25



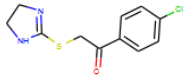
4-1



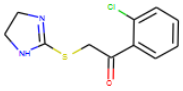
4-2



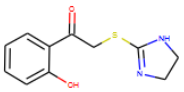
4-3



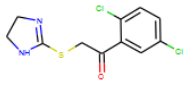
4-4



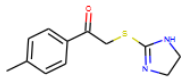
4-5



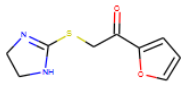
4-6



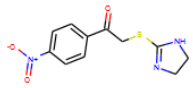
4-7



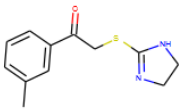
4-8



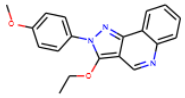
4-9



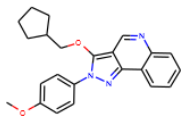
4-10



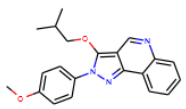
4-11



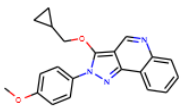
5-1



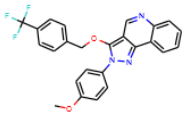
5-2



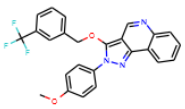
5-3



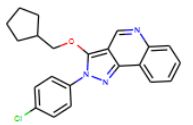
5-4



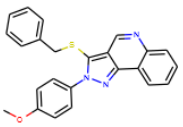
5-5



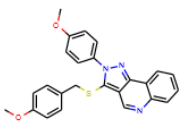
5-6



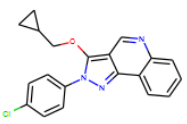
5-7



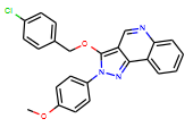
5-8



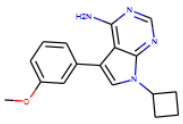
5-9



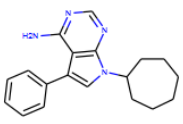
5-10



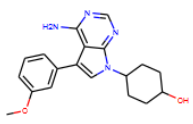
5-11



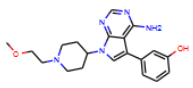
6-1



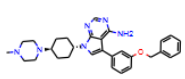
6-2



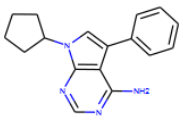
6-3



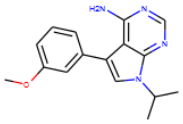
6-4



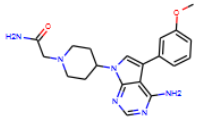
6-5



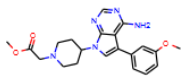
6-6



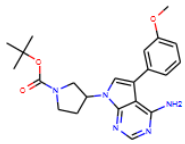
6-7



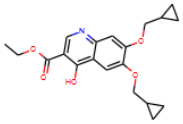
6-8



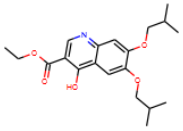
6-9



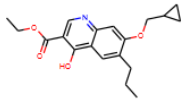
6-10



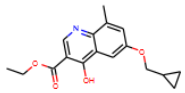
7-1



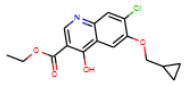
7-2



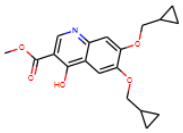
7-3



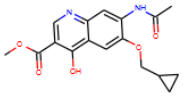
7-4



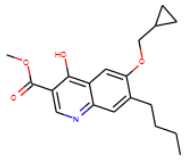
7-5



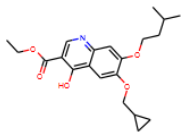
7-6



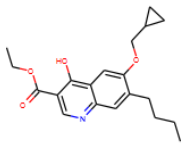
7-7



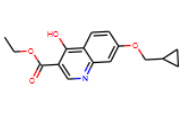
7-8



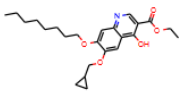
7-9



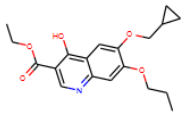
7-10



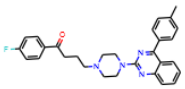
7-11



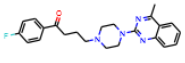
7-12



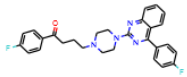
7-13



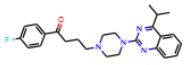
8-1



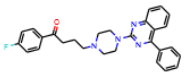
8-2



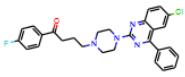
8-3



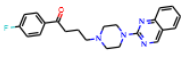
8-4



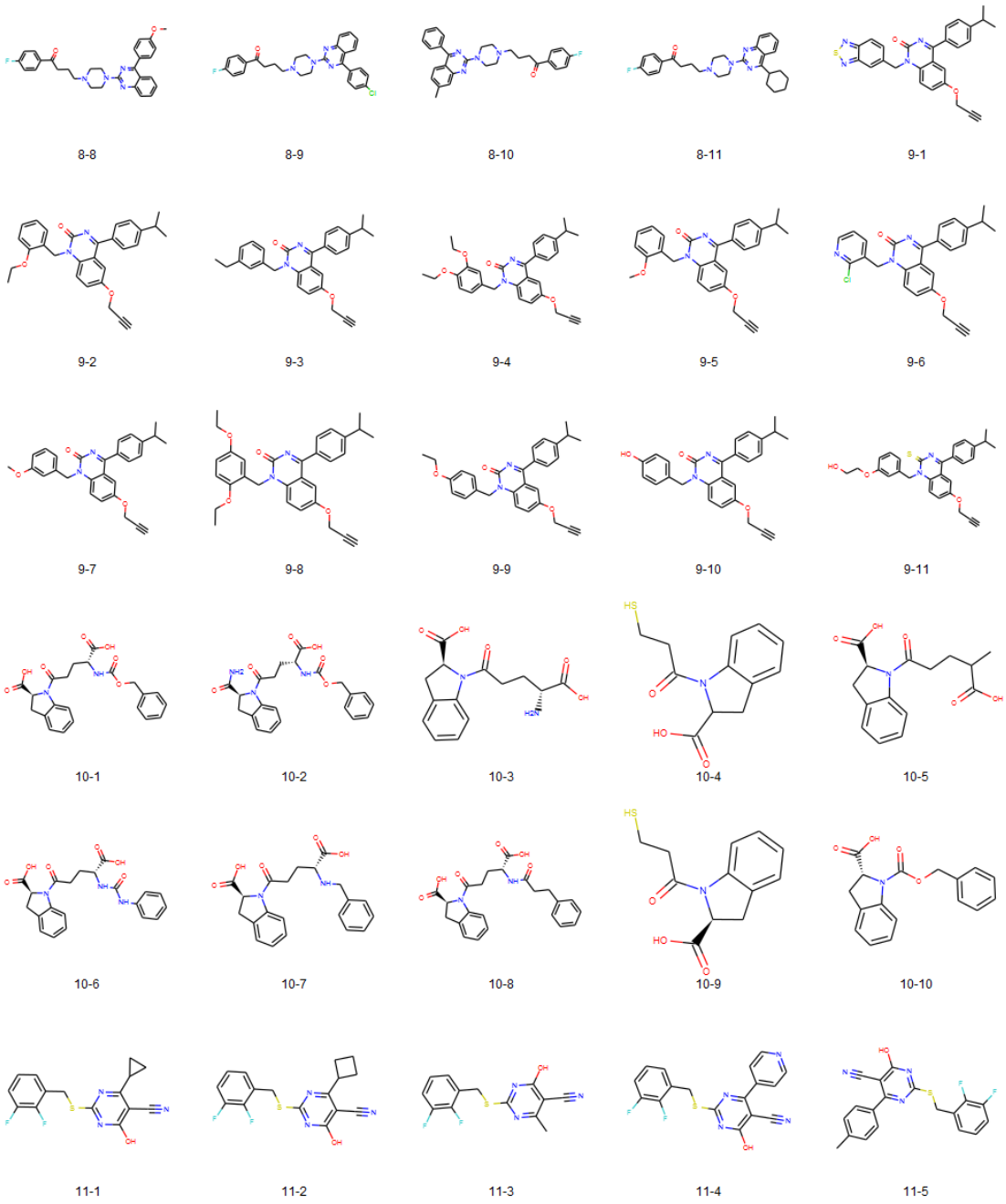
8-5

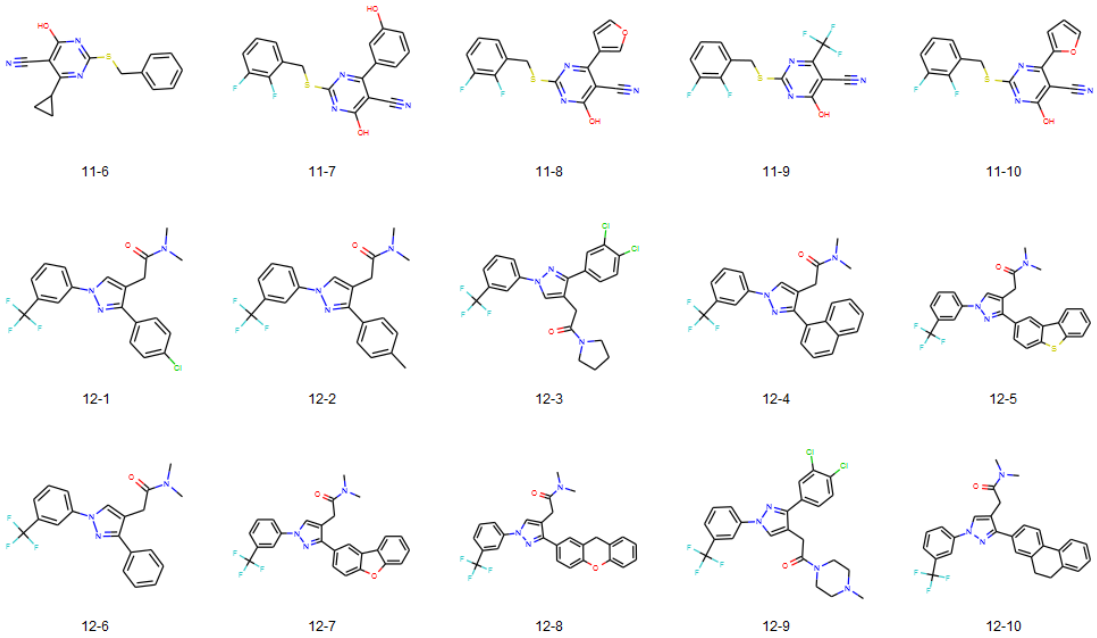


8-6

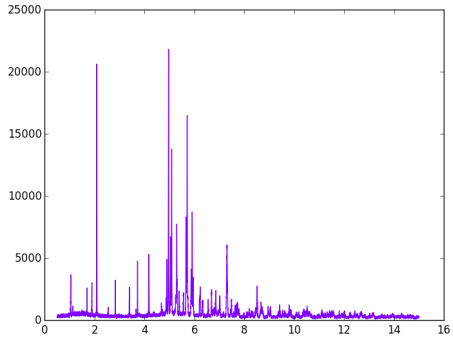


8-7

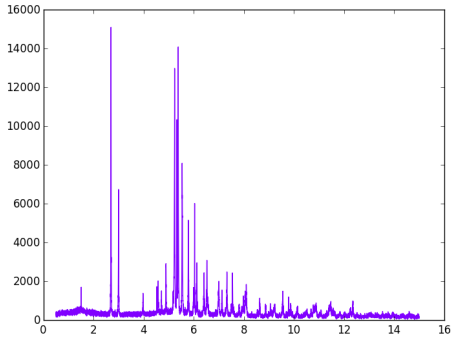

8-8
8-9
8-10
8-11
9-1
9-2
9-3
9-4
9-5
9-6
9-7
9-8
9-9
9-10
9-11
10-1
10-2
10-3
10-4
10-5
10-6
10-7
10-8
10-9
10-10
11-1
11-2
11-3
11-4
11-5



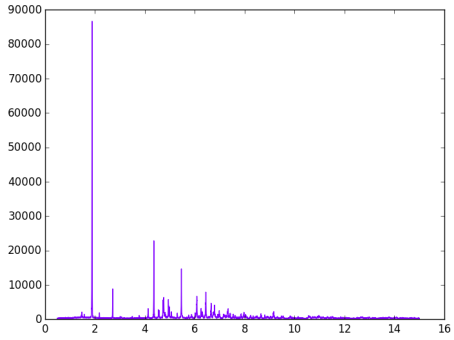
B.2 Powder diffraction patterns



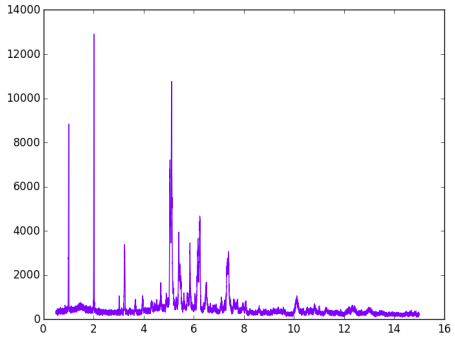
1-1.



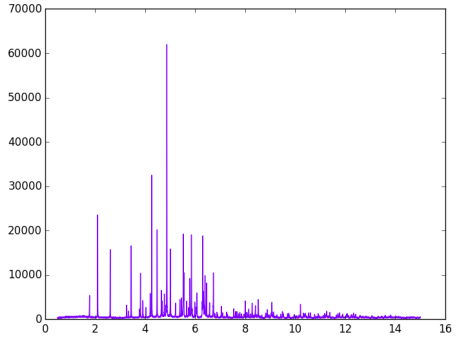
1-3.



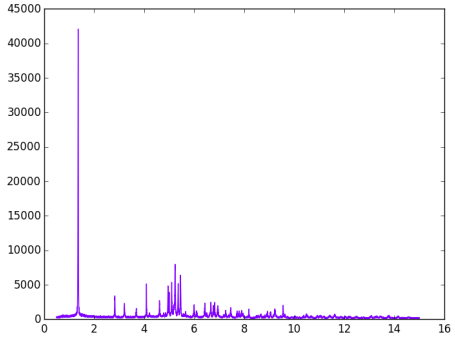
1-2.



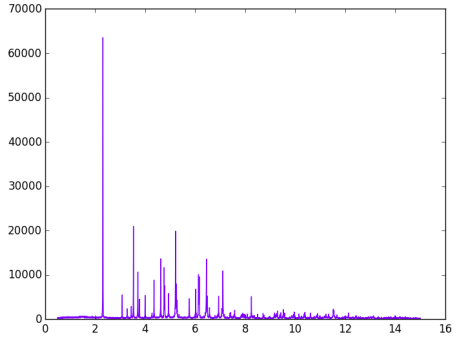
1-4.



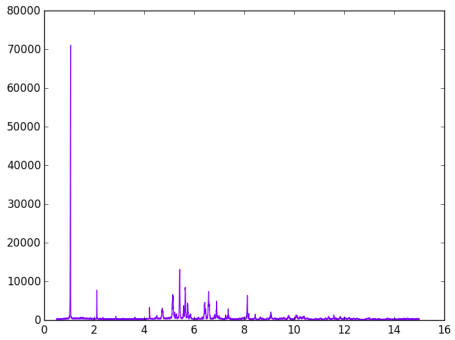
1-5.



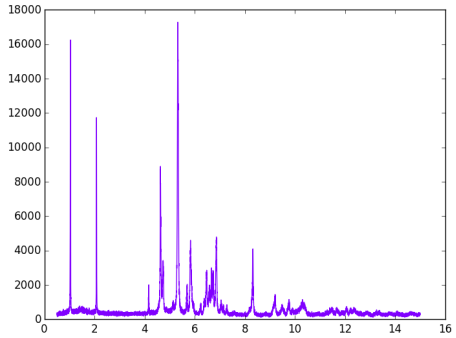
1-9.



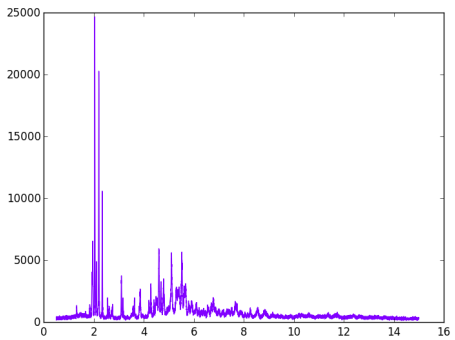
1-6.



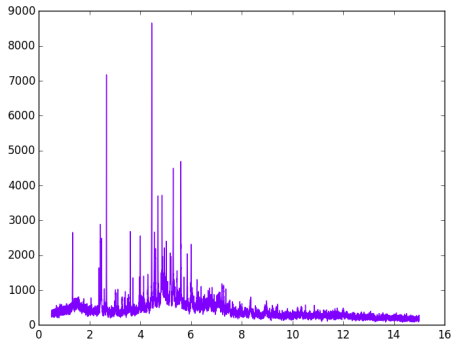
1-13.



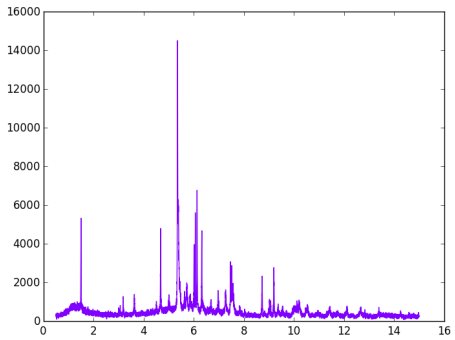
1-7.



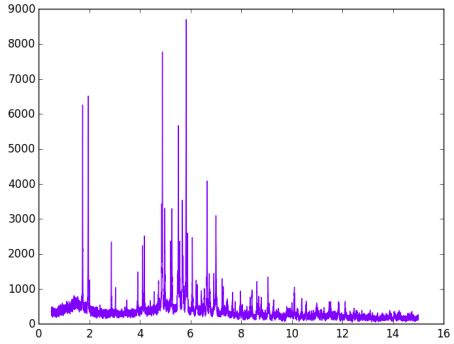
1-14.



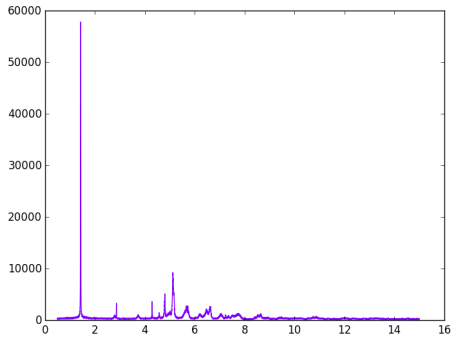
1-8.



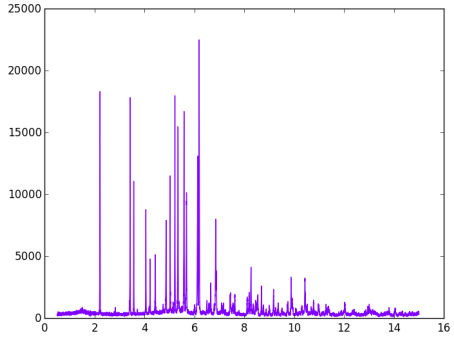
1-15.



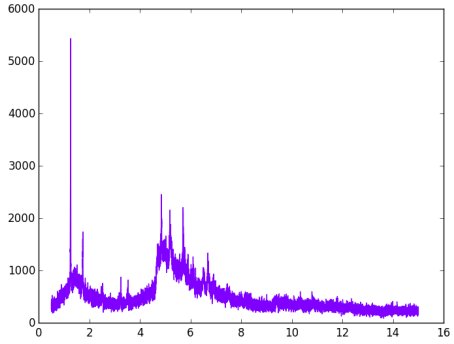
1-16.



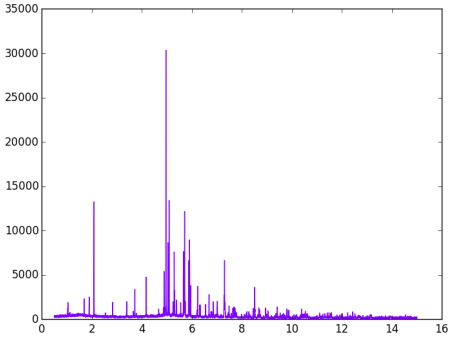
1-17.



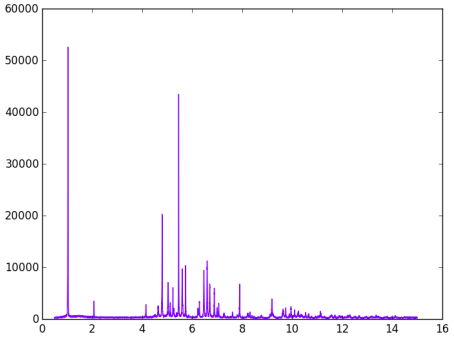
1-18.



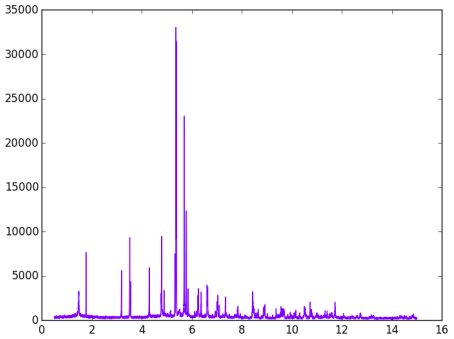
1-19.



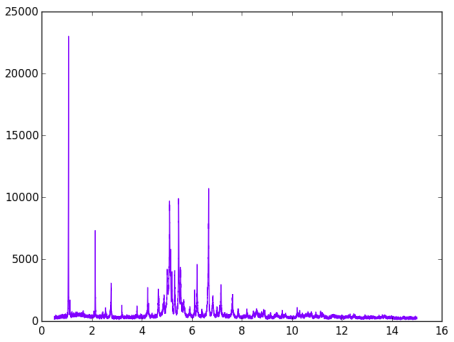
1-21.



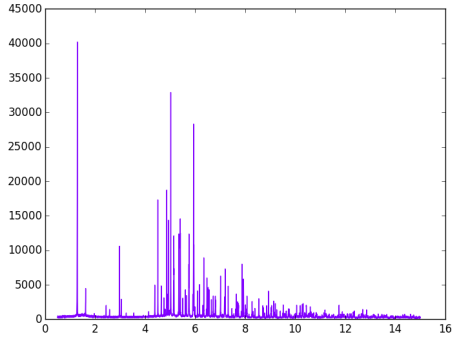
1-22.



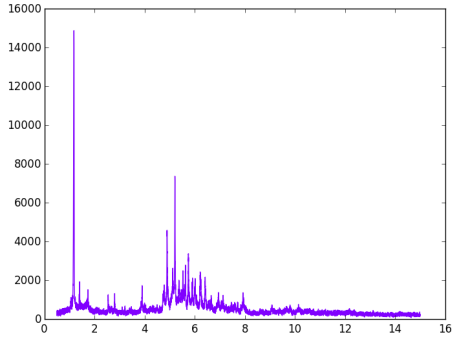
1-23.



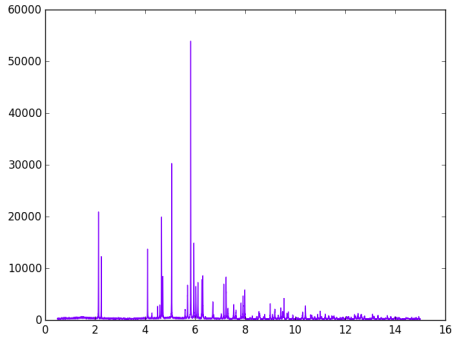
1-24.



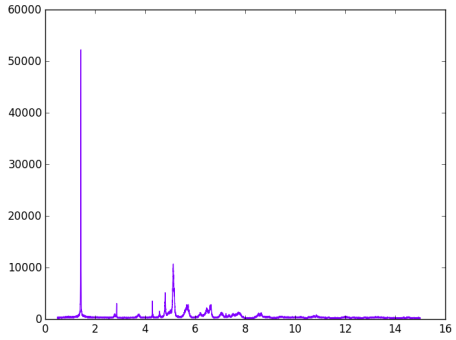
1-25.



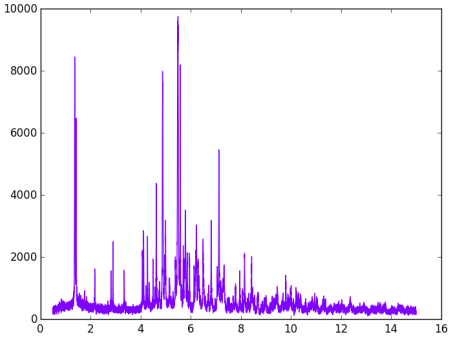
1-26.



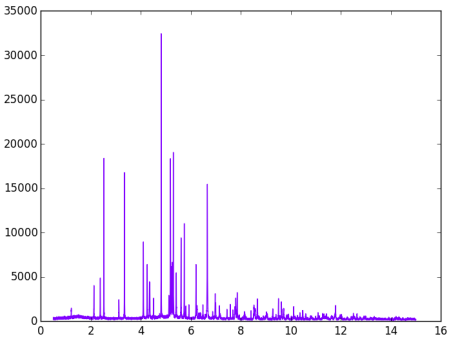
1-27.



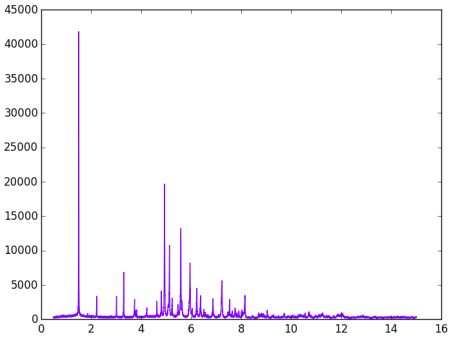
1-28.



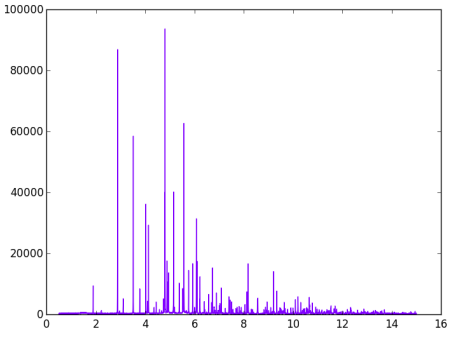
1-29.



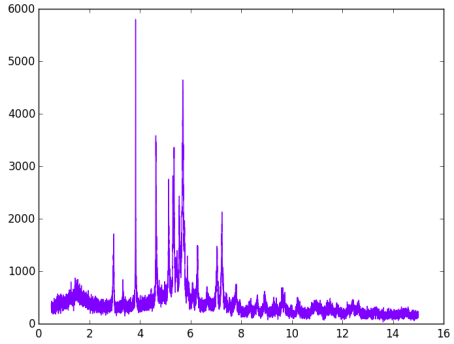
1-30.



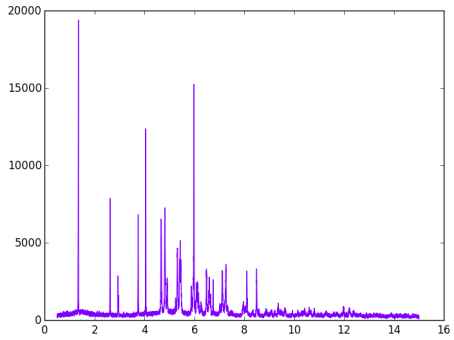
1-31.



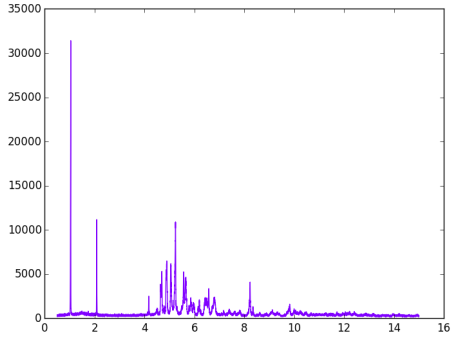
1-32.



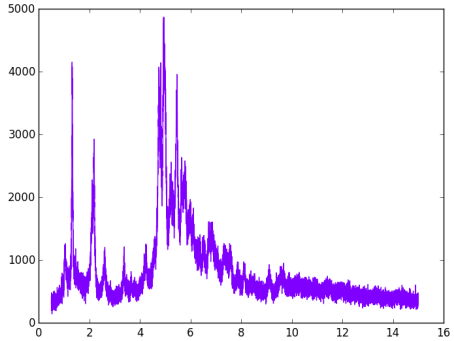
1-33.



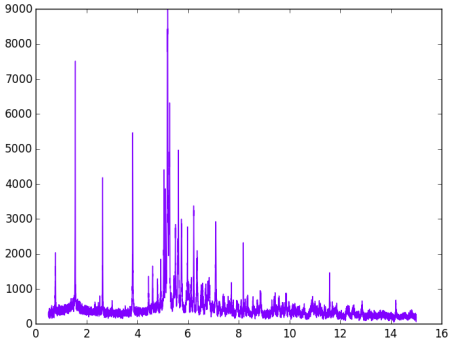
1-34.



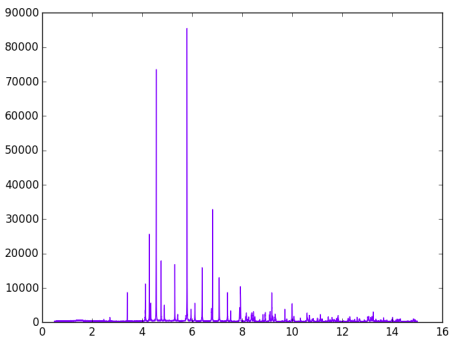
1-35.



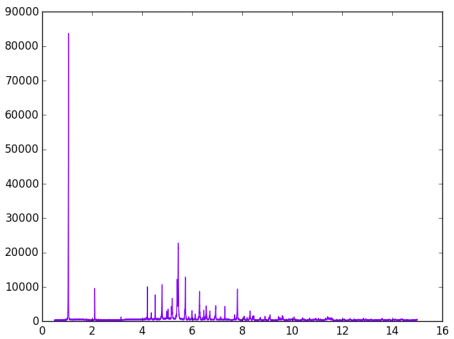
1-36.



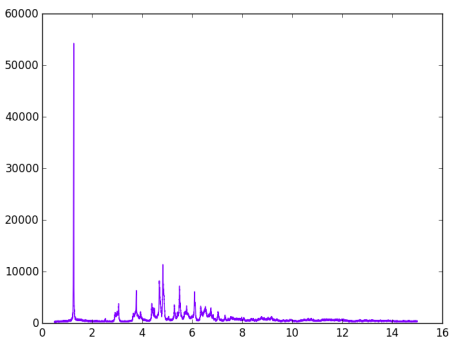
1-37.



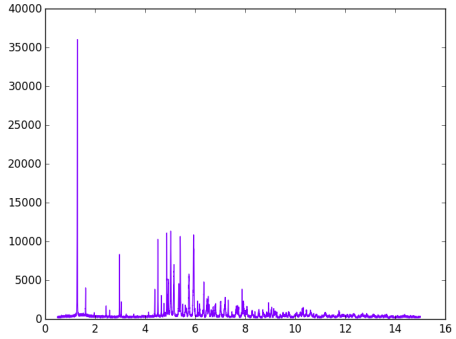
1-38.



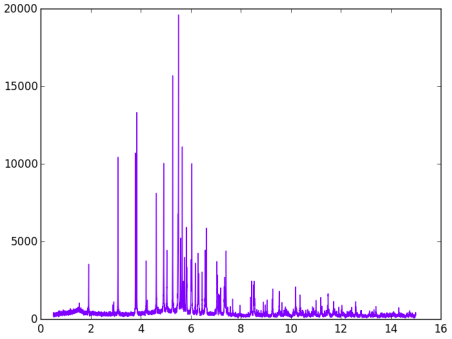
1-39.



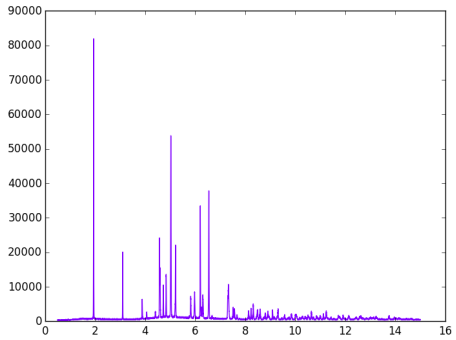
1-40.



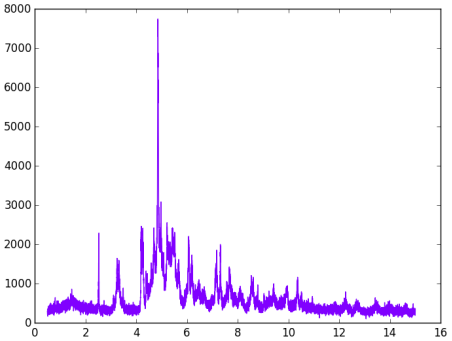
1-41.



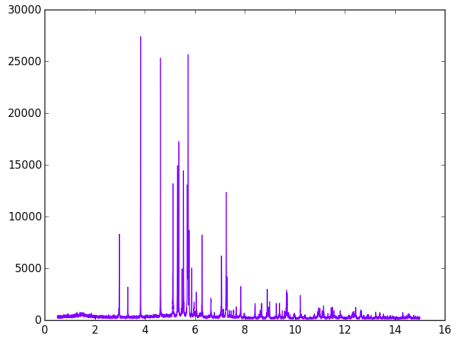
1-45.



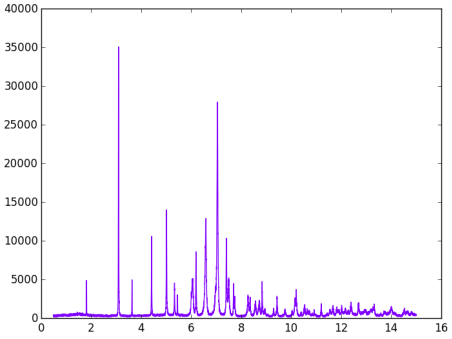
1-42.



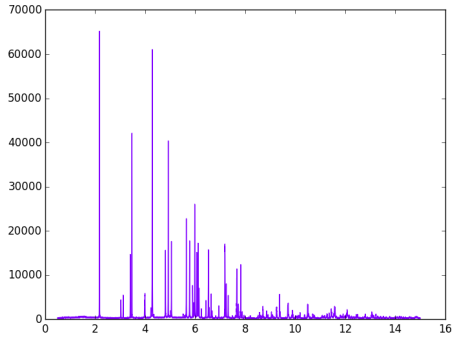
1-47.



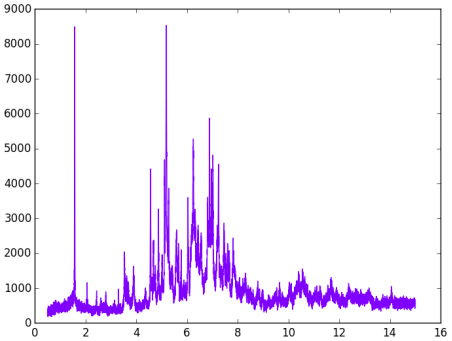
1-43.



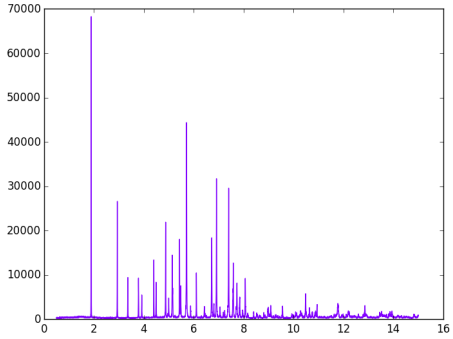
2-1.



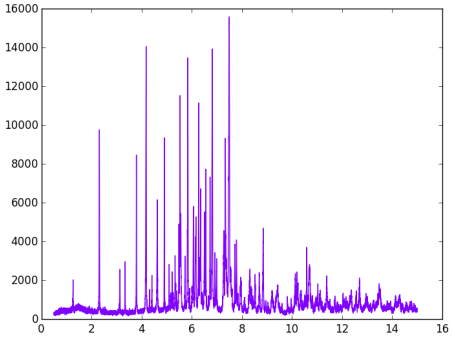
1-44.



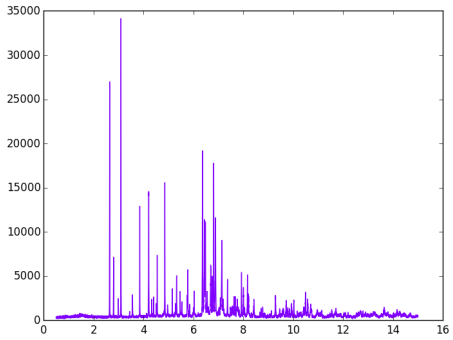
2-2.



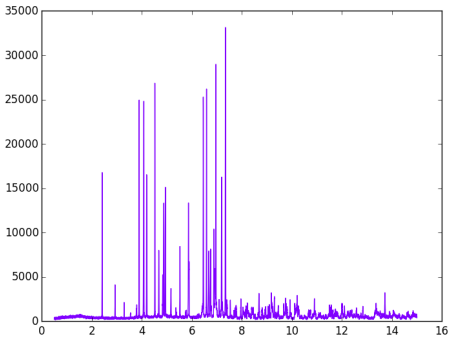
2-3.



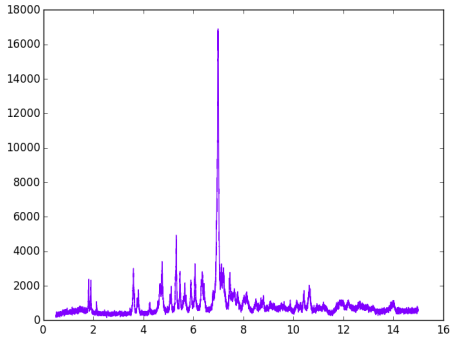
2-8.



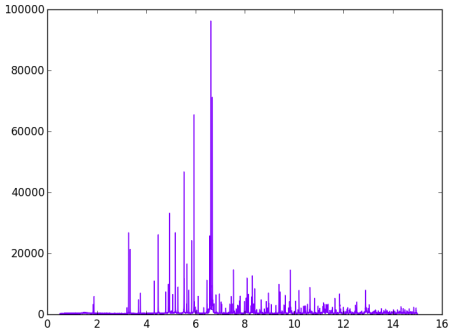
2-4.



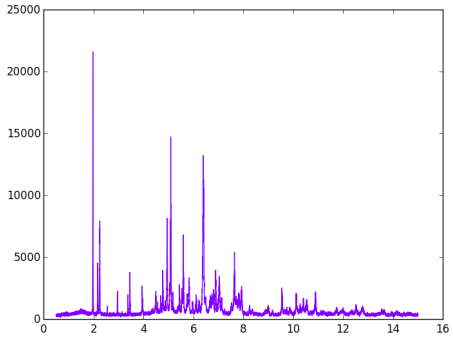
2-9.



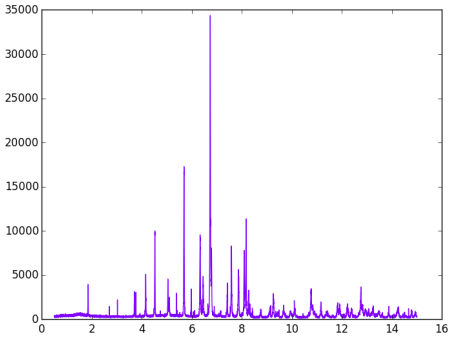
2-5.



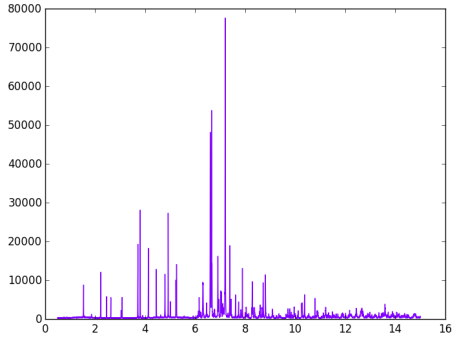
2-10.



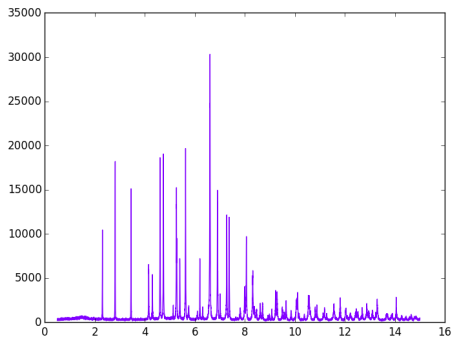
2-7.



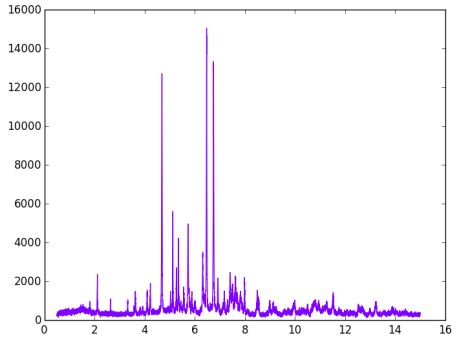
2-11.



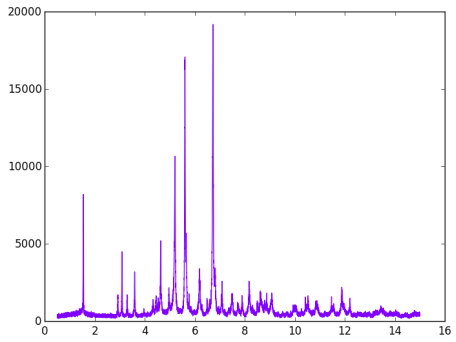
2-12.



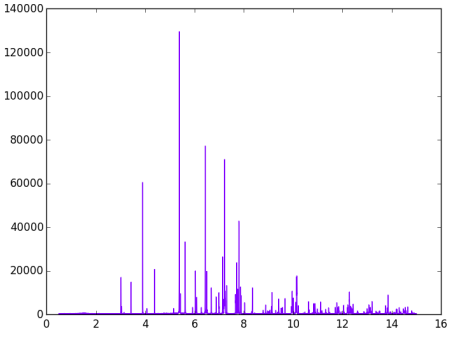
2-13.



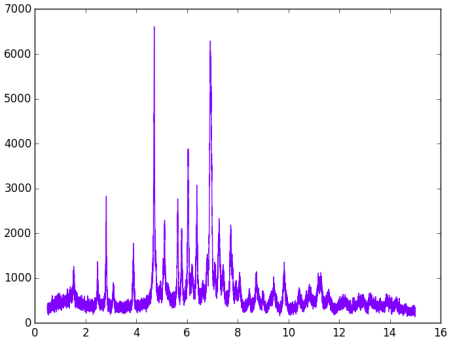
2-14.



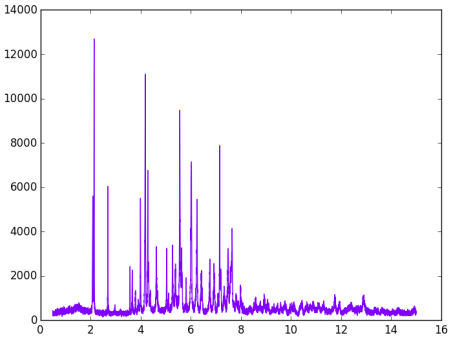
2-15.



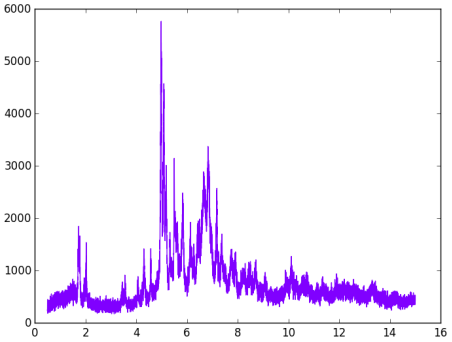
2-16.



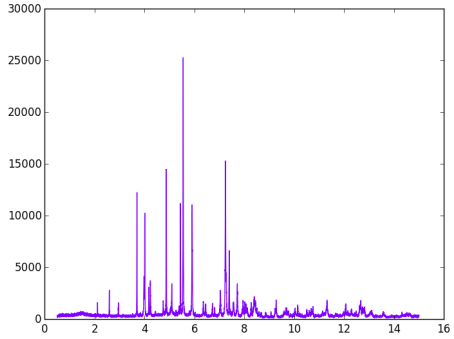
2-17.



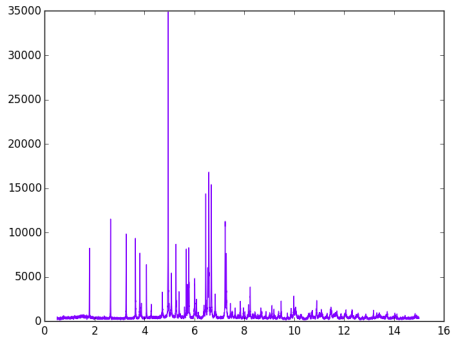
2-18.



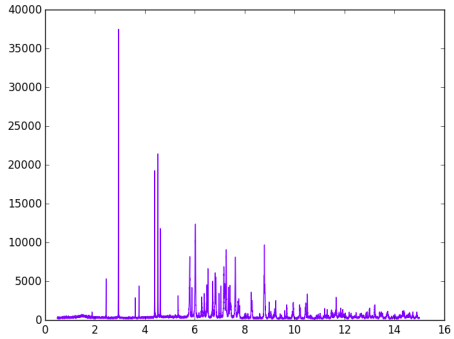
2-19.



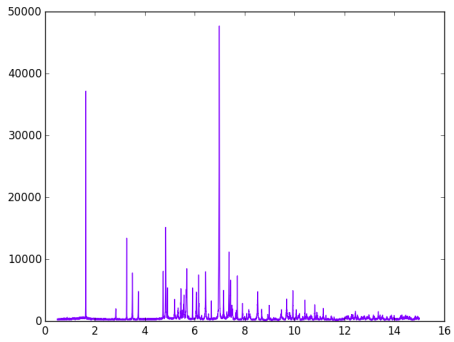
2-20.



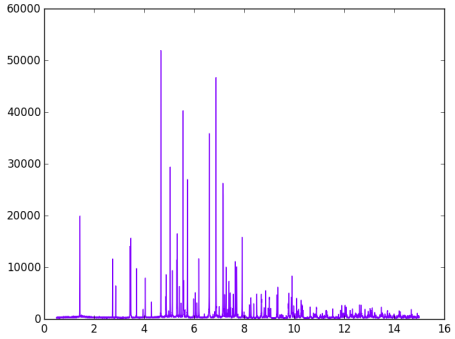
2-21.



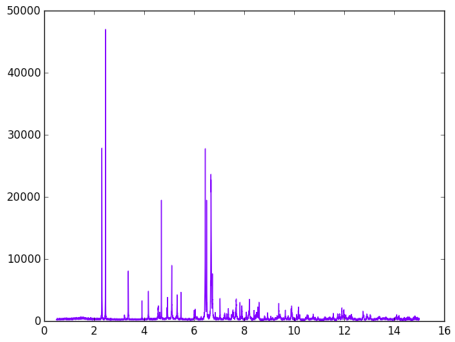
2-22.



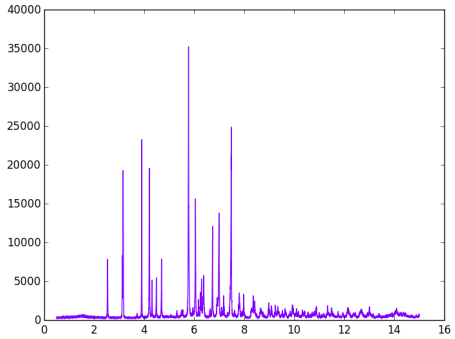
2-23.



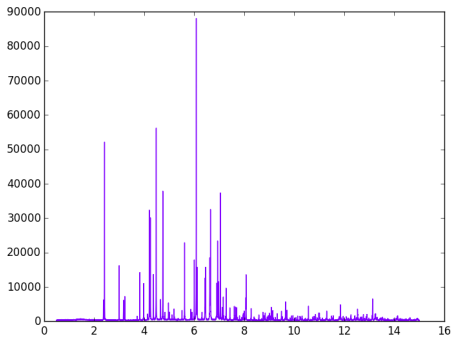
2-24.



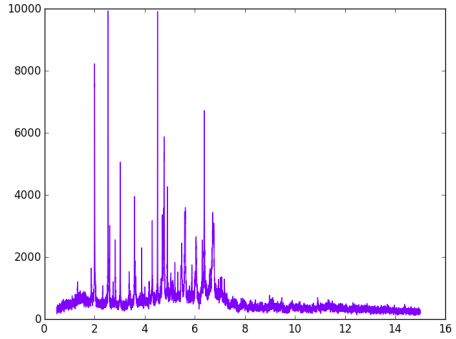
2-25.



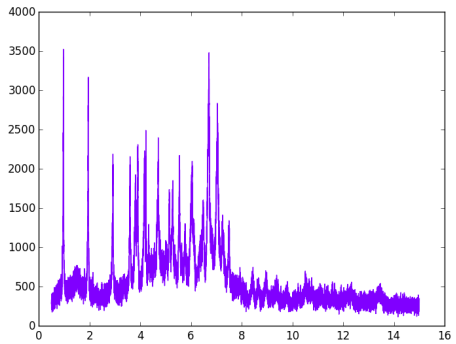
2-26.



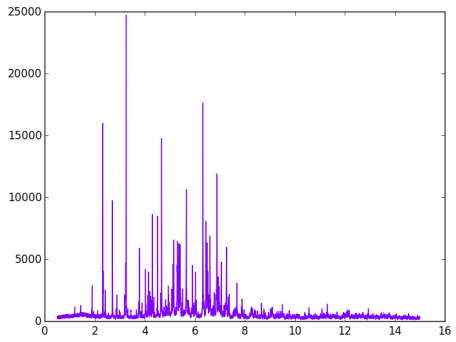
3-1.



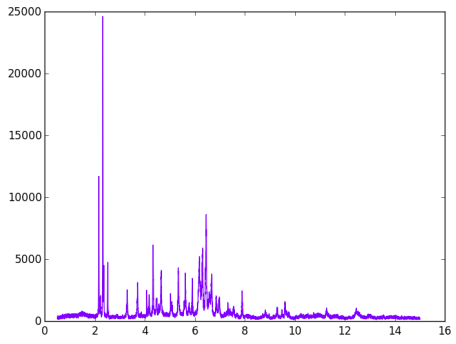
3-2.



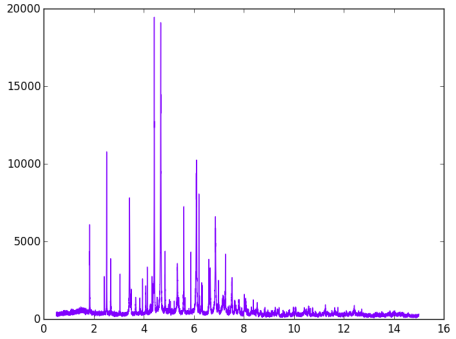
3-3.



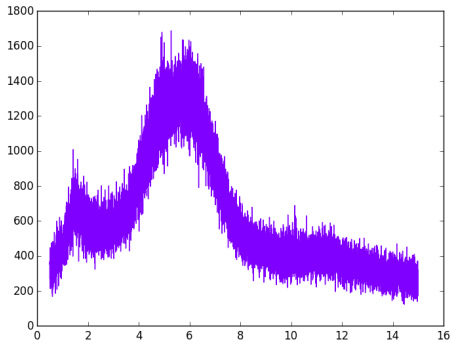
3-4.



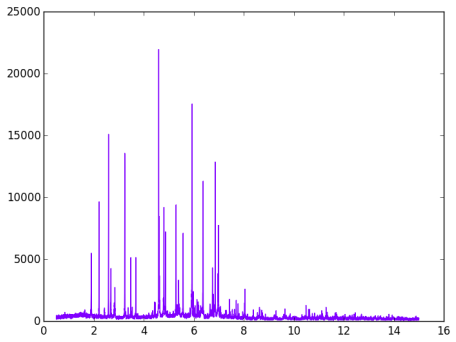
3-5.



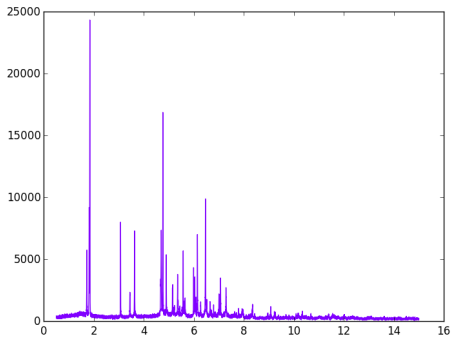
3-6.



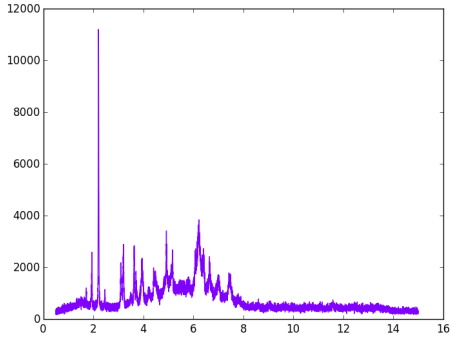
3-7.



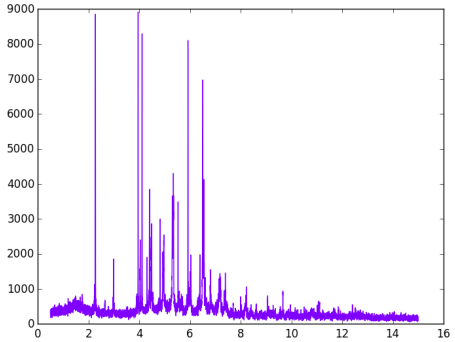
3-8.



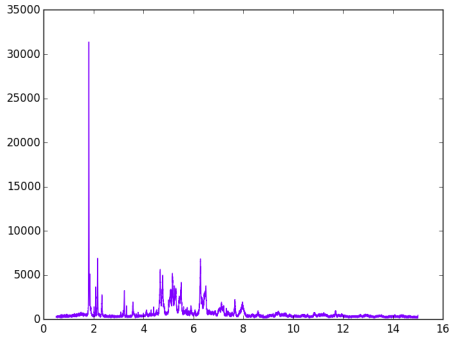
3-9.



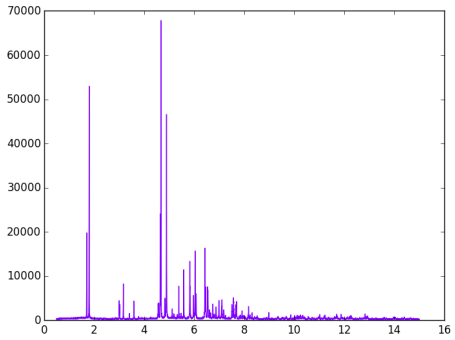
3-10.



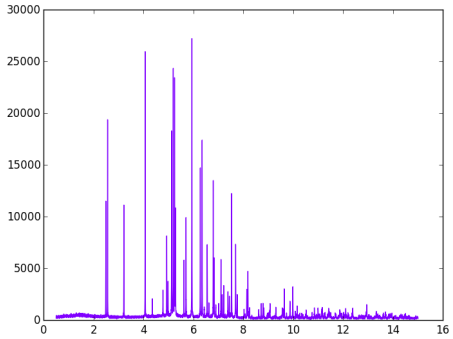
3-14.



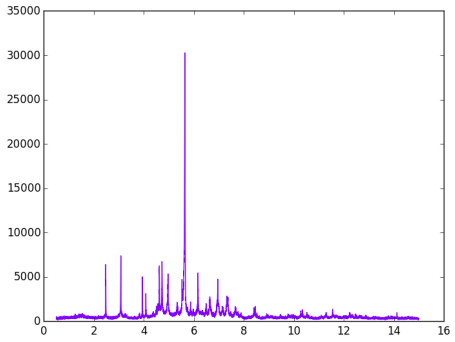
3-11.



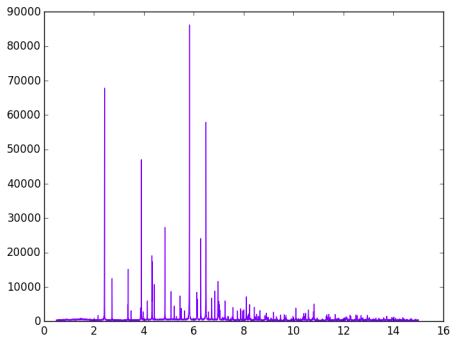
3-15.



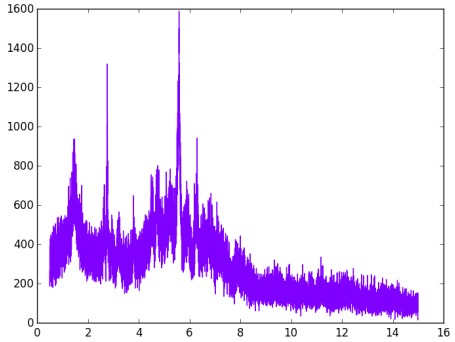
3-12.



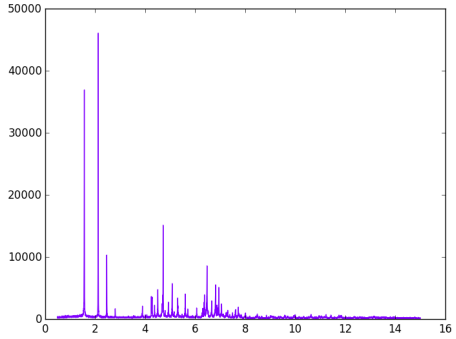
3-16.



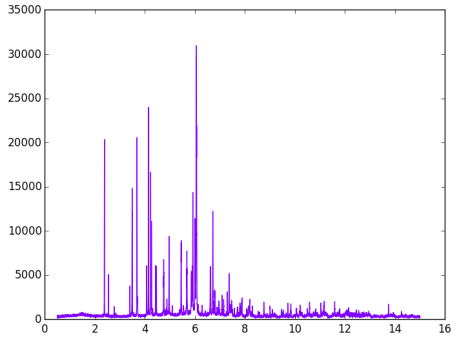
3-13.



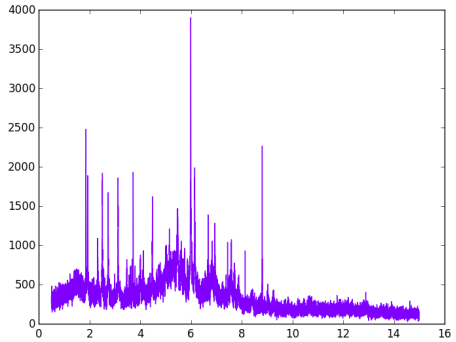
3-17.



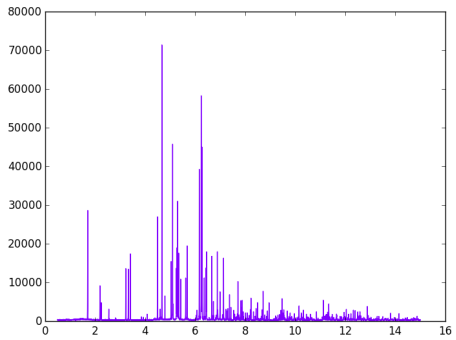
3-18.



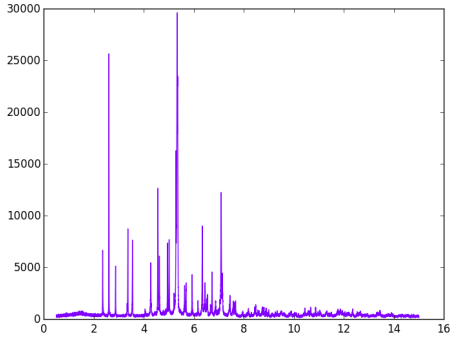
3-19.



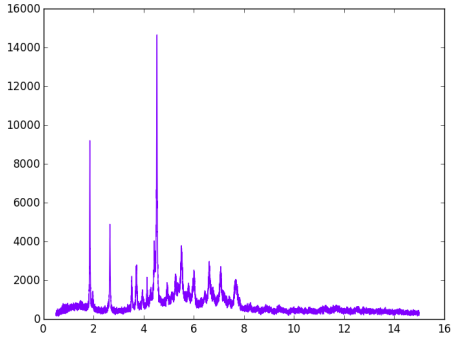
3-20.



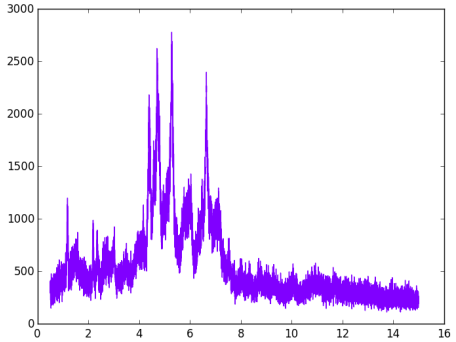
3-21.



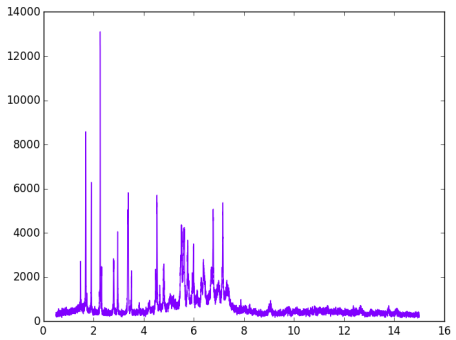
3-22.



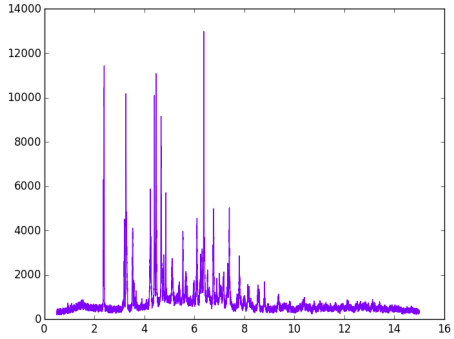
3-23.



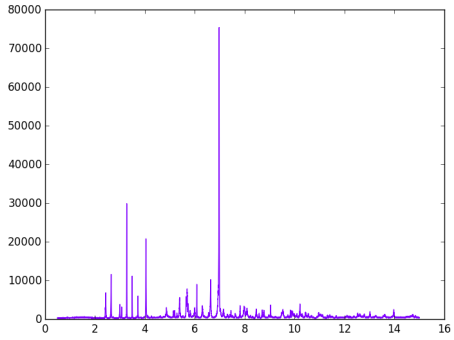
3-24.



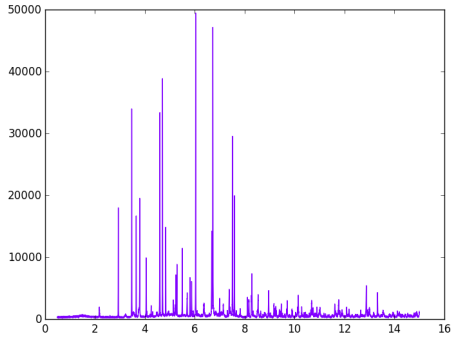
3-25.



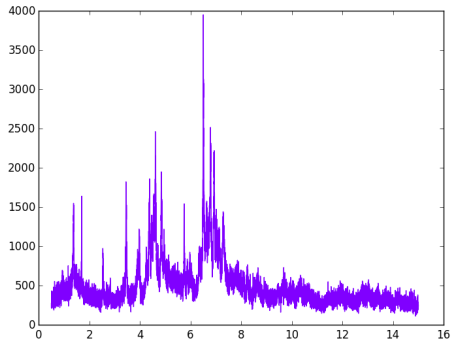
4-1.



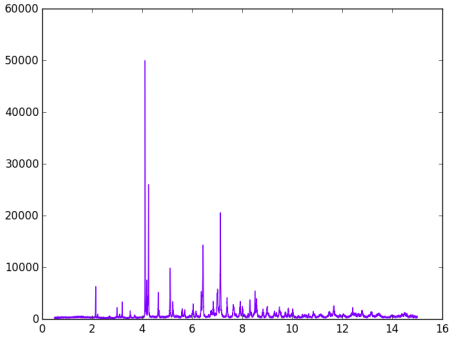
4-2.



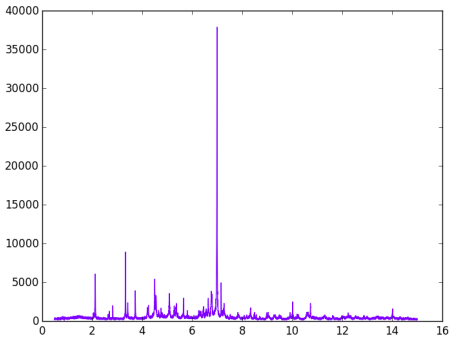
4-3.



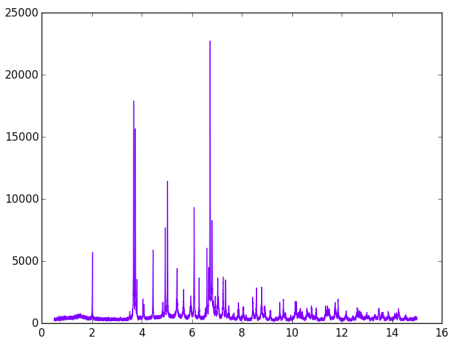
4-4.



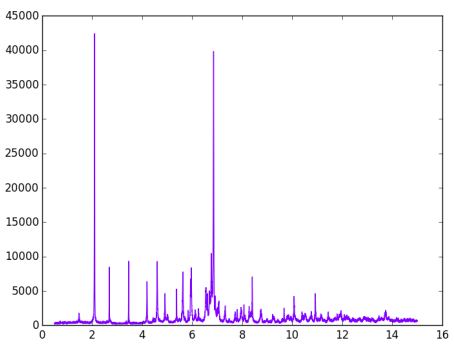
4-5.



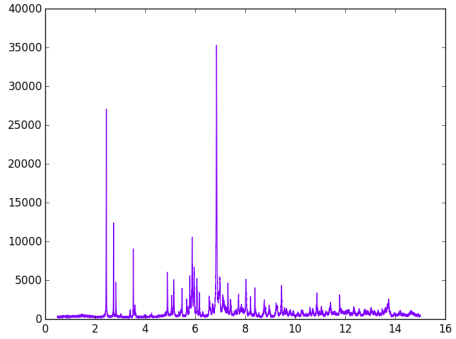
4-6.



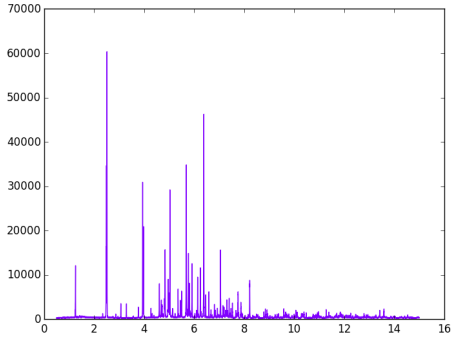
4-7.



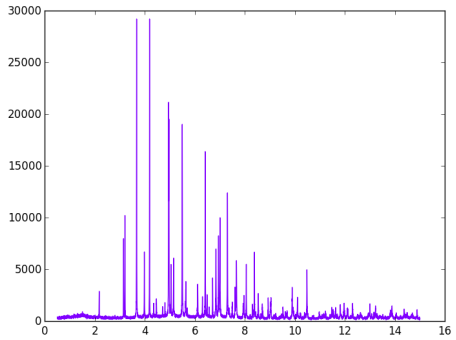
4-8.



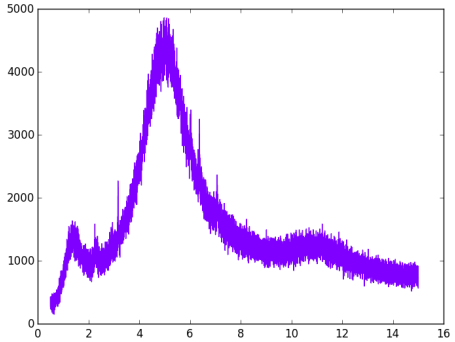
4-9.



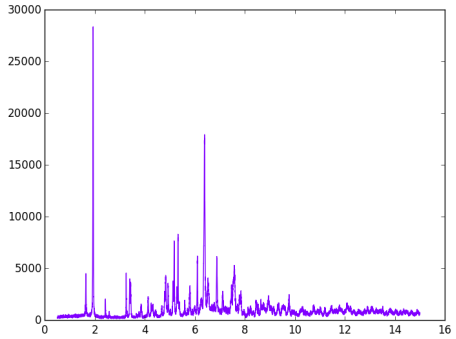
5-3.



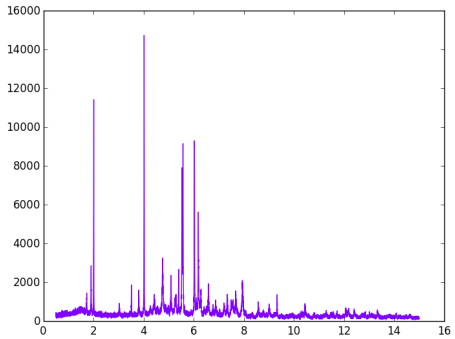
4-10.



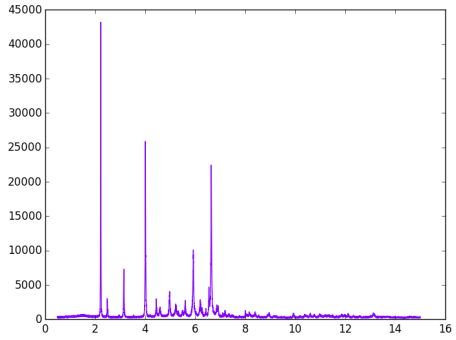
5-4.



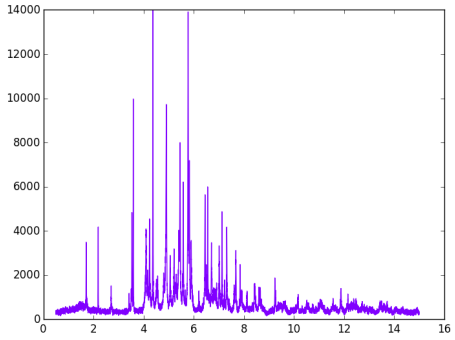
4-11.



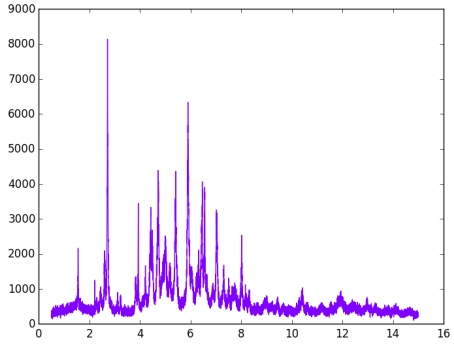
5-5.



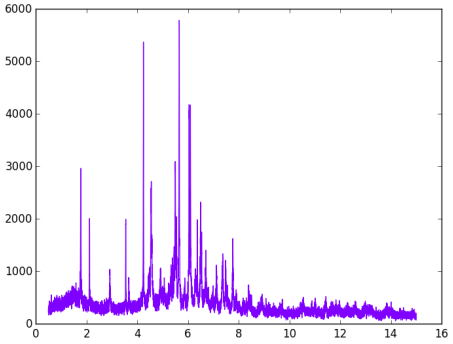
5-2.



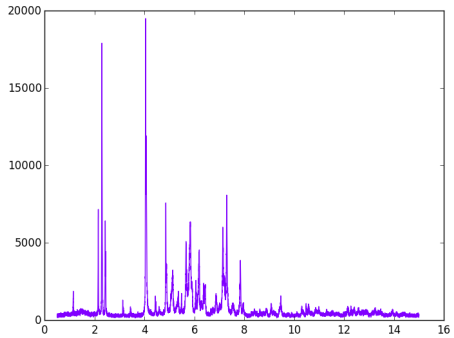
5-6.



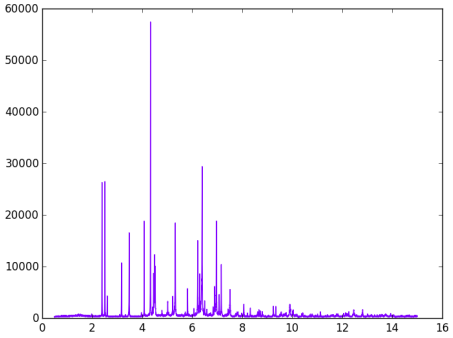
5-7.



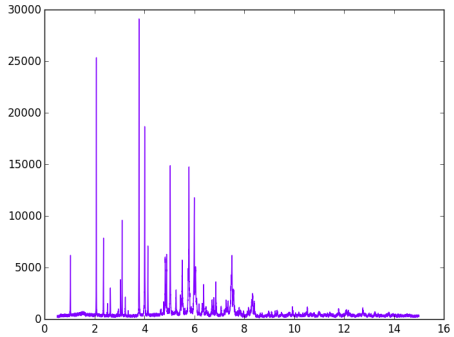
5-11.



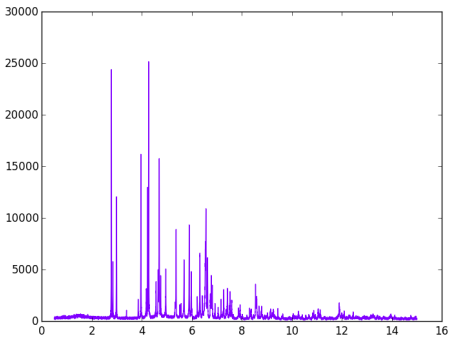
5-8.



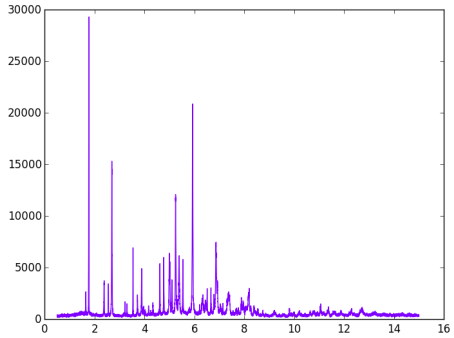
6-1.



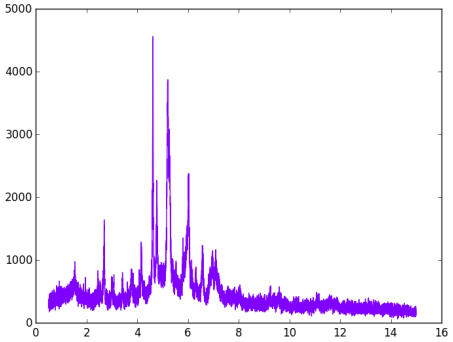
5-9.



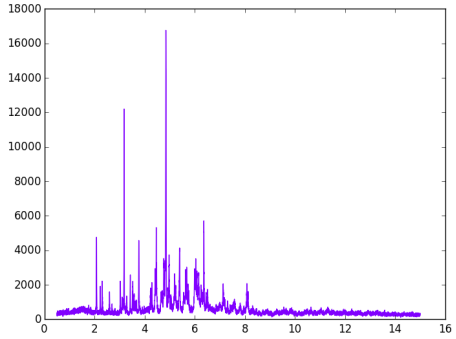
6-2.



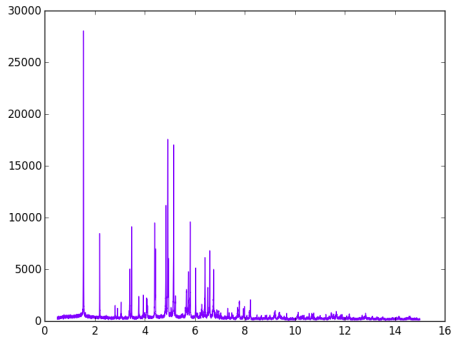
5-10.



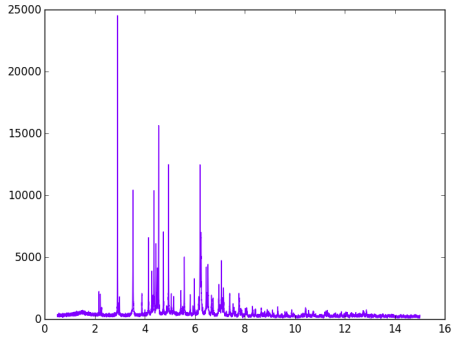
6-3.



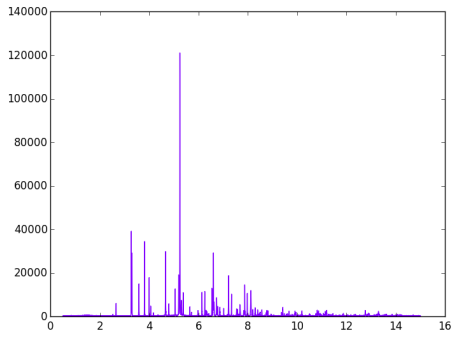
6-4.



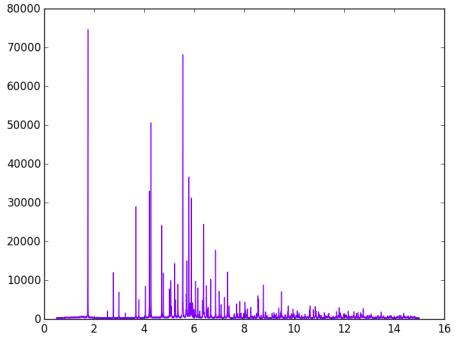
6-5.



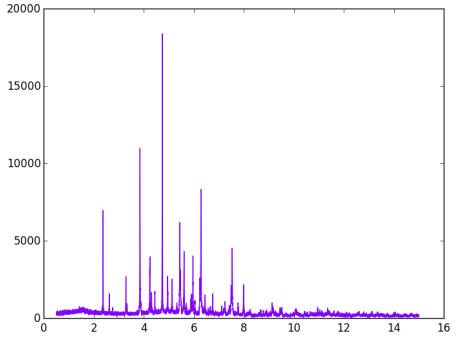
6-6.



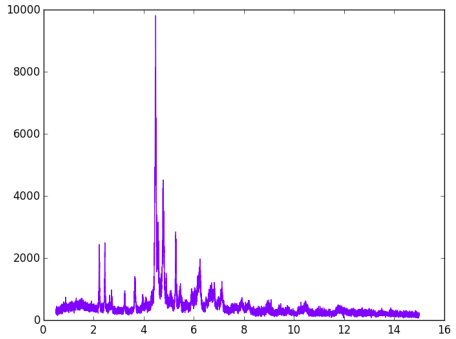
6-7.



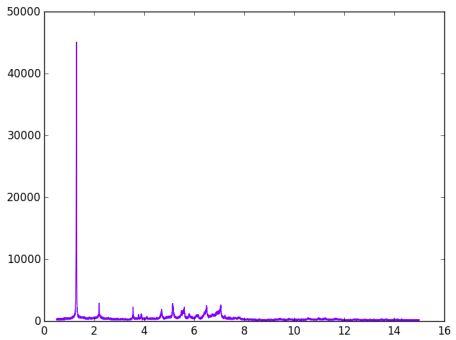
6-8.



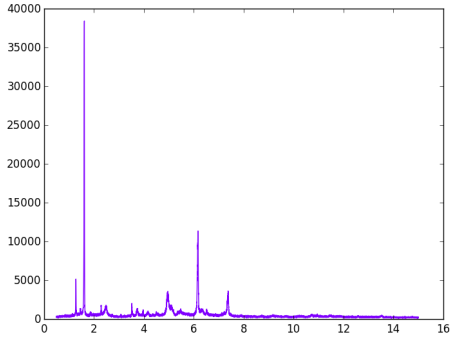
6-9.



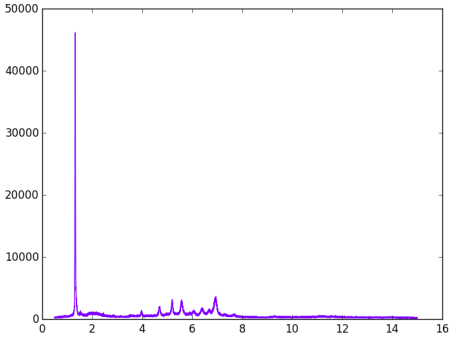
6-10.



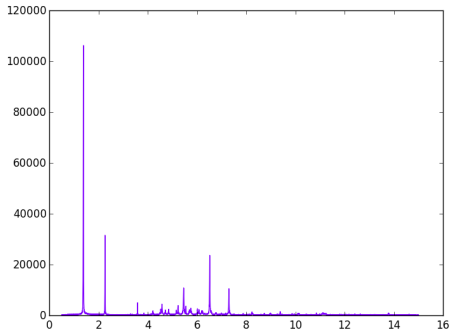
7-1.



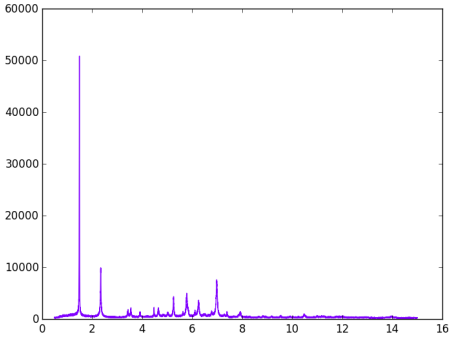
7-2.



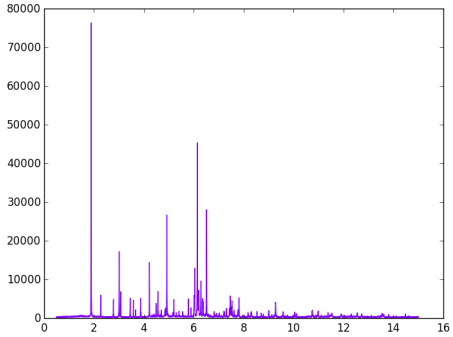
7-6.



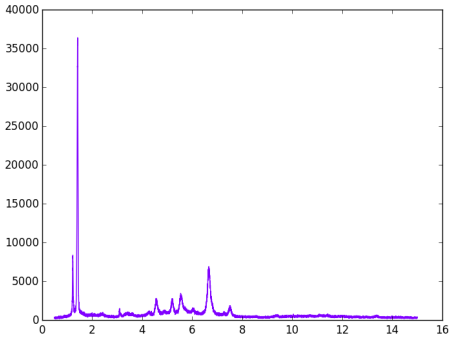
7-3.



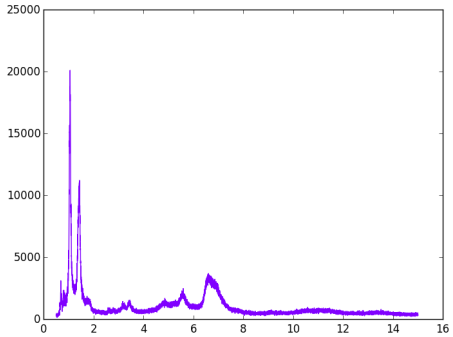
7-7.



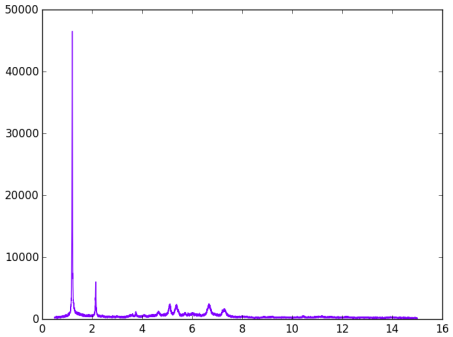
7-4.



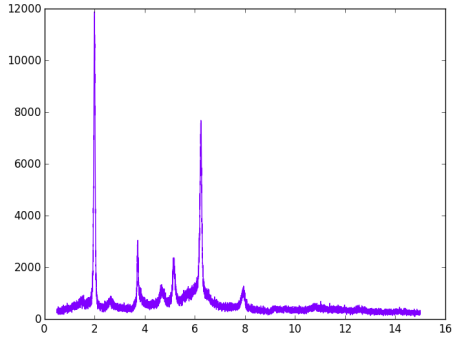
7-8.



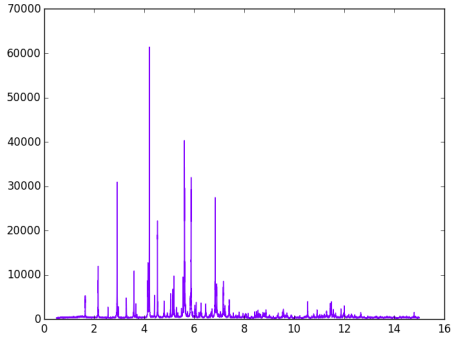
7-5.



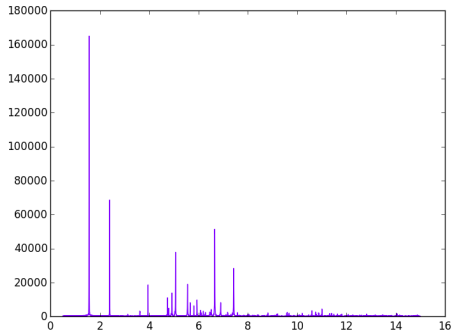
7-9.



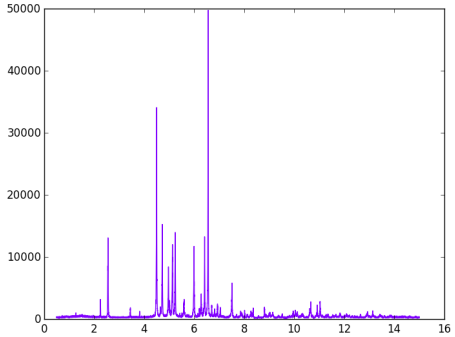
7-10.



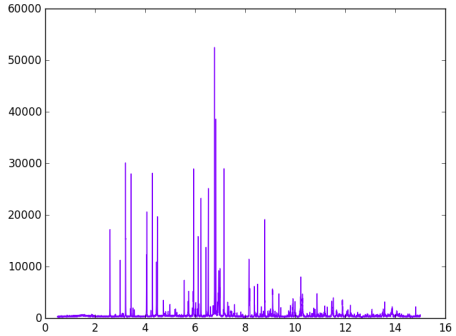
8-1.



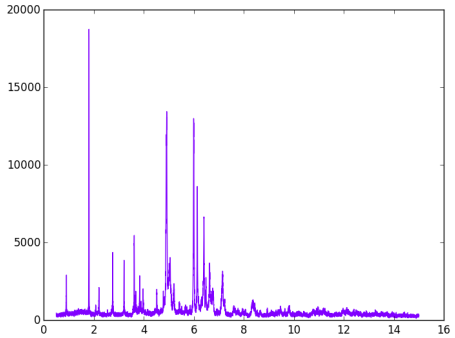
7-11.



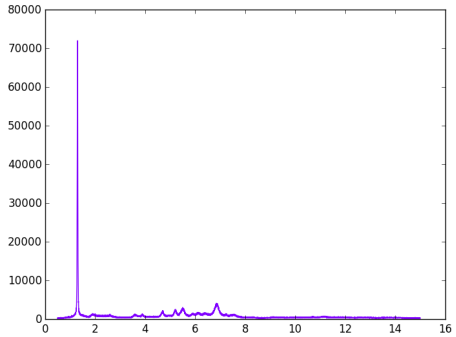
8-2.



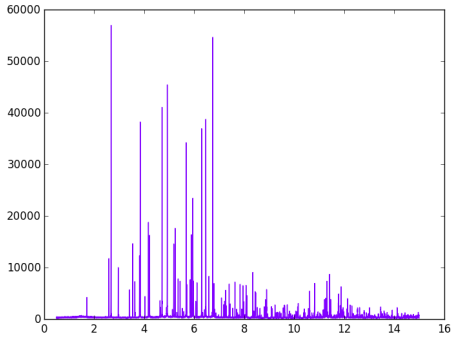
7-12.



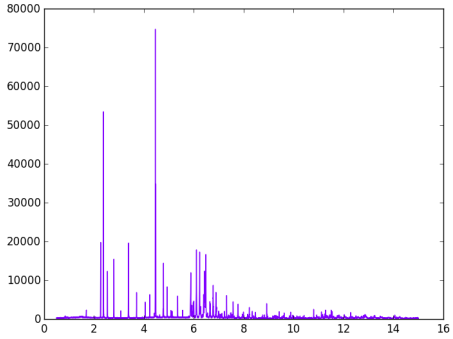
8-3.



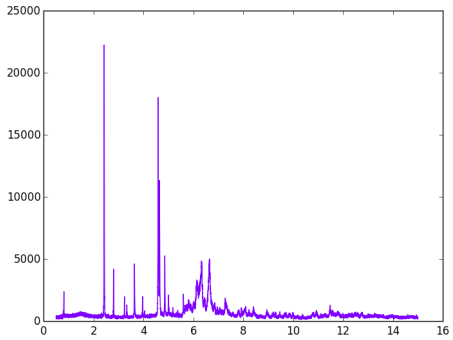
7-13.



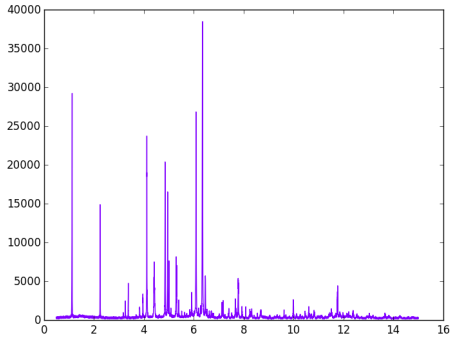
8-4.



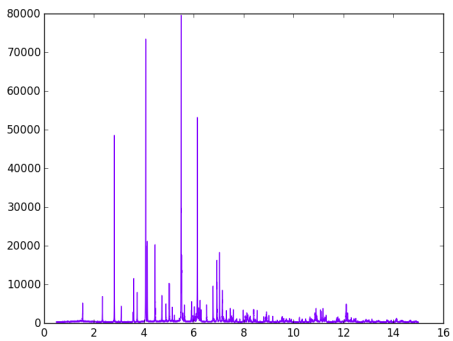
8-5.



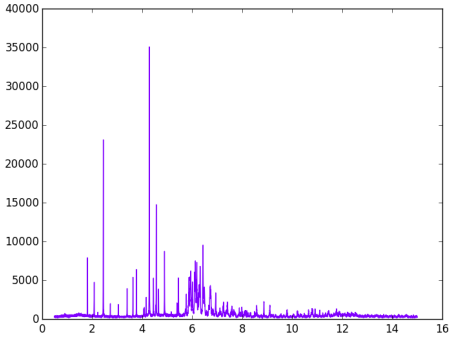
8-6.



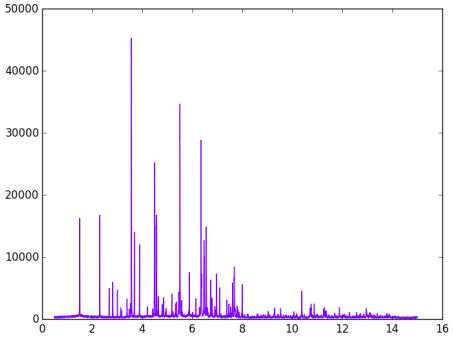
8-7.



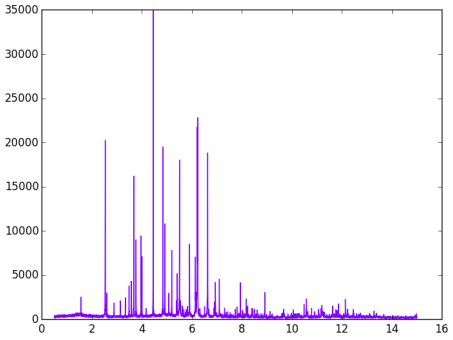
8-8.



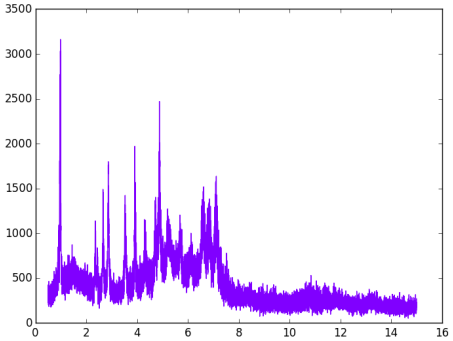
8-9.



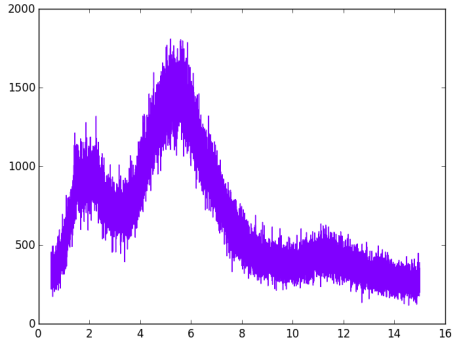
8-10.



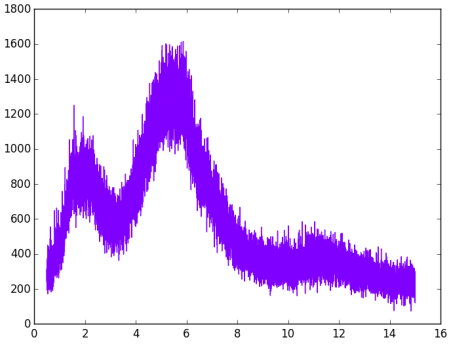
8-11.



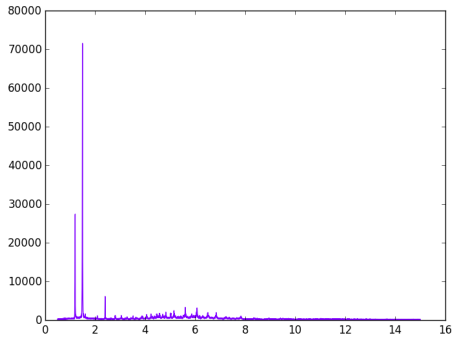
9-1.



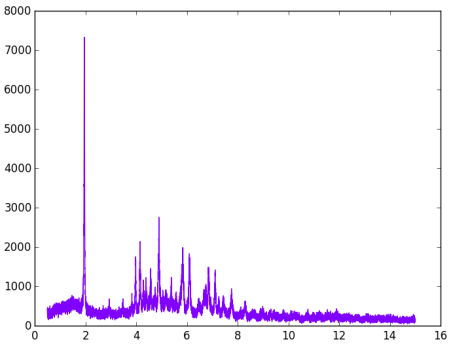
9-2.



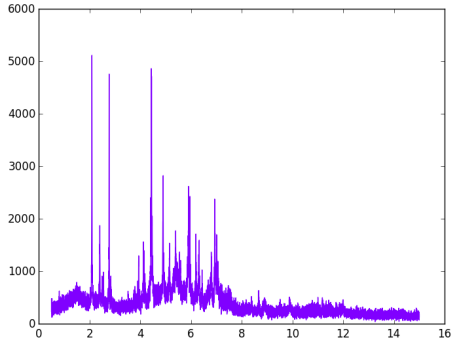
9-8.



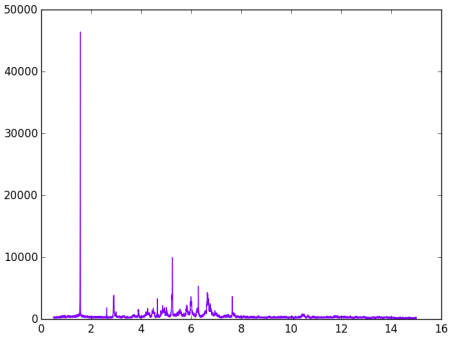
9-3.



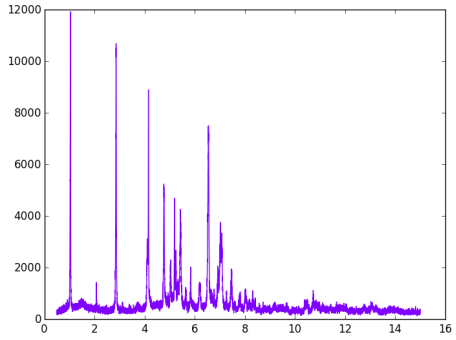
9-9.



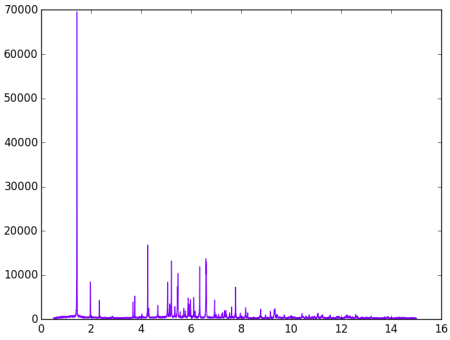
9-5.



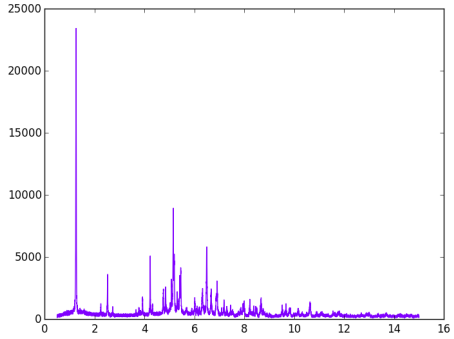
9-10.



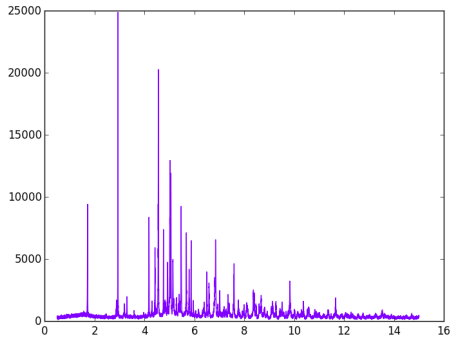
9-6.



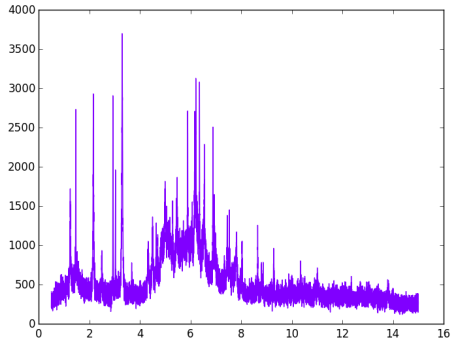
10-1.



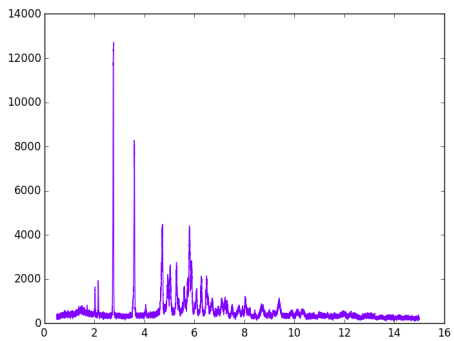
10-2.



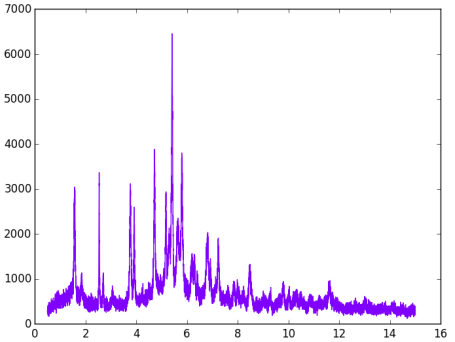
10-3.



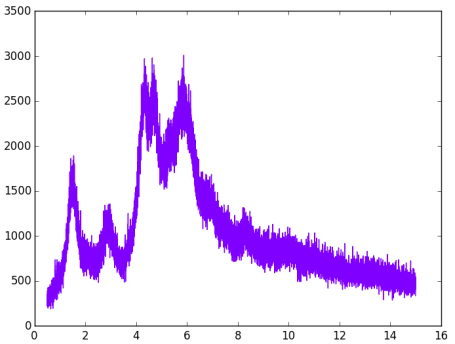
10-4.



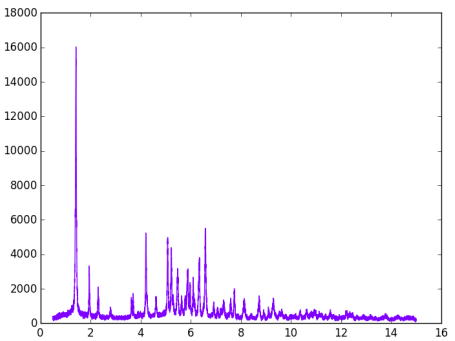
10-5.



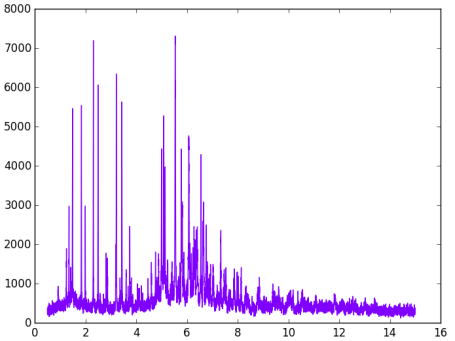
10-6.



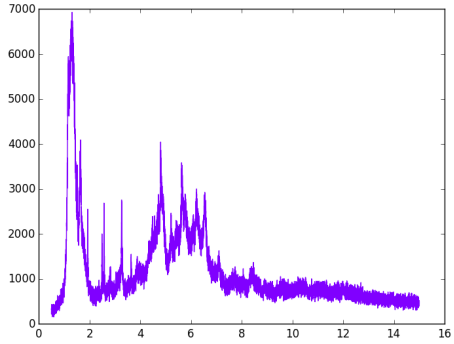
10-7.



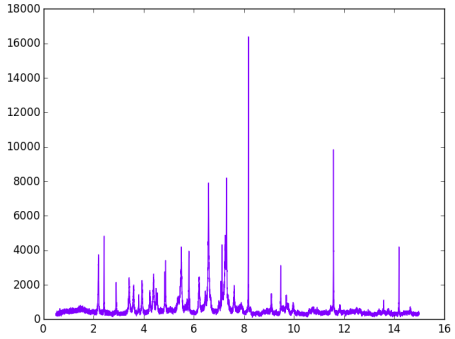
10-8.



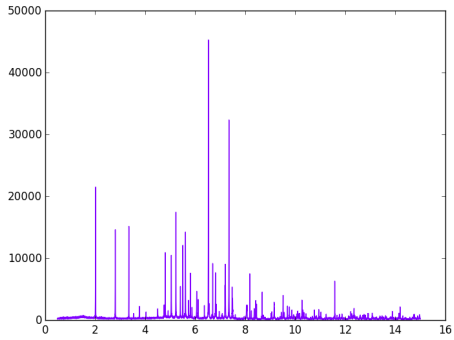
10-9.



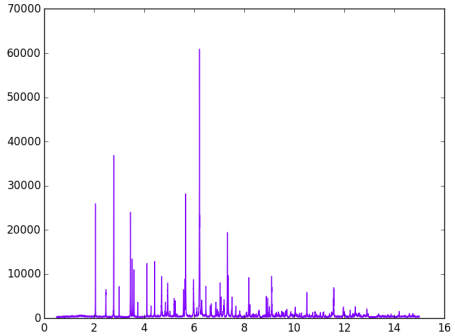
10-10.



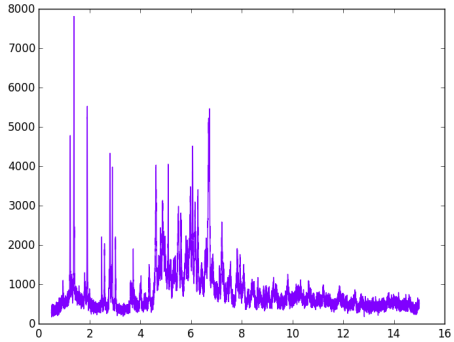
11-4.



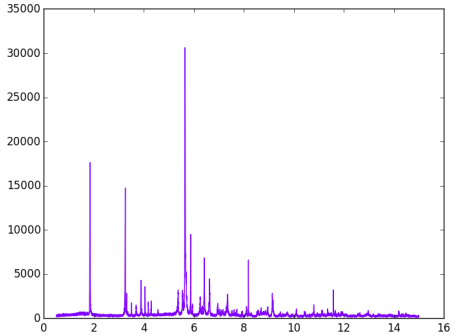
11-1.



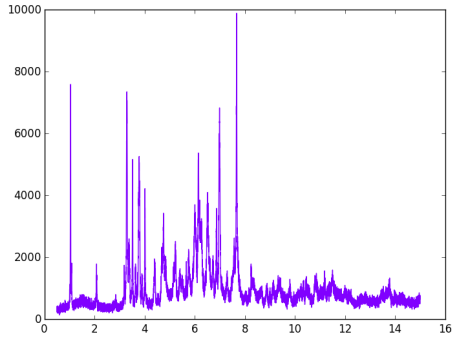
11-5.



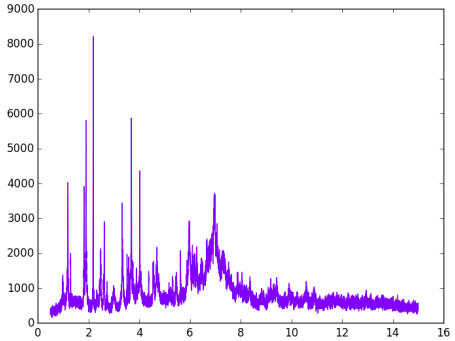
11-2.



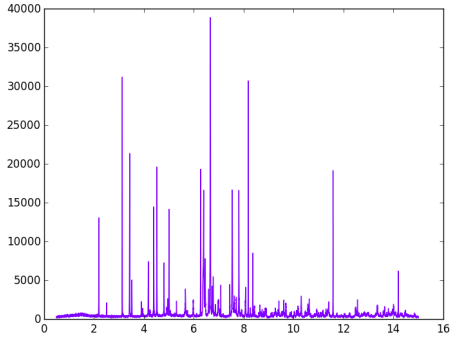
11-6.



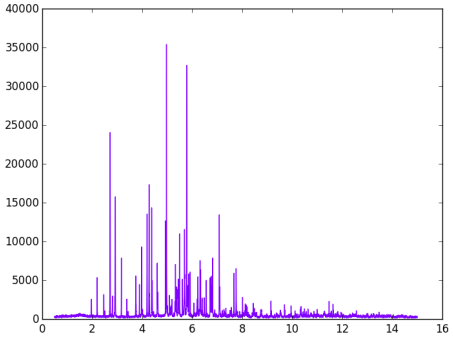
11-3.



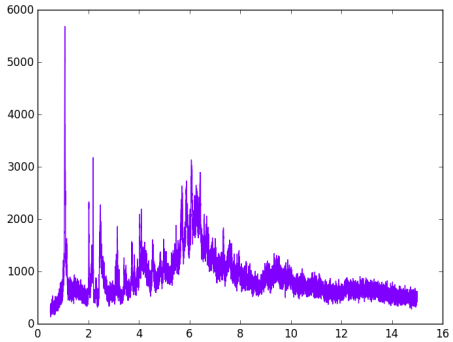
11-7.



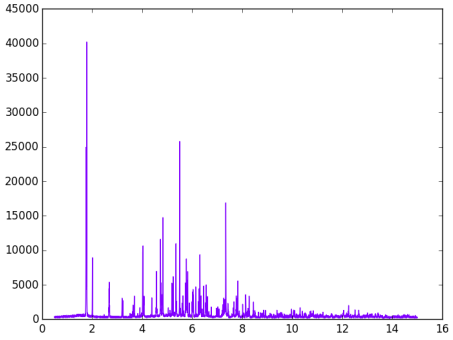
11-8.



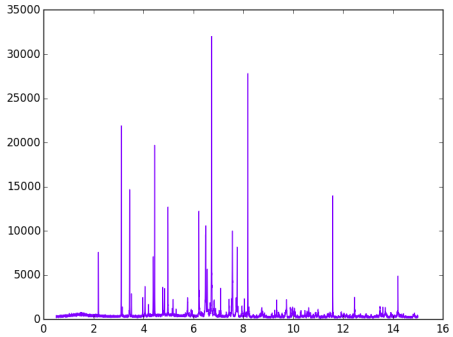
12-2.



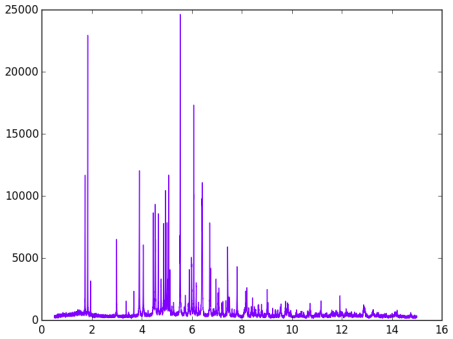
11-9.



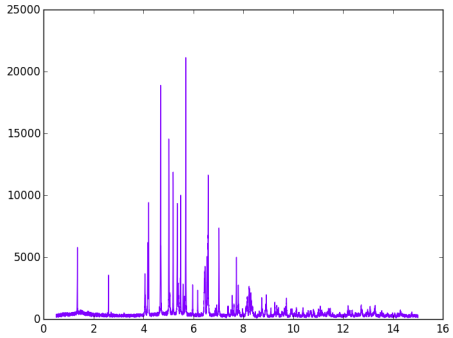
12-3.



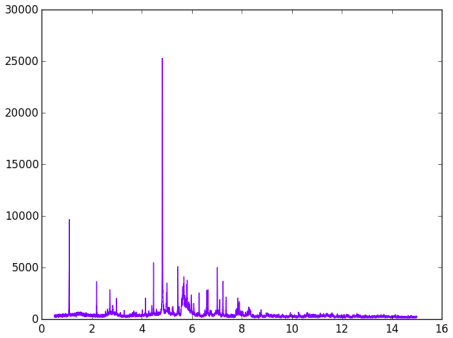
11-10.



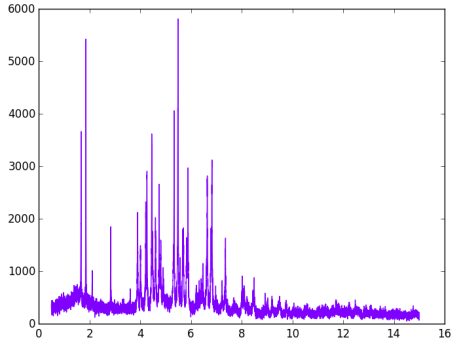
12-5.



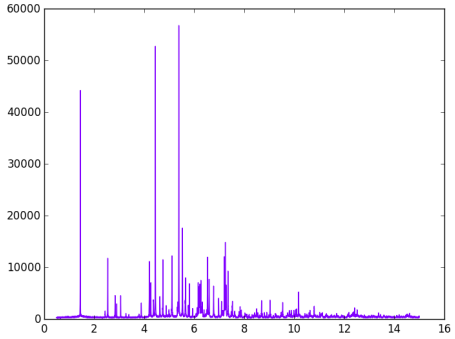
12-1.



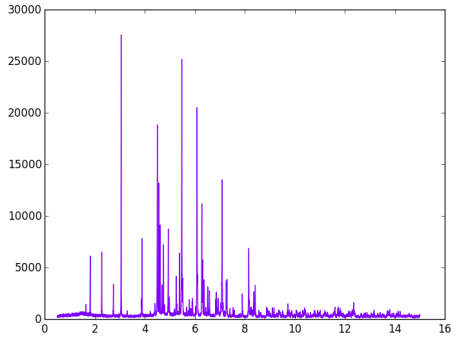
12-6.



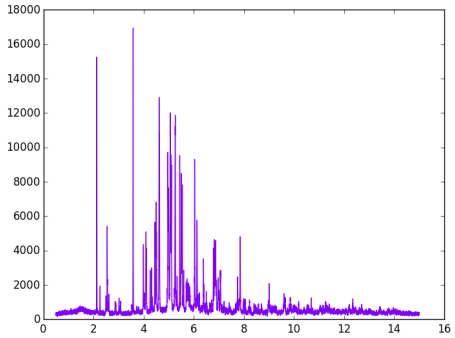
12-7.



12-9.



12-8.



12-10.