



## University of Southampton Research Repository

Copyright © and Moral Rights for this thesis and, where applicable, any accompanying data are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis and the accompanying data cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content of the thesis and accompanying research data (where applicable) must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holder/s.

When referring to this thesis and any accompanying data, full bibliographic details must be given, e.g.

Thesis: Author (Year of Submission) "Full thesis title", University of Southampton, name of the University Faculty or School or Department, PhD Thesis, pagination.

Data: Author (Year) Title. URI [dataset]



**UNIVERSITY OF SOUTHAMPTON**

Faculty of Engineering and Physical Sciences  
School of Chemistry and Chemical Engineering

***Machine Learning Methods for Analysis of  
Organic Molecular Crystal Structure Prediction  
Landscapes***

Volume 1 of 1

**Jennifer Eleanor Martin**  
*MChem*

ORCID ID: 0009-0004-0343-6309

Thesis for the Degree of Doctor of Philosophy

October 2025





# UNIVERSITY OF SOUTHAMPTON

## Abstract

Faculty of Engineering and Physical Sciences  
School of Chemistry and Chemical Engineering

## Doctor of Philosophy

### **Machine Learning Methods for Analysis of Organic Molecular Crystal Structure Prediction Landscapes**

Jennifer Eleanor Martin

This thesis presents work on the analysis of the Crystal Structure Prediction (CSP) landscapes of organic molecules. The work presented here adapted an existing approach to identifying stabilisable crystal structures from prediction sets - the Generalised Convex Hull (GCH) [1] - such that its application to molecular crystal structures was more theoretically reasonable. A new global Smooth Overlap of Atomic Positions (SOAP) kernel to reasonably define the similarity of molecular crystal structures was developed and then used within the GCH approach - which identifies stabilisable crystal structure candidates by using unsupervised machine learning [1]. The results were compared to those from a GCH approach that utilised a simple average SOAP kernel to assess the impact of kernel construction. The new kernel was assessed regarding three key metrics useful to materials discovery: the effectiveness of the GCH in identifying stabilisable candidates, the interpretability of machine learned (ML) descriptors derived from the kernel, and the utility of the kernel in machine learning of energies. Comparisons revealed a complex picture of results - from which a clearly superior kernel for identifying stabilisable structures could not be identified. However, the new kernel construction showed potential promise, particularly in leading to interpretable ML descriptors. Findings highlighted a sensitivity of similarity kernel based landscape analysis methods to kernel construction.

A secondary project developed a proof of concept for performing fast and approximate molecular CSP by formation and optimisation of structural analogues of previously predicted crystal structures of similar molecules. Preliminary results indicated strong potential of the method in predicting the most crucial regions of the CSP landscape with greatly reduced sampling relative to quasi-random CSP approaches. This suggested that the concept is a promising area for further development – with some key areas for improvement being highlighted.

# Contents

Abstract . . . . .	i
List of Figures . . . . .	xxii
List of Tables . . . . .	xxv
List of Accompanying Material . . . . .	xxvi
Declaration of Authorship . . . . .	xxvii
Acknowledgements . . . . .	xxviii
Acronyms . . . . .	xxx
<b>1 Introduction</b>	<b>1</b>
1.1 The Important Definitions . . . . .	1
1.1.1 What is a Molecular Crystal Structure? . . . . .	1
1.1.2 What is Crystal Structure Prediction? . . . . .	5
1.2 Literature and Background . . . . .	6
1.2.1 Where Does Crystal Structure Prediction Stand Today? . . . . .	6
1.2.2 Approaches to Representations and Analysis of Crystal Structure Prediction Landscapes . . . . .	14
1.3 Project Motivations . . . . .	23
1.4 Thesis Overview . . . . .	24
<b>2 Theory and Methods</b>	<b>25</b>
2.1 Overview . . . . .	25
2.2 Crystal Structure Prediction Workflows . . . . .	26
2.2.1 Getting the Molecular Conformation . . . . .	27
2.2.2 Density Functional Theory . . . . .	27
2.2.3 Quasi-random Sampling . . . . .	29
2.2.4 Optimisation of Trial Structures . . . . .	30
2.2.5 Force Fields . . . . .	30
2.2.6 Multipoles . . . . .	31

2.2.7	Summarising the Rigid-CSP Workflow . . . . .	32
2.2.8	Flexible Crystal Structure Prediction . . . . .	33
2.2.9	Sampling Multiple Conformations . . . . .	33
2.2.10	Differences in Initial Optimisations . . . . .	34
2.2.11	Re-optimisation Steps . . . . .	35
2.2.12	Periodic DFT . . . . .	35
2.2.13	DFTB . . . . .	36
2.2.14	MACE . . . . .	36
2.2.15	Duplicate Removal Methods . . . . .	37
2.3	Landscape Analysis Methods . . . . .	39
2.3.1	Overview . . . . .	39
2.3.2	Energetic Ranking . . . . .	39
2.3.3	Convex Hull Methods . . . . .	39
2.3.4	Defining the Convex Hull . . . . .	40
2.3.5	Vertices, Facets and Equations . . . . .	42
2.3.6	The Chemically Relevant Convex Hull . . . . .	43
2.3.7	Dressed Energies . . . . .	43
2.3.8	Candidate Pools . . . . .	44
2.4	Generalised Convex Hull Methods . . . . .	45
2.4.1	Overview . . . . .	45
2.4.2	The Generalised Convex Hull Workflow . . . . .	45
2.4.3	Smooth Overlap of Atomic Positions (SOAP) Descriptors . . . . .	46
2.4.4	SOAP kernels . . . . .	48
2.4.5	Kernel Principal Component Analysis . . . . .	49
2.4.6	Constructing and Using the Hull . . . . .	51
2.5	Supervised Machine Learning Methods . . . . .	55
2.5.1	Overview . . . . .	55
2.5.2	Support Vector Classification . . . . .	55
2.5.3	Gaussian Process Regression . . . . .	57
2.6	Additional Tools Used in this Work . . . . .	59
2.6.1	Overview . . . . .	59
2.6.2	ASE . . . . .	59
2.6.3	Librascal . . . . .	59
2.6.4	PYMATGEN . . . . .	59

2.6.5	Molecular Graphs . . . . .	59
2.6.6	CSD . . . . .	60
2.6.7	Mercury . . . . .	60
2.6.8	CSD API . . . . .	60
<b>3</b>	<b>Constructing a Kernel Suited to Molecular Crystals</b>	<b>61</b>
3.1	Overview . . . . .	61
3.2	The Problem with Conventional Kernels . . . . .	62
3.3	Adaptation Concept . . . . .	66
3.3.1	Overview . . . . .	66
3.3.2	Crystals of Asymmetric Molecules . . . . .	66
3.3.3	Accounting for Symmetry . . . . .	69
3.3.4	Why Average the Possibilities? . . . . .	75
3.3.5	Summarising the Construction . . . . .	76
3.4	Implementation . . . . .	78
3.4.1	Overview . . . . .	78
3.4.2	Re-indexing . . . . .	78
3.4.3	Accounting for Symmetry . . . . .	79
3.4.4	Making the Kernel . . . . .	80
3.4.5	Gathering Kernels . . . . .	82
3.4.6	Summarising the Implementation . . . . .	83
3.4.7	Forming a Cohesive Codebase . . . . .	83
3.4.8	Facing Large Datasets . . . . .	84
3.5	Considerations and Concerns . . . . .	85
3.5.1	Overview . . . . .	85
3.5.2	The Asymmetry of Possible Mapping Kernels . . . . .	85
3.5.3	The Importance of Local Symmetry . . . . .	86
3.5.4	Ideality . . . . .	87
3.5.5	Computational Costs . . . . .	88
3.5.6	Choosing an ‘Original’ Kernel Construction . . . . .	88
3.6	Concluding Remarks . . . . .	90
<b>4</b>	<b>Crystal Structure Prediction</b>	<b>91</b>
4.1	Overview . . . . .	91
4.2	Crystal Structure Prediction of Semi-conductors . . . . .	92

4.3	Crystal Structure Prediction of Primidone	96
4.3.1	Overview	96
4.3.2	Structure Prediction Process	96
4.3.3	Restoring Symmetry	103
4.3.4	Results	104
4.3.5	Uncharacterised Polymorphs	106
4.4	Crystal Structure Prediction of DAP	108
4.5	Crystal Structure Prediction of CL-20	111
4.5.1	Overview	111
4.5.2	Unsuccessful CL-20 CSP	111
4.5.3	Successful CL-20 CSP	113
4.6	Duplicate Removal Investigations	118
4.6.1	Overview	118
4.6.2	Approach	118
4.6.3	Results	120
4.7	Concluding Remarks	125
<b>5</b>	<b>Comparing Landscape Analysis Methods</b>	<b>126</b>
5.1	Overview	126
5.2	The Rationale of Comparison	127
5.2.1	The Need for Method Selection	127
5.2.2	Metrics and Comparisons	128
5.3	Energy Cut-offs and Energy-Density Hulls	133
5.3.1	Overview	133
5.3.2	Results	133
5.4	Comparing Implemented Kernels	136
5.4.1	Overview	136
5.4.2	Simple Comparisons	136
5.4.3	Intrinsic Dimensionality	139
5.4.4	Polymorph Rankings and Candidate Pools	143
5.4.5	Accounting for Energetic Uncertainty	153
5.4.6	The Impact of Energetic Calculation Methods	159
5.5	Comparing Kernel Based Methods to Traditional Approaches	162
5.5.1	Overview	162
5.5.2	Polymorph Rankings and Candidate Pools	162

5.6	Concerns and Considerations . . . . .	167
5.6.1	Reasonable Constructions . . . . .	167
5.6.2	Centering . . . . .	168
5.7	Concluding Remarks . . . . .	169
<b>6</b>	<b>Gathering Meaning from the Kernel</b>	<b>170</b>
6.1	Overview . . . . .	170
6.2	Intuitive Descriptors . . . . .	172
6.2.1	Overview . . . . .	172
6.2.2	Density . . . . .	172
6.2.3	Molecular Conformation . . . . .	172
6.2.4	Hydrogen Bonding . . . . .	172
6.2.5	Unsuccessful Intuitive Descriptors . . . . .	173
6.3	Relationships to kPCA Components . . . . .	174
6.3.1	Overview . . . . .	174
6.3.2	ROY . . . . .	174
6.3.3	Galunisertib . . . . .	178
6.3.4	Porous Systems . . . . .	185
6.4	Investigating Relationship Strength . . . . .	188
6.4.1	Overview . . . . .	188
6.4.2	Impact of SOAP Cut-Off . . . . .	188
6.4.3	Systematically Comparing Kernels . . . . .	191
6.4.4	Summarising Investigation of Relationship Strengths . . . . .	199
6.5	Concluding Remarks . . . . .	200
<b>7</b>	<b>Machine Learning of Energies</b>	<b>201</b>
7.1	Overview . . . . .	201
7.2	Approach . . . . .	202
7.3	Initial Testing . . . . .	204
7.3.1	Datasets . . . . .	204
7.3.2	Training and Validation Process . . . . .	205
7.3.3	Initial Results . . . . .	206
7.3.4	Investigating Chlorpropamide . . . . .	210
7.4	Extended Testing . . . . .	212
7.4.1	Overview . . . . .	212

7.4.2	Dataset	212
7.4.3	Pre-Processing	213
7.4.4	Training and Validation Process	215
7.4.5	Results	217
7.5	Concluding Remarks	225
<b>8</b>	<b>Templating CSP for Similar Molecules</b>	<b>226</b>
8.1	Overview	226
8.2	Key Definitions	228
8.3	Choosing Appropriate Systems	229
8.3.1	Overview	229
8.3.2	Substituted Molecules	229
8.3.3	Differently-Sized Analogues	230
8.4	Generating Analogous Crystal Structures	232
8.4.1	Approach	232
8.4.2	Optimisation	239
8.5	Evaluation of Templating CSP	241
8.5.1	Tested Approaches	241
8.5.2	Metrics	242
8.5.3	Recovery of Known Crystal Structures	243
8.5.4	Recovery Percentages	248
8.5.5	Efficiency Ratios	263
8.5.6	Distribution of Matches	269
8.6	How Broadly Can Templating Be Applied?	274
8.7	Considerations and Concerns	277
8.7.1	Overview	277
8.7.2	Templates with $Z' > 1$	277
8.7.3	Flexible Molecules	277
8.7.4	Substructure Limitations	277
8.7.5	Defining the Analogue	280
8.7.6	Optimisation	281
8.8	Concluding Remarks	283
<b>9</b>	<b>Conclusions and Future Work</b>	<b>285</b>
9.1	Overview	285

9.2	Work Summary . . . . .	286
9.3	Issues . . . . .	289
9.4	Potential for Future Work . . . . .	290
<b>APPENDICES</b>		<b>292</b>
<b>A</b>	<b>SOAP Parameters</b>	<b>292</b>
A.1	Parameters used in SOAP descriptor calculations . . . . .	292
<b>B</b>	<b>Key Scripts</b>	<b>293</b>
B.1	Script to Enforce Consistent Atom Indexing in Structure Sets . . . . .	293
B.2	Script to Calculate Adapted Kernel for Systems with Rigid Underlying Molecules	295
B.3	Script to Calculate the Direct Product of Symmetry Mappings . . . . .	313
<b>References</b>		<b>320</b>



# List of Figures

1.1	Some examples of conceptual crystal structures -regular patterns of a repeated motif	1
1.2	Simple periodic array of points and corresponding real-world example - the simple cubic crystal structure of elemental polonium [3]	2
1.3	Conceptual demonstration of symmetry within a unit cell. A 2D crystal structure generated from a ‘unit cell’ of crescent objects related by a mirror plane and an example representing the same concept albeit with a molecule - forming a hypothetical 2D crystal structure of the molecule ROY	3
1.4	An example of a predicted molecular crystal structure,here of molecule DAP, and its corresponding unit cell	4
1.5	A conceptual CSP workflow: The knowledge of the connectivity of a molecule is fed through a series of computations which output many predicted crystal structures	5
1.6	Demonstration of a simplified structure-energy landscape defined by one degree of freedom. Blue circles represent trial structures - with red arrows indicating the basin (local minimum) into which the corresponding structure will relax during the geometry optimisation stage	7
1.7	Conceptual demonstration of packing and conformational polymorphism. The ‘molecule’ here is represented by a blue polygon,with different shapes representing different conformations of the same molecule	10
1.8	Examples of some of the target molecules used in each of the CSP Blind Tests [17, 39–44]. Numbers in each box indicate the test from which the target molecules are taken	12
1.9	An example of an energy-density plot to represent the CSP landscape of molecule NTCD. Each point represents a single predicted crystal structure	16

1.10	A hypothetical inorganic convex hull constructed on a landscape of composition and energy. Red lines indicate the hull - with orange points representing the hull vertices. The hull vertices indicate stable structures and the corresponding stable compositions. Dark blue spots represent the most stable crystal structure for a given unstable composition . . . . .	17
1.11	An example of an Energy Structure Function map - representing the stability and gas storage capacity of predicted crystal structures of molecule T2. Known stable polymorphs are labelled. Image adapted with permission from ref [10] . . . . .	18
1.12	Demonstration of a hypothetical dataset with lower intrinsic dimensionality than full dimensionality. The left hand diagram indicates the dataset - a series of points spanning three dimensions. The right hand image reveals the lower intrinsic dimensionality - as all points lie almost within a 2D plane. Despite variance across all three dimensions - the original data could be restructured to be represented in just two dimensions, by noting its position within the indicated plane . . . . .	19
1.13	Example of the use of clustering to analyse CSP landscapes. The left hand image shows a CSP landscape, coloured by the cluster to which algorithms have assigned the points. The right hand image shows the same landscape coloured by the packing class of the structure. It can be seen that there is a correspondence between the two. Particularly, the $\gamma$ and herringbone classes correspond to identified clusters - demonstrating the potential of clustering algorithms to identify meaningful automated groupings of structures. The 2D mapping here was generated by dimensionality reduction using the sketch-map algorithm [61]. Image adapted with permission of the Royal Society of Chemistry from ref [62] . . . . .	20
2.1	Simple flowchart of the basic process used in Rigid-CSP . . . . .	27
2.2	Summary of the Rigid-CSP workflow . . . . .	33
2.3	Summary of the Flexible-CSP workflow. Differences to the Rigid-CSP approach are highlighted in red . . . . .	33
2.4	A hypothetical example of a convex hull used in materials discovery. Each point represents a predicted crystal structure, and the orange lines indicate the hull. The meta-stability of a predicted structures is given by its height above the hull . . . .	40
2.5	Graphic of the quickhull algorithm for a two-dimensional set of points, based upon an iterative process of defining triangles with 'extreme' vertices - and thereby selecting hull points (red) and non-hull points (grey) . . . . .	41

2.6	Examples of important attributes of the convex hull - as they relate to the hull on a 2D set of points . . . . .	42
2.7	Diagram showing how dressed energies are calculated for a hypothetical hull using energy (vertical axis) and one structural descriptor. The equations, shown by dashed lines, corresponding to the facets of the hull are taken, the distances from a given point to these planes are calculated and the minima of the distances are taken. Colour coding matches the measured distance to the corresponding facet . . . . .	44
2.8	Workflow used in the GCH approach . . . . .	45
2.9	Conceptual graphic of analysis of a structure via SOAP descriptors. Red circle represents the selected local environment of the central atom - and all neighbouring atoms within the environment have been decorated with a Gaussian function . . . . .	46
2.10	Conceptual example of 'new descriptors' that may be found via PCA approaches. The red arrow indicates a new descriptor, capturing much of the variance of the original dataset . . . . .	50
2.11	Example of a 'triangle' formed of non-hull and hull points . . . . .	52
2.12	Example of a separating hyperplane in a simple SVC model . . . . .	55
2.13	Examples of different possible separating hyperplanes to separate classes in a simple SVC model . . . . .	56
2.14	Example of a separating hyperplane in a simple SVC model and the corresponding margins - the distance from the hyperplane to the nearest data point . . . . .	56
2.15	Example of a set of points of two classes that is not perfectly linearly separable . . . . .	57
3.1	Example of two molecular crystal structures and hypothetical atom-atom comparisons between them. The green arrow signifies a reasonable comparison, while the red gives an example of an unreasonable comparison. . . . .	63
3.2	Examples of combinations of atom-atom comparisons that would be trialled in calculating the global best-match or ReMatch kernels. One example a) is a theoretically reasonable combination while the other, b), is not. Note in practice, these comparisons would likely be carried out across the entire unit cell, and not just asymmetric units . . . . .	64
3.3	Example of two predicted crystal structures of ROY [55]. The circled atoms represent examples of analogous atoms between the two structures . . . . .	66
3.4	Example of a ROY molecule - numbered according to its indexing as derived from a structure file formed during predictions [55]. The numbers represent the determined 'molecular atom indices'. . . . .	67

3.5	Example construction of the adapted kernel for $Z'=1$ crystals of asymmetrical molecules . . . . .	68
3.6	The molecule-molecule comparisons included in the final kernel construction when comparing crystal structures with $Z'>1$ using the a) averaging and b)best-match schemes. . . . .	68
3.7	Hypothetical crystal structures A, B, and B'. B and B' are identical, however a molecular point group operator has transformed the asymmetric unit of B - forming B'. The arrows indicate a specific atom-atom comparison (index 24- index 24) that would be included in $K(A,B)$ and $K(A,B')$ demonstrating the difference in the local environment comparisons that would be included. . . . .	71
3.8	Hypothetical 3-atom molecule within crystal structures. Coloured ellipses behind each atom represent the surrounding local environment. The molecule in one of the structures has undergone transformation by a molecular point group operator prior to crystal structure comparison. In the case of a) and b) this is the operation $C_3$ , and in c) it is the inverse operation $C_3^2$ .The local environment comparisons that would be induced by comparing atoms of matching indices is shown in each case. This demonstrates that whilst the cases of a) and b) differ - c) is equivalent to b). Therefore both cases a) and b) can be covered simply by considering all transformations of the left-hand structure, including the inverse operations. . . . .	73
3.9	Example adapted kernel construction in the case of $Z'=1$ crystal structures of a symmetrical molecule. a) shows the symmetry mappings relevant to the molecule, b) shows the atom-atom comparisons included in the corresponding possible mapping kernels, and c) indicates how this information is combined to form the final kernel. . . . .	74
3.10	An example of the possible combinations of molecular point group operators for which kernels would need to be generated in the case of three operators and a largest $Z'$ of two - showing that the number of kernels to create to cover all possibilities grows quickly. . . . .	75
3.11	Flowchart indicating the conceptual process for constructing the adapted kernel .	77
3.12	Conceptual demonstration of a full kernel matrix formed by concatenation of $Z'$ pair sub-matrices. . . . .	82
3.13	Flowchart of the full process of constructing the adapted kernel. . . . .	83

3.14	Hypothetical example of a set of possible mapping kernels for a system with three relevant mappings. Here green would represent the identity mappings and the blue and red mappings would be the inverse of one another. . . . .	86
4.1	Structures of the four semiconductor-type molecules . . . . .	92
4.2	Example CSP landscape, here of predicted crystal structures of PTCDA. Matches found to the experimental $\alpha$ and $\beta$ polymorphs are shown. . . . .	94
4.3	Overlay of 30-molecule clusters of the experimental structure of MeNTCDI (DAHMUX)(element-Colour) and the predicted global minimum (green) . . . . .	95
4.4	Molecular structure of primidone . . . . .	96
4.5	Primidone molecule with important flexible torsions indicated . . . . .	97
4.6	Enter Caption . . . . .	97
4.7	Conformers of primidone extracted from torsional scans with torsions indicated - (heterocyclic ring conformation, phenyl torsion angle, ethyl torsion angle) . . . .	98
4.8	Conformational energy surfaces of primidone with <i>a</i> , <i>b</i> , and <i>c</i> heterocyclic ring conformations and MOLDIS conformational sampling shown as overlaid points. Colours of points separate the original conformers on each surface from which a distorted conformation was created, with the original conformers indicated by rings of the same colour . . . . .	100
4.9	Comparison of a) pXRD pattern of uncharacterised form C of primidone - digitized from Figure 7b in ref [144] and b) simulated pXRD pattern of a predicted crystal structure . . . . .	106
4.10	Possible tautomers of DAP (a) and the tautomer used for CSP work, being present in both experimental HOF structures (b) . . . . .	108
4.11	Overlays between predicted crystal structures of DAP (green) and the known experimental forms of a) DAP-HOF-1 and b) DAP-HOF-2 (element-colour) . . . .	110
4.12	Structure of CL-20 molecule . . . . .	111
4.13	Example of 3D molecular conformation of CL-20, with one of six analogous improper torsions indicated by the yellow highlighted atoms . . . . .	112
4.14	Experimental CL-20 structure example ( $\gamma$ [158]) after DFTB+ re-optimisation, showing breaking of intramolecular bonds. . . . .	113
4.15	Pie charts showing the effectiveness of pXRD clustering using different parameter sets . . . . .	121
4.16	Pie charts showing the risk of fallacious removals of using pXRD clustering with different parameter sets . . . . .	123

5.1	Venn diagram showing the number of structures selected as synthesisable only by the corresponding method or set of methods - indicated by the number within each section. The area of each section scales with the number of structures represented. The generalised convex hull method used in this instance is a single case, with a 4Å SOAP cut-off , an adapted SOAP kernel, and a 1D hull. . . . .	128
5.2	Plot of an example of a GCH landscape (here of predicted ROY structures). The determined convex hull is marked by the grey lines (facets) joining the orange vertices. The grey horizontal bar indicates the global minimum height. Black arrows represent the measure of the relative energy of some example experimental polymorphs. Red arrows indicate the corresponding dressed energies - which are lower in all cases. . . . .	129
5.3	Energy density landscapes for each explored CSP set,with the structures matching to the experimental polymorphs indicated by orange stars. . . . .	134
5.4	GCH landscapes for the set of predicted ROY crystal structures derived using the average and adapted SOAP kernels. Landscapes are constructed upon lattice energy and the top-ranked kPCA component from each kernel implementation . . . .	136
5.5	The candidate pools for each system - as assessed using GCH implementations (4 Å cut-off radii, 1D hull) employing the average and adapted SOAP kernels. Each block represents a single predicted crystal structure in the candidate pool. . . . .	138
5.6	Eigenspectra for the kPCA projections for each system explored in this chapter. Spectra are shown over the limited range of the first 32 components of the projection in each case. Different line-styles and colours denote the kernel used to form the kPCA projection. . . . .	141
5.7	Graphs of cumulative variance captured by the first $x$ components of the kPCA projections for each system explored in this chapter. Graphs are shown over the limited range of the first 32 components of the projection in each case. Different line-styles and colours denote the kernel used to form the kPCA projection. . . .	142
5.8	Candidate pools for each system as a function of hull dimensionality, SOAP cut-off, and kernel construction for each. Different curves denote different kernel constructions - with different kernel types and underlying SOAP cut-offs. . . . .	152

5.9	Boxplots showing the spread of calculated candidate pools for a single example case (T2, adapted kernel, 4Å cut-off, 1D hull), when the iterative workflow to account for energetic uncertainty was applied. Each colour signifies a different number of iterations of the loop used and each boxplot of a given colour corresponds to a separate ‘run’ of the workflow with that many iterations. . . . .	155
5.10	Box plots showing the spread of calculated candidate pools arising from the finalised iterative workflow for each system investigated from the average and adapted kernel GCH implementations (4 Å cut-off) using different hull dimensionalities. .	157
5.11	Energy density and GCH (Adapted kernel, 4Å cut-off, 1D hull landscapes of the ROY CSP set. . . . .	162
5.12	Candidate pools for each system as a function of hull dimensionality, SOAP cut-off, and kernel construction for each. Different curves denote different kernel constructions - with different kernel types and underlying SOAP cut-offs. Horizontal bars indicate the candidate pools for each system as calculated via traditional landscape analysis methods. . . . .	163
6.1	Structure of the ROY molecule - the bonds defining the key torsional angle $\alpha$ are highlighted in red. . . . .	174
6.2	Visualisations of the asymmetric units of the three $Z'=2$ experimental crystal structures of ROY.[55]. It can be seen that in each respective asymmetric unit, the two molecules will share similar absolute values of torsion $\alpha$ . . . . .	175
6.3	1D GCH landscape of ROY (4Å SOAP cut-off, Adapted kernel construction) with points coloured by values for the key torsional angle $\alpha$ in the underlying molecules.	175
6.4	1D GCH landscapes (4 Å SOAP cut-off) of ROY from the average and adapted kernel constructions - coloured by the value of the molecular conformation binary classification descriptor. . . . .	176
6.5	1D GCH landscape of ROY (4Å cut-off) using the second-ranked ML descriptor from the average kernel construction. Points are coloured according to the molecular conformation classification descriptor. . . . .	177
6.6	Truncated dendrogram for the agglomerative clustering upon the full galunisertib conformational assignment data. Each ‘splitting’ of the tree represents the separation of two clusters. The y axis denotes the distance between clusters and their parent clusters. Each of the eight clusters used in final clustering has been uniquely coloured within the diagram. . . . .	179

6.7	1D GCH landscapes of $Z'=1$ Galunisertib crystal structures using differently ranked ML descriptors derived from the average and adapted kernel constructions. All kernel constructions used a 4 Å SOAP cut-off. Points are coloured according to the cluster to which the structure was assigned based upon its in crystal molecular conformation. . . . .	180
6.8	Fragments of molecular structures used within the motif search to identify utilised hydrogen bond acceptors (a) and donors (b) within the crystal structures. The vector construction shown in c) indicates how class labels were applied to structures base upon the presence (1) or absence (0) of hydrogen bonds utilising the respective acceptors. . . . .	182
6.9	1D GCH landscape of galunisertib (Adapted kernel construction, 4Å SOAP cut-off). Points are coloured by the class to which each structure was assigned based upon the hydrogen-bonding motifs present in the crystal. . . . .	183
6.10	1D GCH landscapes of galunisertib crystal structures using differently ranked ML descriptors derived from the average and adapted kernel constructions. All kernel constructions used a 4Å SOAP cut-off. Points are coloured according to the cluster to which the structure was assigned based upon its use or neglect of the oxygen h-bond acceptor. . . . .	184
6.11	Plots of the top-ranked ML descriptors (derived using average and adapted kernel constructions with a 4 Å SOAP cut-off) for the systems of DAP, T2 and trimesic acid against the density of the crystal structures. . . . .	186
6.12	Plots of the second-ranked MI descriptors (derived using average and adapted kernel constructions with a 4 Å SOAP cut-off) for the systems of trimesic acid against the density of the crystal structures. . . . .	187
6.13	1D GCH landscapes of ROY from the adapted kernel constructions using various SOAP cut-off radii- coloured by the value of the molecular conformation binary classification descriptor. . . . .	189
6.14	1D GCH landscapes of galunisertib from the adapted kernel constructions using various SOAP cut-off radii- coloured by whether or not the crystal structures utilised the oxygen h-bond acceptor. . . . .	190
6.15	Plots of the top-ranked MI descriptors of DAP (derived using adapted kernel constructions with varying SOAP cut-offs) against the density of the crystal structures. . . . .	190



6.16	Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor from either single (blue) or multiple (yellow) ML descriptors as a function of the underlying SOAP cut-off radii. Line style denotes the type of kernel construction used to derive the ML descriptors. . . . .	195
6.17	Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor of ROY structures from single ML descriptors as a function of the ranking -within the kPCA decomposition - of the ML descriptor. Line style and colour denotes the type of kernel construction used to derive the ML descriptors.	197
6.18	Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor of galunisertib structures from single ML descriptors as a function of the ranking -within the kPCA decomposition - of the ML descriptor. Line style and colour denotes the type of kernel construction used to derive the ML descriptors. . . . .	198
7.1	Process used to derive the kernel, training set feature data, and test set feature data for GPR workflows from the full structure-set kernel. Sub-matrices are extracted by selecting the corresponding rows and columns to obtain the training-set training set similarities (training set feature data) and test set - training set similarities (test set feature data). The training set - training set similarity submatrix is itself a complete similarity kernel across the training set and is also used in GPR as the kernel to fit the prior. . . . .	203
7.2	Molecular diagrams of each system used in initial ML exploration : Target XXXI, Target XXXII and Chlorpropamide . . . . .	205
7.3	Workflow for selecting training and test sets during cross validation in initial ML exploration of each system. A repeated 5-fold cross validation is used, with shuffling of the dataset between each run. During testing on each test fold of each run, subsets of different sizes are randomly selected from the respective training fold to sample different training set sizes. The RMSE and MAE are evaluated for all trialled cases . . . . .	206
7.4	Average RMSE and MAE of energy predictions for each system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Different colours indicate different cut-off radii of the underlying SOAP descriptors used to derive the kernel. . . . .	207

7.5	Average MAE of energy predictions for each system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Only cases with 6 Å cut-off radii of the underlying descriptor are shown , for simplicity. Shaded areas centred about each curve indicate a margin of uncertainty of the declared errors - with the width of the shading being equal to a single standard deviation either side of the curve. . . . .	209
7.6	Training curves (left) and display of uncertainty of measured errors (right) - analogous to the results in Figures 7.4 and 7.5 respectively - for the $Z'=1$ subset of the chlorpropamide landscape. . . . .	211
7.7	Example of physically unrealistic predicted chlorpropamide structure found in the set. The measured intermolecular oxygen-hydrogen atomic separation (1.359 Å) is unreasonably short - leading to formation of a bond between molecules in the visualisation . . . . .	213
7.8	Histogram displaying the distribution of calculated total energies for the first 5000 chlorpropamide crystal structures tested. The red shading indicates the region declared unrealistic - determined by use of an interquartile range criterion on the set of energies - and any structure whose energies is calculated to lie within this region is rejected. . . . .	214
7.9	Scatterplot demonstrating the random spread of selected training and test set structures across the landscape. Marker colour denotes the set for which the structure was selected. The 'Full set' displayed does not include structures whose energy was considered unreasonable or that failed energy-calculations. . . . .	216
7.10	Average RMSE and MAE of energy predictions for the extended case of the chlorpropamide system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Different colours indicate different cut-off radii of the underlying SOAP descriptors used to derive the kernel. . . . .	217

7.11	Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon the entire set of 8000 structures for which pDFT single point energies were calculated. Colour denotes the underlying descriptor cut-off radius and marker style indicates the kernel construction used in the GPR model. The error bars about each point display the uncertainty in the declared errors, with bars being of height (either side of the centre point) equal to one standard deviation of errors measured in cross validation. This displays the significance of the performance gap, with error bars arising from respective adapted and average kernel GPR implementations not overlapping. . . . .	220
7.12	Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon given low-energy subsets of the entire set of 8000 structures for which pDFT single point energies were calculated. Colour denotes the underlying descriptor cut-off radius and marker style indicates the kernel construction used in the GPR model. The error bars about each point display the uncertainty in the declared errors, with bars being of height (either side of the centre point) equal to one standard deviation of errors measured in cross validation. . . . .	223
8.1	A simple CSP workflow, in which the trial structure generation stage (orange) is performed via templating - which requires gathering of previously predicted crystal structures, followed by creation of structural analogues containing a new molecule . . . . .	227
8.2	The structures of two families of chemically-substituted similar molecules used in investigation of templating CSP . . . . .	230
8.3	Molecular Structures of NTCDA and PTCDA - a pair of molecules investigated for templating CSP . . . . .	231
8.4	Conceptual process for generating analogous structures . . . . .	232
8.5	An example of a template-target pair investigated, and the maximum common substructure (highlighted in yellow) identified between them. . . . .	233
8.6	A conceptual example demonstrating the need to consider overlay of all pairs of instances of the shared substructure in order to capture all valid analogous crystals. Each example overlay shows the overlaying of the substructure in the new molecule with different instances of the substructure within the old molecule - leading to different crystal structures. . . . .	234

8.7	A conceptual example demonstrating the need to consider all isomorphic overlays of a single pair of substructure instances in order to capture all valid analogous crystals. Each example overlay shows the overlaying of the single present substructure instance in each molecule, albeit overlaying different pairs of atoms - each according to isomorphic mappings of the substructure instances - leading to different crystal structures. . . . .	235
8.8	An example demonstrating the impact of shifting the centroid positions of molecules after forming an analogue via substructure overlay - altering the final analogue formed . . . . .	236
8.9	Flowchart of full process for generating analogues . . . . .	238
8.10	Examples of template structures, and one of the analogous crystal structures that was formed from each of them . . . . .	239
8.11	An example of an initial structural analogue formed in templating CSP, and the corresponding final structure after lattice energy minimisation. . . . .	240
8.12	Overlays of known a) $\alpha$ and b) $\beta$ crystal structures of PTCDA (Element Colour) and crystal structures predicted via templating CSP (green). . . . .	245
8.13	Overlay of the known experimental crystal structures of VENYUI (green) and VENZAP (Element-Colour), demonstrating their isostructural nature. . . . .	247
8.14	Overlay of the traditionally-predicted global minimum crystal structure of VENYUI (green) and a structure generated via templating CSP (Element-Colour), demonstrating a close match between the structures. . . . .	248
8.15	Overlay between a structure predicted by quasi-random CSP (element colour) and a structure predicted by templating CSP (green)- showcasing an almost perfect overlay . . . . .	249
8.16	Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . .	256
8.17	Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	257

8.18	Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	258
8.19	Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . .	260
8.20	Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	261
8.21	Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	262
8.22	Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the second family of NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	264
8.23	Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	265
8.24	Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	266
8.25	Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . .	267
8.26	Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	268

8.27	Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy . . . . .	269
8.28	Heatmaps for each template-target pair, showing the DMR values of templating for each region of the target landscape and each trialled starting set. Maps for each system are given by a) NTCDA in PTCDA, b) PTCDA in NTCDA, c) BXDIOX in CONYAH, d) CONYAH in BZDIOX, e) WARPOW in CONYAH, f) CONYAH in WARPOW, g) MEMTED in CONYAH, h) CONYAH in MEMTED, i) BZDIOX in WARPOW, j) WARPOW in BZDIOX, k) MEMTED in WARPOW, l) WARPOW in MEMTED, m) MEMTED in BZDIOX, n) BZDIOX in MEMTED, o) VENYUI in VENZA, p) VENZA in VENYUI . . . . .	272
8.29	A challenging case for which templating was tested - despite key molecular differences . . . . .	274
8.30	Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the MEMTED/NTCDA and NTCDA/PTCDA template-target pairs for each trialled starting set . . . . .	275
8.31	Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the MEMTED/NTCDA and NTCDA/PTCDA template-target pairs for each trialled starting set. . . . .	275
8.32	A case for which an isomorphic mapping between shared substructure instances in two molecules did not correspond to a viable overlay. The mapping required the atoms indexed in orange boxes to be overlaid (i.e atom 1- atom1) . . . . .	278
8.33	A case for which identified shared substructure - the indicated CHN chain- did not correspond to <i>intuitive</i> shared substructure (i.e shared motifs). The identified substructure instances in each case are labelled in equivalent order by the indexing in orange boxes . . . . .	279

# List of Tables

3.1	Metrics for each kernel construction showing how theoretically reasonable the results of the underlying unnormalised kernels are. . . . .	87
4.1	Results of crystal structure prediction on 4 organic semi-conductor molecules . .	93
4.2	Parameters used in each run of the <i>ADDSYM</i> search. All remaining parameters were left at the PLATON default values . . . . .	104
4.3	Structure sets sizes and experimental match results from each stage of re-optimisation of $Z'=1$ crystal structure prediction of primidone . . . . .	105
4.4	Structure sets sizes and experimental match results from each landscape calculated in crystal structure prediction of DAP . . . . .	109
4.5	Structure sets sizes and experimental match results from each landscape calculated in crystal structure prediction of CL-20. Entries marked * required looser ( $0.4\text{\AA}/40^\circ$ ) tolerances to recover . . . . .	117
4.6	Possible pXRD duplicate removal thresholds tested . . . . .	119
5.1	Molecular structures and numbers of known polymorphs for the five systems explored in this chapter. . . . .	131
5.2	Candidate pools, average polymorph rankings, and maximum polymorph rankings for each landscape - as determined using ranking based on relative lattice energy and based on use of an energy-density convex hull . . . . .	133
5.3	The candidate pool, average polymorph rankings, and maximum polymorph rankings for each systems - as assessed using GCH implementations ( $4\text{\AA}$ cut-off radii, 1D hull) employing the average and adapted SOAP kernels . . . . .	137
5.4	Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a $4\text{\AA}$ underlying SOAP cut-off radius. . . . .	146

5.5	Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a 6 Å underlying SOAP cut-off radius. . . . .	148
5.6	Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a 8 Å underlying SOAP cut-off radius. . . . .	150
5.7	Candidate pools for the primidone system calculated using various GCH implementations acting on the original CSP landscapes at different levels of theory. . .	159
5.8	Candidate pools for the CL-20 system calculated using various GCH implementations acting on the original CSP landscapes at different levels of theory . . . . .	160
5.9	Lowest average polymorph rankings calculated via a GCH implementation for each system, hull dimensionality, and SOAP cut-off radius. Cells are colour-coded according to comparison to traditional methods. Green = the best GCH approach results in a lower average polymorph ranking than the best traditional approach in that instance. Orange = the best GCH approach is on par with the best traditional approach Red = the best GCH approach results in a higher average polymorph ranking than the best traditional approach. . . . .	165
6.1	$R^2$ values for the best ML descriptor-density relationships identified for the DAP, T2, and trimesic acid systems using each kernel type. All kernel constructions used a 4 Å SOAP cut-off. . . . .	191
6.2	$R^2$ values for the best ML descriptor-density relationships identified for the DAP systems using each kernel type and various SOAP cut-off radii. . . . .	192
7.1	The structure set sizes and method of final energy-evaluation for each system used in initial ML explanation. True energy-evaluation workflows are multi-step processes - the original citations should be consulted for this information . . . . .	204
7.2	Calculated RMSE and MAE of energy predictions for the extended case of chlorpropamide - for each tested training set size, underlying descriptor cut-off radius, and kernel construction used in the GPR model. . . . .	218
7.3	Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon the entire set of 8000 structures for which pDFT single point energies were calculated. These results are shown for each tested underlying descriptor cut-off radius and kernel construction used in the GPR model . . . . .	219



7.4	Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon given low-energy subsets of the entire set of 8000 structures for which pDFT single point energies were calculated. These results are shown for each tested underlying descriptor cut-off radius and kernel construction used in the GPR model. . . . .	221
8.1	Definitions of terms that will be used to describe work on templating CSP . . . .	228
8.2	Prediction via templating CSP of structures matching to known experimental polymorphs for the NTCDA and PTCDA template-target pairs, for each trialled starting set . . . . .	243
8.3	Prediction via templating CSP of structures matching to known experimental polymorphs for the first family of chemically-substituted template-target pairs, for each trialled starting set . . . . .	244
8.4	Prediction via templating CSP of structures matching to known experimental polymorphs for the second family of chemically-substituted template-target pairs, for each trialled starting set . . . . .	245
8.5	The number of structures in important regions of the traditional CSP landscape for each investigated system . . . . .	251
8.6	Number of unique analogues formed and number of target structures recovered in each region for each template-target pair and starting set. . . . .	255
A.1	Table of SOAP descriptor parameters used within kernel calculations in this thesis. See librascal [113] documentation for further information . . . . .	292

# List of Accompanying Material

Key data supporting the thesis is available via PURE: <https://doi.org/10.5258/SOTON/D3723>

## Declaration of Authorship

I, **Jennifer Eleanor Martin**, declare that the thesis entitled *Machine Learning Methods for Analysis of Organic Molecular Crystal Structure Prediction Landscapes* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research.

I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University
- where any part of this thesis has been previously submitted for a degree at this University or any other institution, this has been clearly stated
- where I have consulted the published work of others, this is always clearly attributed
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
- I have acknowledged all main sources of help
- where this thesis is based on work done jointly by myself with others, I have made clear exactly what was done by others and what I have contributed myself
- Some content in the introductory chapter of this thesis has been adapted from a published review, to which I contributed: R.J.Clements, J.Dickman, J.Johal, J.Martin, J.Glover and G.M.Day, *MRS Bull.*, 2022, **47**,1054–1062.

Signed:.....

Date:.....

# Acknowledgements

There are many people I would like to thank for all of their support in getting me to and through the PhD - far too many to list! Hopefully anyone I miss will let me off!

First and foremost, I would like to heartily thank my supervisor, Professor Graeme Day for all of his guidance throughout the PhD. You have always been a great source of scientific and academic advice and exciting ideas. But your supervision has also always been warm and encouraging and helped show me how to overcome setbacks in my work and keep going. You've been fun and supportive to work with and I am very grateful for all of the help that you have given me.

I'm sure Graeme also had a hand in fostering the friendly and encouraging environment of the Day group - all of whom I would like to thank! I have really enjoyed working with you, and everyone has always been on hand for lots of help with the science and plenty of great chats. Special thanks go to Dr Chris Taylor for all of his help and advice, and to the PhD students of the group - Patrick, Rebecca, Josh, Joe, Zen, Pedro, Hannah, Eleanor, James, Sophie, and Jay - for going through the journey with me.

I'd also like to thank Professor Michele Ceriotti, for his help and scientific advice along the way and - alongside Dr Andrea Anelli - for devising the approach on which this thesis rests! It made for some engaging and interesting years of research.

I'm also grateful to the Leverhulme trust via the Leverhulme Research Centre for Functional Materials Design for the funding that allowed me to undertake this project and to the MCC Consortium and HPC teams behind YOUNG and Iridis 5 for invaluable resources without which none of this work would have been possible.

Personally, I am very grateful to all of the friends who made the last few years so enjoyable and gave me some crucial relaxation away from the screen. The University of Southampton Hill-

walking Club has given me many chaotic adventures and a great set of friends who have supported me. I would like to specifically thank Will and Sarah for always being people I could turn to.

I don't want to forget my undergraduate years that helped me to this point. I would like to express my gratitude to the School of Chemistry at the University of St Andrews - and particularly to my Masters' project supervisor Dr Kevin Jones for all of his scientific support and kind supervision. I'd also like to thank the archery club and my other mates for the fun times - especially Connor for his caring friendship. And sorry to the many friends for not always keeping in touch.

Lastly, and winning the bonus for the longest running support, I would like to thank my family - my mum, dad, brother Adam, and sister-in-law Lacey - for getting me through all of the way to now. You've always been there when I needed you and that means a lot to me.

# Acronyms

API - Active Pharmaceutical Ingredient/Application Programming Interface

ASE - Atomic Simulation Environment

CCDC - Cambridge Crystallographic Data Centre

cDTW - Constrained Dynamic Time Warping

CH - Convex Hull

CSD - Cambridge Structural Database

CSP - Crystal Structure Prediction

DFT - Density Functional Theory

DFTB - Density Functional Tight Binding

DMA - Distributed Multipole Analysis

DMR - Distribution of Matches Ratio

ESF - Energy Structure Function (Maps)

GCH - Generalised Convex Hull

GPR - Gaussian Process Regression

GUI - Graphical User Interface

HOF - Hydrogen-bonded Organic Framework

IQR - Inter-Quartile Range

kPCA - Kernel Principal Component Analysis

KRR - Kernel Ridge Regression

MAE - Mean Absolute Error/ Mean Average Error

MDS - Multi Dimensional Scaling

ML - Machine Learning/Machine Learned

MSE - Mean Signed Error

MVN - Multi-Variate Normal

PCA - Principal Component Analysis

pDFT - Periodic Density Functional Theory

PSD - Positive Semi Definite

pXRD - Powder X-Ray Diffraction  
RMSD - Root Mean Square Difference  
RMSE - Root Mean Squared Error  
SOAP - Smooth Overlap of Atomic Positions  
ssNMR - Solid State Nuclear Magnetic Resonance  
SVC - Support Vector Classification  
VASP - Vienna Ab Initio Simulation Package





# Chapter 1

## Introduction

### 1.1 The Important Definitions

The first move in opening discussion of work into method development for the prediction of organic molecular crystal structures is to take a step back and provide a clear definitions of the concepts at hand.

#### 1.1.1 What is a Molecular Crystal Structure?

The formal definition of a crystal, used by the (IUCr), is a material in which the arrangement of the atoms, ions or molecules overall demonstrates long-range order or - defined in reciprocal space - a material generating sharp diffraction peaks [2].

It is useful to expand below, and provide a more intuitive explanation of what is thought of as a ‘crystal’ or more precisely a crystal structure in this work. A crystal structure can be thought of as a solid form structure with a regularly repeating motif, that may extend in 2 or more dimensions. (see Figure 1.1).

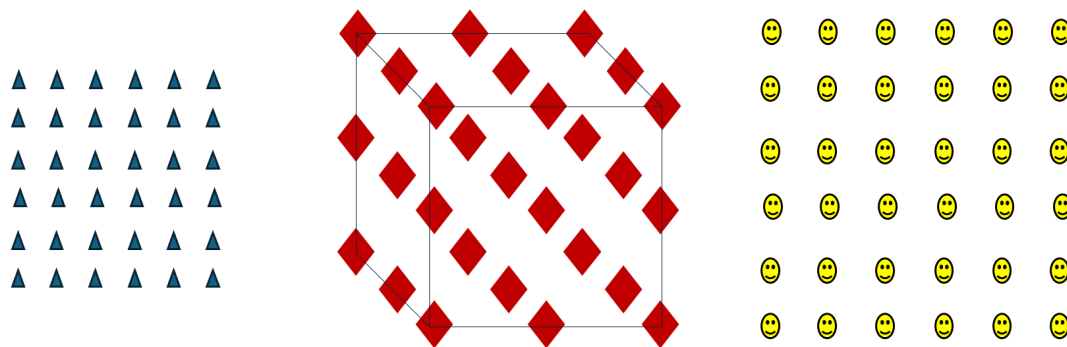


Figure 1.1: Some examples of conceptual crystal structures -regular patterns of a repeated motif

Perhaps the simplest crystal structure to imagine is an array of equally spaced identical points. This structure is rare in nature, but elemental polonium is known to crystallise in this simple structure [3] (Figure 1.2).

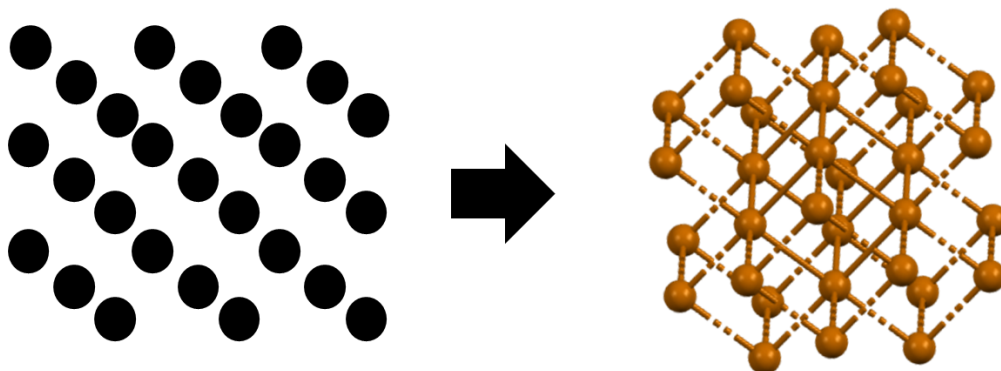


Figure 1.2: Simple periodic array of points and corresponding real-world example - the simple cubic crystal structure of elemental polonium [3]

Most crystal structures, however, are more complex. The regularly repeating motif need not be, and indeed in the vast majority of cases will not be, a single point or atom. In chemistry, a **translationally** repeated unit of a crystal structure is known as a unit cell. This is, in essence, a hypothetical box, containing atoms or molecules. The translational symmetry of this unit cell - extended in all relevant dimensions - generates the full crystal structure. The primitive unit cell is the smallest possible translationally repeated unit cell.

A **molecular** crystal structure is, straightforwardly, a crystal structure in which the unit cell contains molecules. It is important to note though, that for most chemical purposes, and for all of those pertinent to this work, a molecular crystal structure will be 3D. That is that the translational symmetry of the unit cell will extend in three dimensions. And the unit cell will itself be a 3D space.

The translational symmetry of the unit cell may be the only symmetry within the crystal structure. However, in many cases additional symmetry can be found. That is, there can also be symmetry **within** the unit cell- i.e the ‘contents of the box’ can be symmetrically related. Imagine, for example, a unit cell - in which there are two molecules positioned such that a mirror plane could be drawn between the two (Figure 1.3)

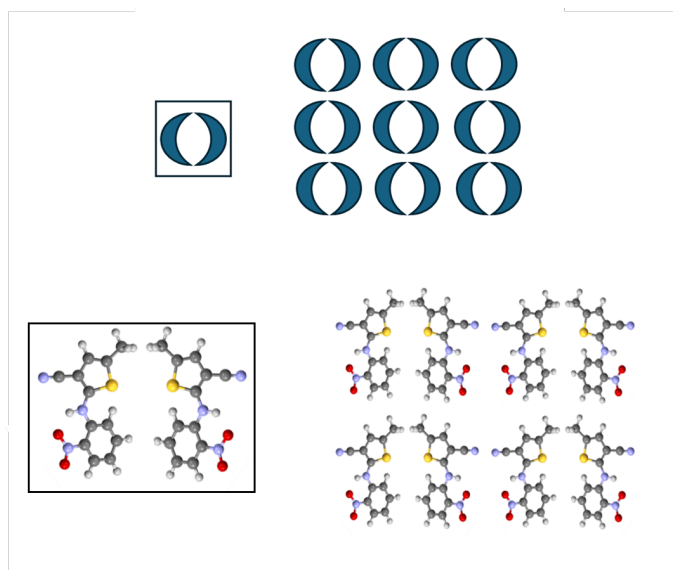


Figure 1.3: Conceptual demonstration of symmetry within a unit cell. A 2D crystal structure generated from a ‘unit cell’ of crescent objects related by a mirror plane and an example representing the same concept albeit with a molecule - forming a hypothetical 2D crystal structure of the molecule ROY

The structure could then be described by a smaller repeating unit (considering replication by more complex symmetry operators and not just translation). This smallest repeated unit is known as the **asymmetric unit**. Crystal structures also have a formula unit which for the work in this thesis is defined as the simplest integer ratio of chemically unique molecules in the unit cell. The number of formula units in the asymmetric unit is known as the  **$Z'$**  of the crystal structure. The set of symmetry operators that replicate the asymmetric unit to generate all the molecules in the unit cell defines the **space group** of the crystal structure. An example of a possible molecular crystal structure - of the molecule DAP - with additional symmetry within the unit cell is shown in Figure 1.4.

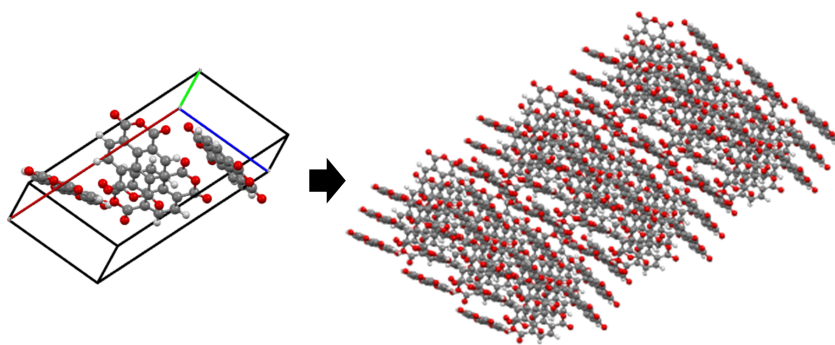


Figure 1.4: An example of a predicted molecular crystal structure, here of molecule DAP, and its corresponding unit cell

Interestingly, despite the wide range of possible values for all of the degrees of freedom (e.g unit cell angles, asymmetric unit molecule positions and orientations) defining a crystal structure, all 3D crystal structures will have one of 230 possible space groups, one of 14 ‘types of lattice’ - known as Bravais lattices - and one of just seven crystal systems, which classify the unit cell based on its lengths and angles. [4, 5]

These restrictions are not based upon chemical motivations, but are mathematical restrictions, which arise due to symmetry restrictions and equivalences of structures when viewed simply as sets of special points - known as lattice points. Lattice points are hypothetical points in a lattice representing the centres of identical environments, that are translationally repeated in the extended crystal structure.[5]

The first restriction to note is that there the possible relationships between different unit cell lengths and angles can define only seven unique ‘box types’, known as crystal systems. Then, when these seven possible crystal systems are combined with different possible centerings of the cell, there are 14 possible lattice types - known as Bravais lattices [4–6]. The centering here refers to the positions of the lattice points (i.e whether lattice points are merely at the vertices of the unit cell, or whether there are additional lattice points within the faces or body of the unit cell). [5] A further restriction is also enforced upon the sets of symmetry operations defining the symmetric relation of molecules about the lattice points. These sets of symmetry operations must form a mathematical group [7] and, due to restrictions upon the maximum order of rotational symmetry when forming a discrete periodic lattice [5], there are 32 such groups that can describe that symmetry. Combining these 32 possible groups with the 14 Bravais lattices, and accounting for redundancy, there are just

230 possible space groups. [4, 5]

### 1.1.2 What is Crystal Structure Prediction?

Crystal Structure Prediction (CSP) is, as the name implies, the prediction of crystal structures. More specifically, at least in the modern day, it refers to a field of research in which computational methods are used to attempt to predict possible crystal structures of a material. Depending upon the precise nature of the work, this prediction process may start from merely a list of element types. i.e - the researcher attempts to predict the possible crystal structures that may be adopted by materials containing a given set of elements - or from a more detailed knowledge of the composition and bonding of the material.

In the case of molecular crystal structure prediction, the researcher attempts to predict the possible crystal structures of a **molecule**. In most cases, this begins with knowledge only of the graph of that molecule. That is, that the composition and the connectivity of the atoms is known, such that it could be represented by a 2D molecular diagram - but that is all. From this knowledge, the researcher applies computational workflows to arrive at a set of predictions of likely crystal structures that could form of that molecule (Figure 1.5).

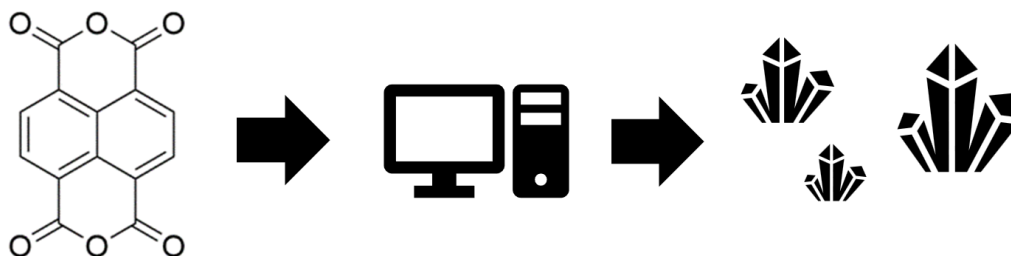


Figure 1.5: A conceptual CSP workflow: The knowledge of the connectivity of a molecule is fed through a series of computations which output many predicted crystal structures

The question of which crystal structures a molecule could adopt is not a purely hypothetical one. The researcher does not merely derive a list of all mathematically possible crystalline spatial arrangements of the molecule. Instead, the computational workflows apply chemical knowledge and theory - such as the energy of atom-atom interactions - to determine structures that are **most likely** to be found as the result of real-world crystallisation.

## 1.2 Literature and Background

With the key terms defined, the next section aims to provide a picture of the current state of crystal structure prediction, giving a quick overview of research in the area including common approaches, as well as key successes and limitations of the field. It also discusses some of the key applications of CSP to the field of materials discovery. While many tools and methods are mentioned, their workings are not discussed in detail here. More extensive explanation of those tools most pertinent to the research in this thesis are discussed in Chapter 2.

### 1.2.1 Where Does Crystal Structure Prediction Stand Today?

Crystal Structure prediction has been a long desirable goal as the properties of materials, which can determine their utility, often depend upon the crystal packing. For example, the crystal structure can affect the bioavailability and solubility of pharmaceuticals [8], the charge mobility in semiconductors [9], and the gas storage capacity of porous materials [10]. It is therefore beneficial in a materials discovery workflow to be able to predict the crystal structure, and so potentially predict the properties and stability of a material, prior to synthesis. Particularly where a discovery approach aims to screen many potential materials for a given task, use of prediction based methods can reduce the pool of candidate materials before experimental work commences. This conserves time and resources - and so is both economically and ecologically advantageous. Today, CSP has found successful applications in functional materials design [10–13] as well as pharmaceuticals [14–16].

Most approaches to CSP aim to identify reasonable predicted crystal structures by construction of an structure-energy landscape. This represents a measure of the energy (such as the lattice energy) of the system as a function of crystal structure. It is assumed that any possible crystal structure will represent a local minimum on the energy landscape. The task is then to identify those minima. This usually involves two key stages [17, 18]:

1. Construction of initial ‘trial’ crystal structures (Sampling)
2. Lattice energy minimisation of trial crystal structures (Optimisation)

This identifies possible basins (local minima) on the structure-energy landscape by finding the relaxed structures resulting from the optimisation stage. Each successful trial structure from the sampling stage will relax into one of these basins during optimisation (Figure 1.6).

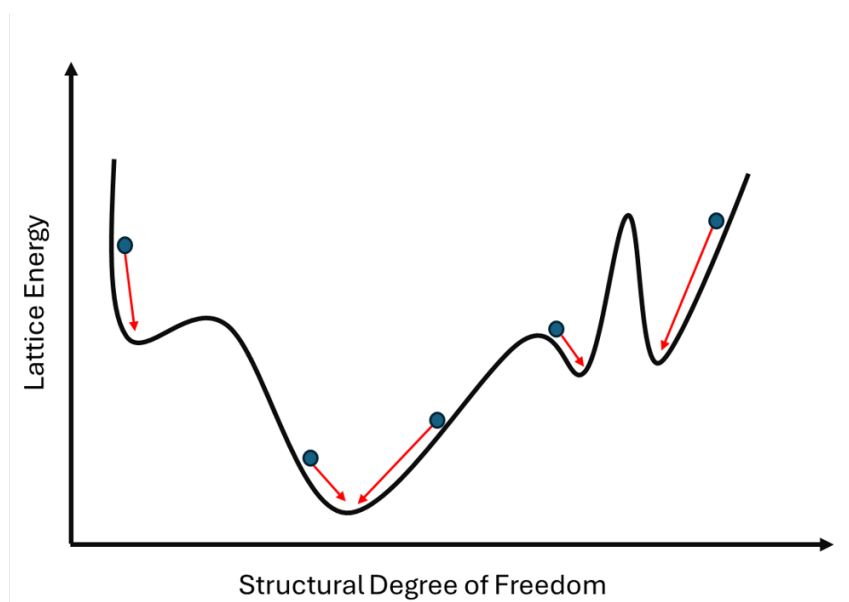


Figure 1.6: Demonstration of a simplified structure-energy landscape defined by one degree of freedom. Blue circles represent trial structures - with red arrows indicating the basin (local minimum) into which the corresponding structure will relax during the geometry optimisation stage

Most approaches will further seek to identify the most likely subset of these optimised structures - often based on an energetic ranking. This is discussed in depth in Section 1.2.2.

The chief difficulty for Crystal Structure Prediction approaches is the size of the necessary search space for trial structures. This makes sampling a large and costly problem to tackle. The search space is defined by many degrees of freedom that describe the parameters of the unit cell, and the position of atoms within the unit cell. In the case of inorganic crystal structure prediction, further degrees of freedom describing the composition of the material must also often be searched.

Some reprieve is offered to the problem of the expansive search space by application of constraints upon what is physically reasonable. Such considerations restrict the possible range of values that the degrees of freedom can take. Further, in the case of molecular crystal structure prediction, the possible atomic positions are constrained by the requirement to maintain the connectivity of the molecule. This transforms the manner in which the search space is considered - most commonly in the molecular case, instead of freely sampling all atomic positions independently, the positions of molecular units are sampled. Whilst this reduces the search space for sampling position, it introduces an additional degree of freedom - the orientation of the molecules. This is a factor that need not be considered when treating crystal structures as collections of atoms - which are usually treated as being isotropic. As an illustrative example of the search space for molecular crystal

structures, a molecular crystal with 3 molecules in the asymmetric unit is usually, excepting some instances of fewer degrees of freedom due to symmetric equivalences, defined by 24 degrees of freedom (three unit cell lengths, three unit cell angles, and three positional and three orientational degrees of freedom per molecule).

Even accounting for the constraints discussed above, the search space cannot be sampled comprehensively. To resolve this, several methods have been developed which aim to sample the space broadly and efficiently. Whilst these methods will never trial **all** hypothetical structures, the philosophy of these sampling methods relies upon the concept that there is an more achievable level of sampling of the space from which all **plausible** structures can be sampled. That more achievable sampling would be that for each basin on the energy landscape, at least one structure will be trialled that relaxes into that basin upon optimisation - i.e all local minima are sampled. The sampling methods aim to come as close as possible to this more achievable definition of complete sampling. One sampling method is random sampling [14, 19], which generates trial structures pseudorandomly to sample the space with minimal bias - therefore hoping to sample all regions of space fairly. A similar approach is the use of quasi-random sampling methods - which aim to provide largely unbiased sampling of the space, similar to random sampling, albeit with constraints to avoid any given region of space being over/under sampled by chance [20]. Other approaches include basin-hopping methods, which aim to search the space by perturbing trial structures from one basin of the landscape to another [21], and genetic algorithms which - inspired by Darwinian evolution - aim to construct **desirable** trial structures via adaptation of high performing (e.g low energy) structures constructed and optimised earlier in the sampling [22, 23].

However, sampling is not the only concern and the optimisation stage also poses a problem due to computational cost. With many trial structures to optimise, methods must balance accuracy and cost. Current approaches used for optimisation usually employ gradient descent to minimise the energy of the predictions by altering structure. There are several different energy calculators that are employed in these methods. These include interatomic forcefield methods (including empirical [24, 25] and more accurate non-empirical [26] or tailor-made forcefields [27]). Forcefields may be accompanied with point charges or atomic multipoles [24, 25] to incorporate permanent electrostatics into energy calculation. Another approach is the use of periodic Density Functional Theory (pDFT) [19]. The choice of approach presents the classic trade-off of performance and cost. Density Functional approaches provide reliable results but are computationally expensive, whereas forcefield based approaches require a fraction of the cost - but provide less accurate en-



ergies and optimised geometries [28]. Recent developments may reduce this battle, however, with growing use of machine-learning approaches [29–31] to provide energy calculations approaching the accuracy of those from high-levels of theory, but at much lower cost. It should be noted that high quality, trustworthy energy calculations are important here, because it is usually these energies - and the energetic ranking of predicted structures stemming from them - that are used to consider the likely synthesizability of a predicted structure. (See Section 1.2.2 for more details). Other factors, such as the slopes of the structure-energy landscape, or the size of the landscape basins corresponding to the minima, are normally ignored, though this does not commonly prove problematic and CSP has developed into a successful field even when considering only the energies of predicted structures.

Another problem, particularly for molecular crystal structure prediction, is the propensity for polymorphism. Polymorphism is the existence of more than one crystalline solid state form (polymorph) for a given material [32] - a particular molecule or inorganic composition. The problem of polymorphism means that there is not a single ‘correct answer’ for crystal structure prediction to find. Indeed there may be several polymorphs, all of which should be predicted. Correctly screening for or predicting polymorphism is crucial. The classic case of Ritonavir demonstrates this. After the development of drug Ritonavir, an anti-viral drug used in the treatment of HIV, a new polymorph was encountered. The solubility properties of this late-appearing polymorph were not appropriate for use and the drug had to be withdrawn from market while the issues were addressed. The withdrawal led of course to economic losses but, crucially, will have led to suffering and uncertainty for those who relied upon the vital treatment [8, 33]. The polymorphism problem however, does lead to another surprising and interesting application of CSP - patent law. The patent for a material or particular use case of a material may be specific to a given crystal structure of that material. Prediction of possible alternative feasible crystal structures can then be used to further secure or to circumvent a patent [34, 35].

Polymorphism is particularly common in molecular crystals with estimates that from 32-51% of small organic molecules are polymorphic [8]. Some molecules have large numbers of Polymorphs. Famously, ROY is highly polymorphic and has 12 known characterised polymorphs [36]. Polymorphism is so frequent because the solid state structures in these cases are formed and maintained by comparatively weak forces between the molecules. Being governed by these weak interactions, there is no clear strong ‘guide’ enforcing constraints on the possible structure - as there may be in inorganic crystal structures formed of oppositely charged ions. Further, this means that the

differences in lattice energy between different structures are small and so there can be several thermodynamically competitive polymorphs.

There are two types of polymorphism for molecular crystals - packing polymorphism and conformational polymorphism. Packing Polymorphism refers to cases where two polymorphs are composed of molecules that are 'packed' differently - the positioning of molecules relative to one another in space - differs between the polymorphs. Conformational polymorphism, refers to cases where the conformation of the molecules differs between the polymorphs (Figure 1.7). Two polymorphs may differ only due to one type of polymorphism - or may feature differences both in packing and in the conformation of the underlying molecule. Important to note for CSP, is the high potential for packing polymorphism - and so the possibility of multiple polymorphs must be considered even when investigating small inflexible molecules.

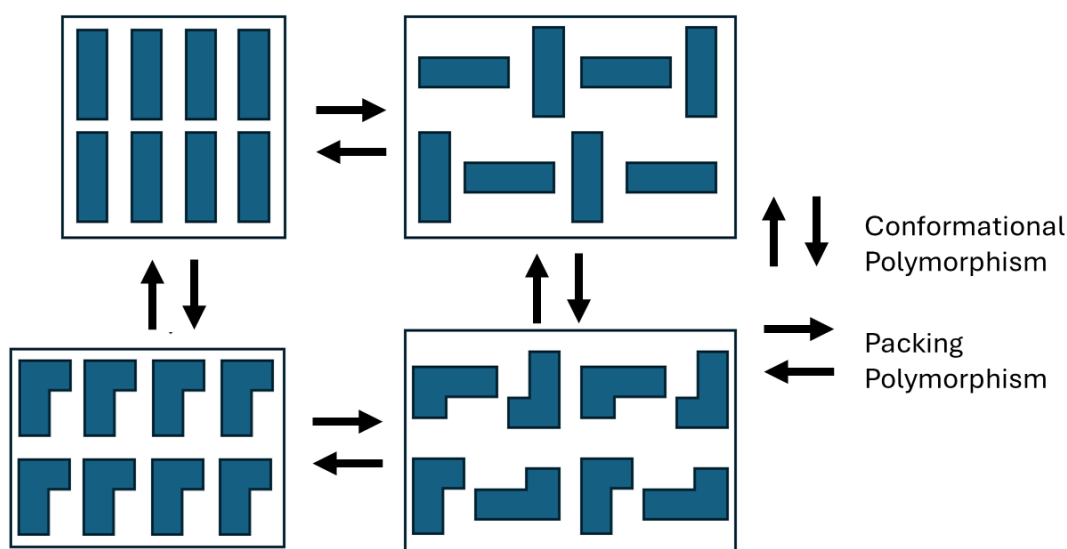


Figure 1.7: Conceptual demonstration of packing and conformational polymorphism. The 'molecule' here is represented by a blue polygon, with different shapes representing different conformations of the same molecule

Because the energy differences between polymorphs are small ( $\sim 2$  kJ/mol) [37], this makes CSP difficult and costly as in order to develop a useful energy landscape of predicted structures, the error in the calculated energies needs to be competitive with or better than this polymorph pair difference. Therefore, highly accurate and costly energy calculation methods must be used.

Due to the aforementioned complexities and computational costs of the task, the field has only

developed in recent decades, taking advantage of advancements in computer technology [38]. Rapid progress has been made in that time, however -with large increases in the popularity of the field, vast speed-ups in the time required for predictions, and great broadening of the systems that can be tackled. Perhaps the clearest insight into the progress made in the area is in the study of the CSP blind tests [17, 18, 39–44]. These are a series of challenges organised by the Cambridge Crystallographic Data Centre (CCDC) in which molecules that have characterised but unpublished crystal structures are set as ‘targets’ and participating researchers attempt to predict the crystal structures using their chosen methods. The tests are designed to push methods and identify the successes and limitations of the field. Over time, the reports on these blind tests demonstrate the progress in what the methods can achieve.

The complexity of the targets in the blind test has generally increased - with targets becoming larger and more flexible, as well as the introduction of salt and co-crystal systems and systems known to feature disorder [17]. Some examples of targets from each blind test are shown in Figure 1.8. The tests demonstrate a notable improvement over time, particularly in the successful prediction of crystal structures of flexible targets [17]. Even the setup of the blind test reflects the improved potential of CSP and the energy calculations used - with the energetic ranking of structure predictions now being tested rather than merely the presence or absence of the known structure within the prediction sets [45]. The tests demonstrate a transition from frequent use of forcefields as the highest level of theory towards more expensive methods and dispersion corrected pDFT-based methods now dominate blind-test submissions. This demonstrates the confidence of the community in these methods and reflects the increased access to fast computational resources [17, 18].

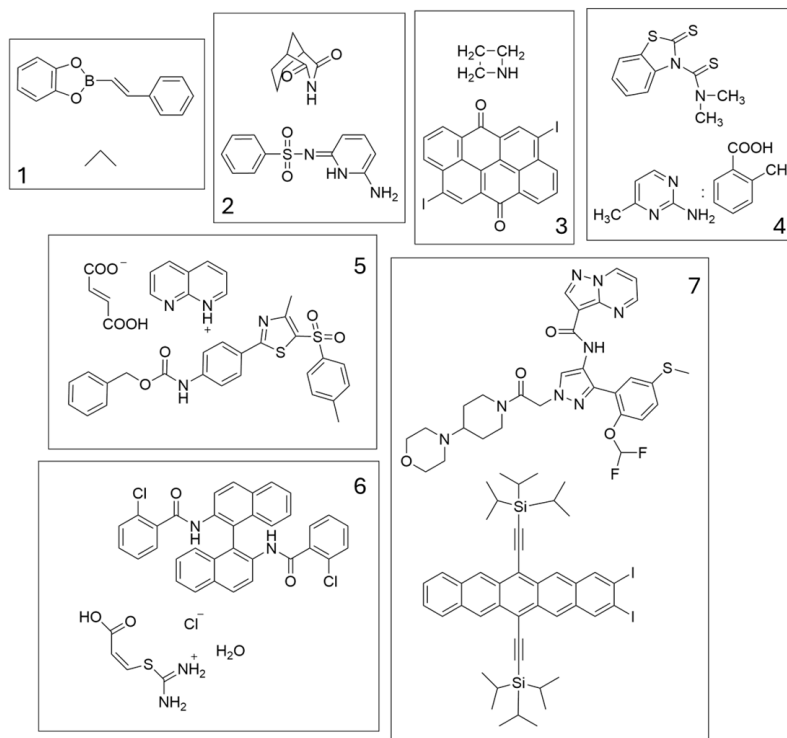


Figure 1.8: Examples of some of the target molecules used in each of the CSP Blind Tests [17, 39–44]. Numbers in each box indicate the test from which the target molecules are taken

Taking a look at the field today, it can be seen that there are wide applications of the research. Molecular CSP has been used recently in the field of pharmaceuticals to reveal diverse polymorphs of potential drug galunisertib [15] and identify the risk for potential high-pressure polymorphs of iproniazid [16]. The utility of CSP is also recently demonstrated in functional materials discovery. Examples include guiding the discovery process of multi-component organic cage pots [11], screening potential organic semiconductors [13, 46], helping discover a new mesoporous molecular crystal [12], and proposing a new high-pressure polymorph of an energetic material [47]. CSP has also been used to aid experimental crystal structure determination, for instance by combination with solid state Nuclear Magnetic Resonance (ssNMR) prediction [14]. There have also been exciting developments in methods available for both molecular and inorganic CSP, including recent work showcasing the use of deep-learning [48] and generative models [49].

Despite recent progress, CSP still faces key limitations. While the field appears to be on the cusp of routine handling of flexible molecules, some approaches - especially those that are more affordable - still rely upon rigid molecule CSP (which assumes a single in-crystal molecular conformation). This ignores the effects of intermolecular forces upon molecular conformation – leading to limited

applicability of the methods, especially to large or highly flexible systems. Further, where methods do consider molecular flexibility, this is often incorporated at separate stages - rather than fully flexible workflows that allow conformational changes throughout [17, 18], which may become a goal.

A further issue is that most methods construct the energy landscape at 0 K. Absolute zero is of course not reflective of the environment in which crystals will be grown and materials used - leading to poor applicability of predictions in some cases. Consideration of thermal effects can alter the energy landscape. For instance, accounting for thermal effects can lead to merging of basins on the structure-energy landscape - leading to a decreased number of stable predictions [50]. Further, including the vibrational contributions to the energy at a given temperature can alter the relative stability of predicted structures. These impacts are not insignificant. At 300 K, 9 % of polymorph pairs reverse in stability order [37] and so consideration of the more realistic room-temperature environment may yield different results.

Whilst neglect of thermal effects is one cause of over-prediction in CSP, these effects alone are insufficient to fully explain the over-prediction problem. It is not always known why predicted stable structures may not be observed [38]. However, one issue is that CSP is limited to considering thermodynamics. Thermodynamically stable predicted structures may be unstable or slow-growing due to kinetic effects - which have been ignored [51].

The CSP blind tests have highlighted a need for more exhaustive sampling - including searching for higher  $Z'$  structures, though such searching still faces a barrier of computational cost [18]. The tests also highlight the pitfalls of the commonplace neglect of potential disorder as well as the incompleteness of sampling/structure generation methods and the dependence of predictions upon the generation method chosen [17].

There is therefore much opportunity for further work and development in the field of molecular CSP to improve upon these issues - with particular targets for development including improved handling of molecular flexibility and consideration of disorder.

Having discussed the common approaches to CSP, its applications, and its drawbacks, it is useful to give a final note that, whilst discussion has focussed on approaches to CSP that revolve around energetic evaluation, this is not the only option. Examples of other approaches have been devised

that arrive at a final CSP landscape using optimisation and/or evaluation of structures based on topological qualities [52] and similarity to known structures of similar systems [53]. These methods have demonstrated some potential - but are not predominant approaches in the field. This is reflected in the CSP blind tests, with earlier test submissions featuring a wide range of non-energetic considerations, and later tests converging toward a focus on energetic stability [17, 18].

### **1.2.2 Approaches to Representations and Analysis of Crystal Structure Prediction Landscapes**

Once a set of predicted structures has been produced, it remains to then analyse the predicted structure-energy landscape of structures to help identify useful structure-function relationships, and determine the likely synthesisable structures.

The structure-energy landscapes for molecular crystals are often complex. Instead of merely funnelling down toward a single basin representing the global minimum structure, the landscapes often contain large numbers of basins corresponding to local minima. A CSP set for any given molecule therefore, if sufficiently sampled, will often contains large numbers of predicted crystal structures, each corresponding to a local minimum on the energy surface. However, most of these will not truly be competitive with the more thermodynamically stable forms and will not be found experimentally. Therefore, analysis of the landscape of predicted structures is required in order to identify the subset of predicted structures that are most likely to be synthesisable.

To do this, most methods rely upon the assumption that experimentally-realisable crystal structures will be significantly meta-stable, i.e. they will lie close to the global minimum structure in energy. Indeed this is not an unreasonable assumption. One study of molecular crystal structure prediction found that 95% of investigated small molecule polymorph pairs are separated by  $\leq 7.2$  kJ/mol [37]. Further, a study of the crystal structure prediction of over 1000 small organic molecules found that one of the polymorphs corresponded to the global minimum on the CSP landscape for 41% of cases [54]. Due to this principle, a common approach to identifying the likely synthesisable subset is to use an energetic cut-off approach and simply state that all structures whose lattice energy (or other sensible measure of energy) lies within  $x$  kJ/mol of that energy of the global minimum of the prediction set are worthwhile considering as synthesisable. All other structures are then disregarded.

Of course, due to the heavy reliance of this approach upon the calculated energy values, highly

accurate levels of theory are needed to produce trustworthy results. Some workflows will therefore include additional step(s) after the initial predictions have been made, in order to ensure high accuracy of the energetic ranking. This may include corrections to energy values [55] or reoptimisation of a subset of structures [12]. Given the importance of energy-ranking steps, and the universality of them within certain workflows - such work could be viewed as part of the prediction process itself, rather than analysis - the boundaries between the two being ill-defined. However, these steps can be performed separately after the fact - for instance to improve upon prior CSP sets from the literature [55] and so are discussed as analysis here.

Another issue with the simple energy-cut-off approach to identifying synthesisable structures is that whilst the principle has sound basis - it is not universally applicable. Some systems, particularly porous systems, may have high-energy polymorphs - that would not be considered as likely predicted structures under the energy cut-off criterion - but may be synthesisable experimentally [10].

Another question in the landscape analysis is that of how to represent the data. To present data in an easily human intuitable and engaging manner, it is often useful to present the energy-landscapes graphically. As discussed previously, the structure energy-landscape contains large numbers of degrees of freedom. Of course, such high-dimensional landscapes cannot be represented visually. Nor is such representation a particularly useful way in which to understand the data - one cannot easily derive meaningful patterns or relationships in the data based upon so many independent variables. Thus, the high dimensionality of CSP landscapes is often prohibitive to analysis.

The most conventional approach to data presentation is to represent the energy landscapes as visual graphs, in which each predicted structure is represented by a point plotted according to its energy and value of a single, useful, structural variable. In molecular CSP, an accepted convention is to represent prediction sets as Lattice Energy-Density plots. An example of such a plot is shown in Figure 1.9.

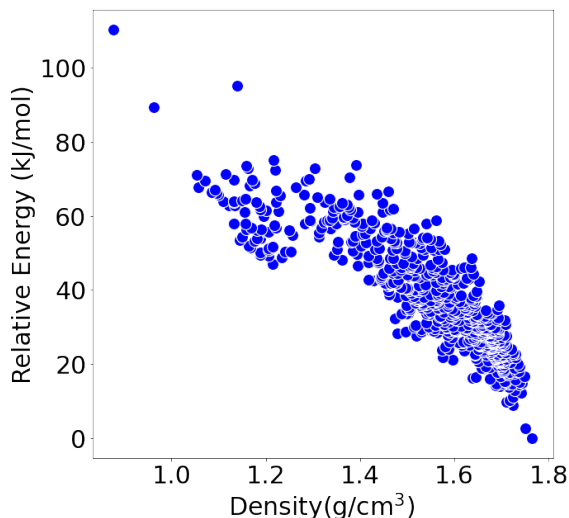


Figure 1.9: An example of an energy-density plot to represent the CSP landscape of molecule NTCDA. Each point represents a single predicted crystal structure

This allows for simple visualisation of the data and some display of the structural variance. Such representations can be useful as density is a meaningful structural feature and of potential interest to experimentalists. For instance, very high density structures can have explosive properties [51] and low density structures may have porous properties [12]. Density is also often related to energy - with denser structures often being more stable. This can be seen as many of the plots adopt the characteristic shape seen in Figure 1.9, where lattice energy can be seen to correlate with density.

However, in inorganic materials, a more sensible and thus more conventionally used variable for CSP landscape representations is composition. Inorganic materials discovery often begins from a list of elements - for which the researcher aims to explore the phase space - exploring both composition and crystal structure of the material. Inorganic CSP sets include predicted crystal structures across a range of compositions. An example is seen in Reference [56], which explored the stability of Uranium-Silicon phases of varying compositions in response to uncertainty over the stability of a  $\text{U}_5\text{Si}_4$  phase. Thus, composition is a key variable, comprising much of the variance of the set. As such, many inorganic CSP sets will be represented as plots of lattice energy and composition.

This leads discussion to an alternative view of stability - that of stability relative to the convex hull (Section 2.3.3). This approach selects a subset of the predicted structures as ‘hull vertices’ of a convex hull drawn across the bottom of a landscape defined by energy and some property - usually composition. The hull vertices are the structures predicted to be stable/stabilisable. Structures above the hull are meta-stable, with the height above the hull being a measure of relative stability



[1]. This is particularly important analysis in the case of exploring varied compositions - as not all compositions will be stable. A composition for which there is not a crystal structure lying on the convex hull will decompose into a phase-separated mixture of stable compositions because the total energy of the mixture will be lower than the energy of the single phase. [57] A proof of this can be seen in reference [57] (Figure 1.10). Therefore, exploring the entire phase space and analysing the results in this way is required not just to determine the likely crystal structures but the accessible compositions themselves.

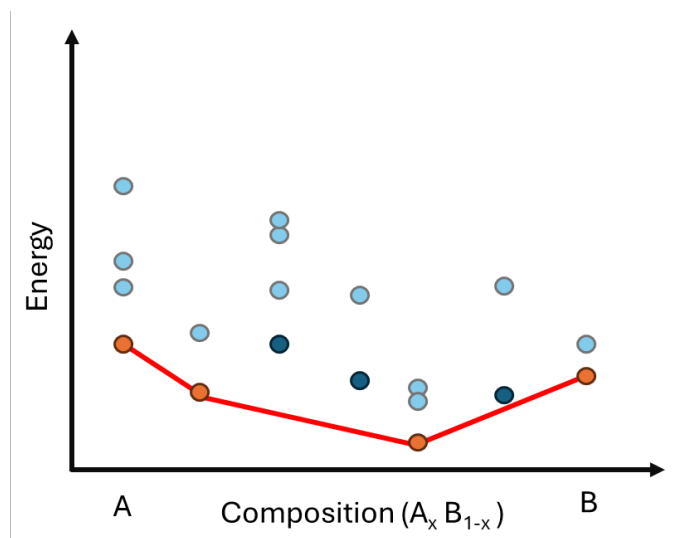


Figure 1.10: A hypothetical inorganic convex hull constructed on a landscape of composition and energy. Red lines indicate the hull - with orange points representing the hull vertices. The hull vertices indicate stable structures and the corresponding stable compositions. Dark blue spots represent the most stable crystal structure for a given unstable composition

The convex hull is used primarily in inorganic CSP, but has potential in organic molecular CSP such as in exploring stability of metal-doped organic crystals [58] and predicting the stoichiometry of solvates [59].

Representations of CSP sets are not just useful for visualisation and understanding stability, however. A useful approach for analysing CSP landscapes to identify promising functional materials is to overlay representations with the results of property prediction, forming Energy- Structure-Function (ESF) maps. This allows for easy visualisation of structure property-relationships and identification of structures that are likely to both be thermodynamically stable and have promising properties - the combination being crucial for successful materials design. Such maps have uncovered a highly-porous low density stable structure [10] and evaluated potential organic semiconductors [60]. An example of such a map is shown in Figure 1.11.

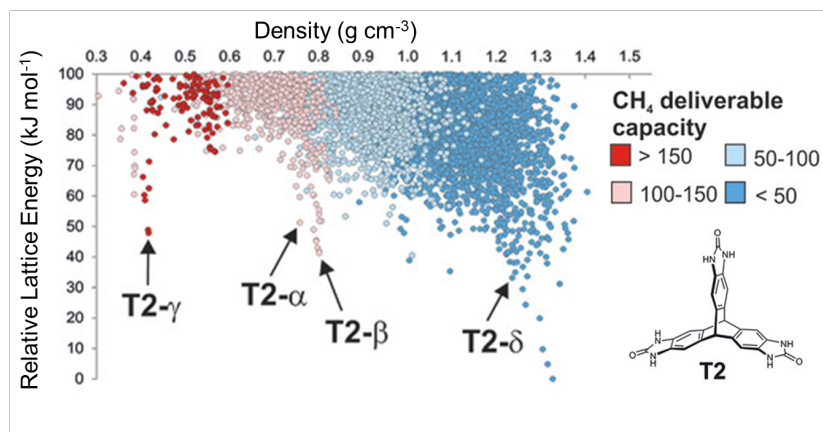


Figure 1.11: An example of an Energy Structure Function map - representing the stability and gas storage capacity of predicted crystal structures of molecule T2. Known stable polymorphs are labelled. Image adapted with permission from ref [10]

Careful choice of representations and analysis of landscapes has allowed better understanding of the explored systems and identification of useful and likely synthesisable new structures. However, traditional approaches do have drawbacks. One issue is that the conclusions drawn are affected by the information loss and researcher bias arising from the manual selection of structural variables across which to display and analyse the prediction sets. Furthermore, the issues in the underlying predictions are carried through. Conclusions drawn about structural stability, for instance, will be impacted by the uncertainty in structures and energies of the methods that predicted that landscape, unless this is explicitly accounted for in the analysis stage. Machine Learning (ML) advances, however, are beginning to address these problems and expand the abilities of and opportunities for landscape analysis.

### Machine Learning in Landscape Analysis

*Material in this section (Machine Learning in Landscape Analysis) is adapted and expanded from a published review article (Reference [31]), to which the thesis author contributed the section of material that has been adapted here. Adaptation is permitted under [Creative Commons License](#)*

A key example of the use of machine learning to address issues in traditional analysis is the use of unsupervised machine learning for dimensionality reduction. The intrinsic dimensionality of a landscape, i.e the minimum number of dimensions needed to fully describe a data set, is often lower than the full dimensionality. This can be due to simply correlation between dimensions or dimensions of negligible variance. Alternatively, it can be due to more complex relationships between combinations of dimensions such that combining information from a subset of the dimen-

sions can replicate much of the information from the full set of dimensions. This means that, by creating new coordinates to define the data, it could be represented in a lower dimensional space, without removing significant information (Figure 1.12).

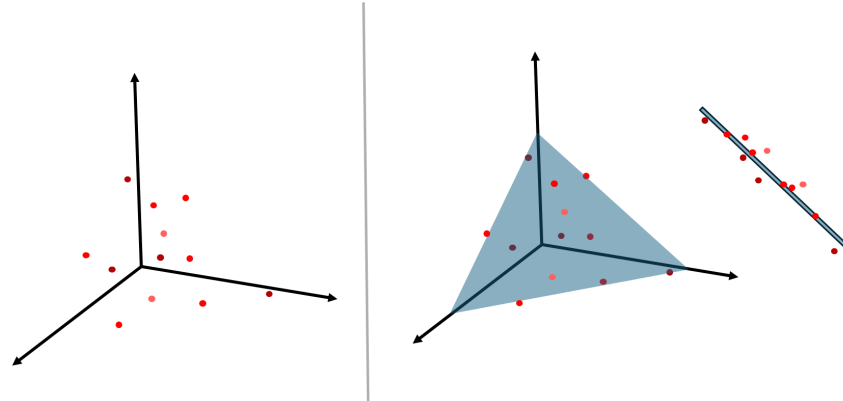


Figure 1.12: Demonstration of a hypothetical dataset with lower intrinsic dimensionality than full dimensionality. The left hand diagram indicates the dataset - a series of points spanning three dimensions. The right hand image reveals the lower intrinsic dimensionality - as all points lie almost within a 2D plane. Despite variance across all three dimensions - the original data could be restructured to be represented in just two dimensions, by noting its position within the indicated plane

A ML process - dimensionality reduction - finds the best approximation to such a low dimensional representation. In simple terms, these processes seek to find a smaller set of **new** variables aim to describe the existing data with less information loss than simply manually removing variables from the original set or selecting a single variable. This allows for creation of lower dimensional - so more intuitive and interpretable - mappings of the data.

There are several different dimensionality reduction algorithms that can be used. Examples that have been used in landscape analysis include sketch-map [13, 61–63], kernel Principal Component Analysis (kPCA) [1, 63, 64], and multi dimensional scaling (MDS) [65, 66].

Another common use of machine learning in the analysis of CSP landscapes is clustering. Clustering algorithms are unsupervised machine learning methods that optimise grouping of data points into clusters such that similarity within clusters and dissimilarity between clusters is maximised. In this way, complex data can be defined by just one descriptor – the cluster to which a data point belongs. This can be useful in landscape analysis particularly for classification of structures into

groups sharing structural characteristics and to identify structure-property relationships.

Clustering and dimensionality reduction have been used to create reduced mappings of pentacene and an azapentacene, and group structures into clusters corresponding to heuristic classes and sub-classes [62] (Figure 1.13). They have also revealed relationships between molecular structure, preferred crystal packing and electron mobility via a reduced mapping of 28 azaphenacenes [13]. Landscapes of porous molecular crystals have also been explored in this way, allowing identification of a pore geometry-gas storage capacity relationship [66] and separation of structures based upon porosity [67].

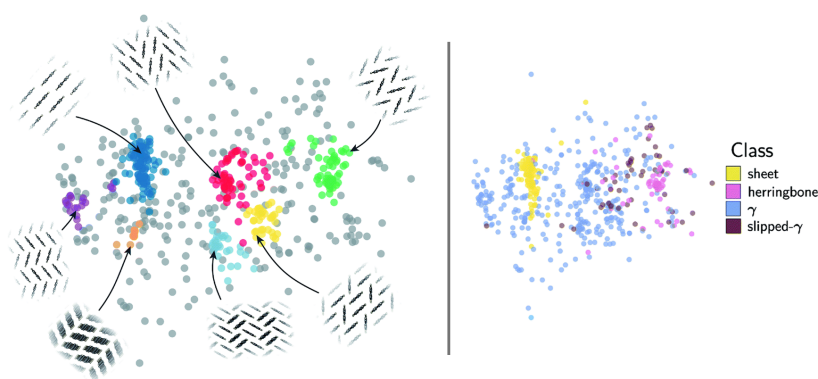


Figure 1.13: Example of the use of clustering to analyse CSP landscapes. The left hand image shows a CSP landscape, coloured by the cluster to which algorithms have assigned the points. The right hand image shows the same landscape coloured by the packing class of the structure. IT can be seen that there is a correspondence between the two. Particularly, the  $\gamma$  and herringbone classes correspond to identified clusters - demonstrating the potential of clustering algorithms to identify meaningful automated groupings of structures. The 2D mapping here was generated by dimensionality reduction using the sketch-map algorithm [61]. Image adapted with permission of the Royal Society of Chemistry from ref [62]

These findings suggest potential for machine learning in accelerating structure classification and thus identification of structure-function relationships. However, the techniques are limited, for instance not all structure sets can be effectively clustered, as was found for a second azapentacene [62].

As ever, there is also a sensitivity to choices in the implementation of the algorithms. Particularly, the underlying descriptors that define the initial landscape is important. For instance the utility and results of dimensionality reduction have been found to depend upon the choice of initial descriptor

[67] and the individual parameters of these descriptors [13]. That is, that the reduced mapping is, as one might expect, dependent upon the specific choice of full-dimensional mapping and so this must be chosen well.

In the case of CSP landscapes, this ‘full dimensional mapping’ is usually calculated from the prediction set. There is little in terms of landscape representation that can be usefully and efficiently achieved using merely lists of atomic co-ordinates or cell parameters - even when implementing dimensionality reduction. Instead, sets of alternative descriptors are obtained to describe the set. These new descriptor sets themselves often still form a high-dimensional space - the starting point of dimensionality reduction. Descriptors that have been used to usefully define the initial landscape include Smooth Overlap Of Atomic Positions (SOAP) [1, 13, 62, 63, 67, 68] descriptors (Section 2.4.3) and descriptors capturing the topological features of pores in the crystals [66, 67].

Another use for machine learning in landscape analysis is in identifying which predicted crystal structures will be synthesisable. As discussed previously, ML can of course be used in the calculation of highly accurate energetic rankings for this purpose. However, this is not the only use of ML in this context. Another method, the springboard for the work in this thesis, is the Generalized Convex Hull (GCH) [1] (Section 2.4), which is designed to identify synthesisable crystal structures - specifically those that, even if not intuitively considered synthesisable by consideration of energetic rankings alone, may be stabilisable by application of some experimental constraint. This approach is an adaptation of conventional convex hull approaches. It uses unsupervised machine learning *via* dimensionality reduction to select optimal descriptors to use alongside energy to plot the landscape and construct the hull. The original GCH implementation also further addressed the uncertainty in predicted energies and structural features via methods incorporating Kernel Ridge Regression (KRR).

The GCH approach has been tested on varying systems, successfully identifying as stabilisable predicted structures corresponding to known structures and identifying important structural variance within the prediction sets. It has, for example, explored a dataset of oxygen-hydrogen binary compounds - confirming composition as the greatest source of structural variance in the set and identifying known stabilisable structures - including oxygen structures stabilisable by magnetic fields. In the area of organic molecular materials design, the GCH was shown to recover both known crystal structures of pentacene and structural analogues of stable azapentacene structures. Importantly, the method has shown potential to more efficiently identify stabilisable structures,

while discriminating against other candidates (other predicted structures) as compared to the conventional convex hull approach. A GCH exploration of hydrogen phases identified as stabilisable known high-pressure phases alongside analogues of low-pressure hydrogen phases - all within a set of just 81 candidate structures. Identifying these using the conventional convex hull would have gathered a pool of over 2000 candidates. These findings suggest that the stability predictions from the GCH could offer useful insight to experimentalists, identifying candidates more efficiently and identifying the key areas of structural diversity in the landscapes [1].

However, there are limitations to the approach. The method has been developed primarily with a view to inorganic materials discovery, and while organic molecular materials are touched upon, the investigation into the applicability of the GCH for such systems is limited. Further, the particular implementation of the approach incorporates steps and calculations that would not appear reasonable for investigation of molecular crystals (see Section 3.2). This suggests that adaptation of the method would be needed to address these issues before the method can reliably benefit molecular materials discovery.

### 1.3 Project Motivations

The project is motivated by the discussed need to address over-prediction and identify synthesisable structures during analysis of CSP landscapes. Particularly, it seeks to act on the issue in a manner appropriate to molecular crystals. The promise of the GCH method drives investigation to address its flaws, namely its limited testing and poor applicability to organic molecular crystals.

Building upon and adapting the existing method and codebase, the work aims to formulate an adapted GCH method, theoretically reasonable and appropriate for use in the organic molecular community. Further, it aims to investigate the importance of these adjustments with respect to the impact upon the methods' potential in a molecular materials discovery workflow. This is to explore whether such changes to this and similar methods arising from the inorganic community are needed, or whether the existing methods may be transferable.

There is further a gap in the literature with regard to comparison of different solutions to identifying synthesisable structures from prediction sets - with work often defaulting to the use of low-energy windows or selecting a single approach without investigation of alternatives. The project aims to address this by comparing the results of different approaches to synthesisable structure identification across a range of organic molecular crystal prediction sets.

The work thus aims to draw conclusions as to the relative performance of common approaches - including convex hull based approaches - to identifying synthesisable molecular crystal structures, investigating the performance and limitations of the methods and the potential for improvement through theoretically reasonable adaptations.

## 1.4 Thesis Overview

This thesis comprises six main components:

- The first component (Chapters 1 & 2) acts as an introductory section to lay the groundwork for the remainder of the thesis. Chapter 1 introduced the research context in the preceding literature and background section. Chapter 2 aims to explain the main computational concepts and methods used within the research to aid understanding of the work and results presented.
- The second component (Chapter 3) discusses the primary development task of the research - demonstrating the work to adapt the average SOAP kernel such that it may suitably describe the similarity of molecular crystal structures - with a view to using the kernels in the GCH approach to identifying stabilisable (and therefore likely synthesisable) crystal structures.
- The third component (Chapter 4) discusses work required in order to obtain data used in assessment of the success of the kernel adaptations, covering the crystal structure predictions performed to obtain key structural datasets used.
- The fourth component (Chapters 5, 6, & 7) is the most important results section - covering work to assess the impact of kernel adaptations. Chapter 5 will demonstrate work to compare the effectiveness of landscape analysis methods in identifying synthesisable candidate structures - including approaches using the adapted and average global SOAP kernels. Chapter 6 introduces work to assess the impact of kernel adaptations on the ability to derive interpretable structural descriptors for CSP structure sets and Chapter 7 aims to further compare the adapted and original kernel constructions - discussing investigation of total energy prediction via the adapted and average kernels.
- The fifth component (Chapter 8) explores a proof of concept for fast trial structure generation in CSP. It investigates the use of seeding the CSP from structural analogues of previously predicted crystal structures of similar molecules. Brief investigation is also made into application of the GCH, among other methods of identifying synthesisable structures, in selecting candidates for analogue formation.
- The sixth and final component (Chapter 9) is the conclusion of the thesis. It will summarise the research, highlighting its main achievements and limitations, and discuss several possible future projects to expand the work presented.



## **Chapter 2**

# **Theory and Methods**

### **2.1 Overview**

This chapter introduces the key theoretical concepts and methods underlying the work in this thesis  
- with a focus on the tools and approaches most crucial to the project.

## 2.2 Crystal Structure Prediction Workflows

As discussed in Section 1.2.1 there are many approaches to CSP, but most molecular CSP workflows proceed by first generating a collection of trial crystal structures to sample the full structure space, before lattice-energy minimising those trial structures to find the nearest local thermodynamic minima on the structure-energy surface.

All **in-house** CSP performed for this thesis was conducted in CSPy [20]. CSPy is an in-house software developed and maintained by members of the Day Group at the University of Southampton. A release version of this software, containing much of the CSPy functionality (mol-CSPy [20, 69]) is now publicly available. However, work performed here used an earlier version of the development codebase and did not implement the open-source mol-CSPy.

In CSPy, there are two key prediction workflows used - ‘Rigid-CSP’ and ‘Flexible-CSP’. The simplest of these is the Rigid-CSP workflow, which is applied to molecules that could be defined as ‘rigid’. This refers to molecules in which there is little to no flexibility - for instance molecules having few rotatable bonds or significant hindrance, such as steric factors, to bond rotation. Ideally, molecules treated as rigid should also lack other forms of flexibility such as the tendency to ring-puckering. If a molecule is inflexible in this way, it can be safely assumed that it will likely only adopt a single 3D conformation - even in the presence of intermolecular forces in the crystal.

If a molecule can be treated as rigid, this simplifies the CSP workflow - as crystal structure predictions can be made assuming a single fixed conformation of the underlying molecule. This reduces the sampling problem, as intramolecular degrees of freedom need not be sampled. In the Rigid-CSP workflow used in CSPy, there are three key steps (Figure 2.1):

1. An optimal 3D conformation of the molecule is calculated
2. Trial crystal structures of the molecule in that conformation are quasi-randomly generated
3. Trial structures are lattice energy minimised using inter-atomic forcefields and distributed atomic multipoles

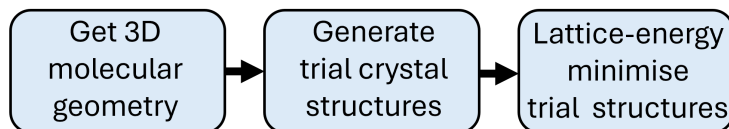


Figure 2.1: Simple flowchart of the basic process used in Rigid-CSP

### 2.2.1 Getting the Molecular Conformation

The ‘optimal 3D conformation’ of the molecule in the rigid workflow is taken to be the gas phase conformer of the molecule. The gas phase conformer is obtained by geometry optimising the isolated molecule using Density Functional Theory (DFT) implemented in quantum chemistry calculation software Gaussian09 [70].

### 2.2.2 Density Functional Theory

Density Functional Theory (DFT) is a popular electronic structure method used for energy calculations. Most electronic structure methods revolve around attempts to find the energy of a system by approximating solutions to the Schrödinger equation:

$$H\psi = E\psi \quad (2.1)$$

where  $\psi$  is a wave-function defining the system,  $E$  is the energy of the system, and  $H$  is the hamiltonian operator. The Hamiltonian operator includes terms to exactly calculate all contributions to the energy of a system - including kinetic energy, nuclei-electron interactions, and electron-electron interactions. Methods seek to find wavefunctions describing the system, and the corresponding energies, that can approximately satisfy the Schrödinger equation.

The concept of DFT revolves around reformulating this problem such that the hamiltonian can be treated as the sum of one-electron hamiltonians, which each ignore electron-electron interaction contributions to the energy. This simplified problem could, in principle, be solved exactly when given a known density. However, it is itself an approximation. To correct for the neglect of electron-electron interactions, an approximate term - the exchange correlation potential - is added. This approximate exchange correlation term is determined by the chosen exchange correlation functional. There are many different exchange correlation functionals that can be used. These are parameterised functionals that estimate the exchange correlation potential based on a given approximated electron density.

Once a functional has been chosen, the ground-state wave-function and energy defining the system can be found. In reality, this is not calculated exactly as this would require the full electron density to be known. However, a useful principle - the variational principle - states that the **true** ground state energy can never exceed the energy corresponding to an approximate wave-function. Drawing from this, it can be said that the approximate wave-function that gives the lowest energy will be the closest approximation. Therefore, DFT energies are calculated using a self-consistent field method - in which an initial approximation of the wave-function is iteratively updated until the corresponding estimated energies converge.[71].

The approximate wavefunction for the system is a slater determinant of molecular orbitals. Those molecular orbitals must satisfy the Kohn-Sham equations [71]:

$$\hat{h}_i^{KS} \chi_i = \epsilon_i \chi_i \quad (2.2)$$

where  $\chi$  denotes the molecular orbital,  $\epsilon$  is the energy of the molecular orbital, and  $i$  indexes the molecular orbital.

The Kohn-Sham equations can be seen as a series of one-electron analogues of the Schrödinger equation, which employ the aforementioned one-electron non-interacting hamiltonians  $\hat{h}_i^{KS}$ :

$$\hat{h}_i^{KS} = -\frac{1}{2}\nabla_i^2 - \sum_k^{nuclei} \frac{Z_k}{|\mathbf{r}_i - \mathbf{r}_k|} + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} d\mathbf{r}' + V_{xc} \quad (2.3)$$

where  $k$  indexes the nuclei, with  $Z$  being the nucleic charge,  $\mathbf{r}$  represents position,  $\nabla^2$  is the Laplacian operator, and  $V_{xc}$  is the exchange correlation potential.

The molecular orbitals are constructed as linear combinations of atomic orbitals. The atomic orbital functions are pre-defined by a chosen basis set, but the coefficients of their linear combinations to form molecular orbitals can be varied - and it is these coefficients which are updated to form new approximate wavefunctions. [71]

Basis sets are collections of basis functions that can be used to define the wave-functions of atomic orbitals. Most commonly in DFT for calculations for molecules, Gaussian type basis functions are used. These describe atomic orbitals using sets of Gaussian functions, with angular momenta appropriate to the orbital they are intended to describe [71, 72].

The approach used for DFT calculations for molecules in this work primarily used the PBE0 functional [73, 74] and the 6-311G\*\* basis set [75]. PBE0 is a hybrid functional that aims to

produce high-quality results by combining parameterised approximations with some degree of calculated exact exchange [71]. Calculations also incorporated an empirical dispersion correction to better account for long-range dispersion effects, which are poorly handled in uncorrected DFT with common functionals[76].

DFT methods for calculation of energies and forces can then be coupled with gradient descent approaches to find not just the intramolecular energy of any molecular conformation - but the conformation that minimises intramolecular energy. [71]

### 2.2.3 Quasi-random Sampling

In this work, trial crystal structures of the optimised molecule are generated by quasi-random sampling. Similarly to random sampling, this approach is designed to ‘blindly’ sample the structure space, relying on the sheer extent of the sampling to ensure that the desired synthesisable structures are found. However, the method advances upon fully random methods by ensuring that the sampling is more even - i.e that no one region of space is arbitrarily over/under sampled. This is achieved via use of sobol sequences - a type of low-discrepancy sequence. Sets of sobol vectors contain elements selected such that each dimension of the corresponding space is quasi-randomly sampled. The elements of each sobol vector are used to determine the key degrees of freedom defining the crystal. These degrees of freedom are:

- (Up to) three unit cell angles
- (Up to) three unit cell lengths
- Three parameters defining the centroid positions of each asymmetric unit molecule
- The orientations of each asymmetric unit molecule

The sobol vector values are not in all cases directly equal to any one parameter they are used to define, but where necessary are used in conjunction with one another and with additional formulae to ensure that the parameters are not only quasi-randomly sampled, but also define physically reasonable crystal structures. This includes constraints to ensure that unit cell angles fall within a reasonable range and that the unit cell volume is appropriate for the molecules it contains [20]. For more details see reference [20].

Finally, for each trial crystal, the remaining unit cell molecules - i.e the ‘other copies’ of the asymmetric unit are positioned by application of the symmetry operations of the assigned space

group.

An additional step of the structure-generation process iteratively increases unit cell lengths so as to relieve any unreasonably close intermolecular contacts. A ‘reasonable-volume’ criterion ensures that this does not create unfeasible structures.

New structures are continually generated and optimised until the required number of successfully optimised structures in each space group has been obtained.[20]

#### 2.2.4 Optimisation of Trial Structures

The trial structures generated, given their quasi-random nature, are unlikely to represent local minima on a structure-energy landscape. Therefore all trial structures need to undergo geometry optimisation. Due to the number of structures to be optimised, and the likely extent of their deviation from the corresponding minima, the methods used must be of low computational cost.

For the simplest cases in this work, implementations of Rigid-CSP, initial geometry optimisations seek to find the structure that minimises the lattice energy. That is, that they seek to find minima of the **intermolecular energy**  $U_{inter}$ . By coupling calculations of  $U_{inter}$  with gradient descent approaches, the local minima to which each trial structure optimises can be found. Crucially, in lattice energy minimisations in this work, the molecules are held rigid, and molecular conformation changes are prohibited.

In this work, most calculations of  $U_{inter}$  are performed using a combination of pairwise interatomic force fields with distributed atomic multipoles - to handle the contribution from permanent electrostatics.

#### 2.2.5 Force Fields

Pairwise interatomic force fields describe the intermolecular potential between two molecules as a sum of the pairwise attractive and repulsive components between all pairs of atoms of the two molecules. The forcefields used in this work use exp-6 potentials [77], which take the form:

$$U_{inter} = \frac{1}{2} \sum_{ik} U_{ik} = \frac{1}{2} \sum_{ik} (A_{pq} \exp(-B_{pq} R_{ik}) - \frac{C_{pq}}{R_{ik}^6}) \quad (2.4)$$

where  $A$ ,  $B$ , and  $C$  are parameters,  $R$  is the inter-atomic distance,  $i$  and  $k$  index the atoms, and  $p$  and  $q$  specify the atom species of  $i$  and  $k$  respectively[24].

These inter-atomic potentials describe the contributions to the intermolecular energy due to exchange-repulsion and dispersion. These contributions depend upon the inter-atomic distance and the species of the atoms involved. The empirical interatomic forcefield, FIT [78] used in this work has been parameterised by fitting to experimental data to derive parameters - specific to each atom type - for interactions between atoms sharing the corresponding type. The required mixed-species parameters such as  $A_{pk}$  were then derived using combining rules.

A limitation is that simple parameterisation of the potential fitting parameters to atom-types is not sufficient to describe permanent electrostatic interactions, which are not dependent simply upon atom type but dependent upon the specific molecule and its charge distribution. To account for this, permanent electrostatic contributions to the intermolecular potential are often added as an additional contribution, by calculating the contribution from the interaction of point charges or multipoles.[24] Both approaches are used in this work, but for Rigid-CSP workflows, distributed atomic multipoles were used. All forcefield level geometry optimisations in this work were implemented using DMACRYS [79], an external program for energy calculation and optimisation of organic crystals.

The use of empirical forcefields for lattice energy calculations in molecular crystals is well-established. The FIT forcefield used in this work, when coupled with distributed atomic multipoles has been shown to lead to accurate energy and unit cell length calculations [24, 28], making the methods used here reliable and trustworthy for obtaining an initial structure-energy landscape.

### 2.2.6 Multipoles

Multipole expansions describe the charge distribution of a system about a point - in terms of a series of multipoles of increasing rank - such as monopole, dipole, quadrupole etc. Higher-order terms add more angular detail and more nodes to the multipole expansion, allowing it to describe charge distributions with greater resolution. The summation of this series defines the function for the charge distribution about that point - and the summation of a truncated series describes an approximation to that function. The centres of the multipole expansions as calculated may not be beneficial for interatomic energy calculations, but using Distributed Multipole Analysis (DMA) the multipole centres can be ‘moved’ to more convenient sites [80]. In this work, multipole expansions up to the hexadecapole are used. These are calculated at DFT level - and shifted to the required sites - using external software GDMA [81]. It has been shown that the use of distributed

atomic multipoles in calculation of intermolecular energy leads to more accurate energy calculations than the use of point charges as the sole permanent electrostatic contribution [24].

The calculation of the contribution to the intermolecular energy from the interaction of multipoles of molecules A and B takes the form of a multiple summation - first over contributions from the interactions of pairs of multipoles (across the included ranks) at given sites  $a$  and  $b$  of the molecules A and B. That is, it is a summation over charge-charge, charge-dipole, dipole-dipole etc. interactions at those sites. This is followed by a double summation over the possible  $a$  and  $b$  sites:

$$\begin{aligned} \text{Interaction Operator} = \sum_{a \in A} \sum_{b \in B} [ & T^{ab} \hat{q}^a \hat{q}^b + T_a^{ab} (\hat{q}^a \hat{\mu}_\alpha^b - \hat{\mu}_\alpha^a \hat{q}^b) + \\ & T_{\alpha\beta}^{ab} (\frac{1}{3} \hat{q}^a \hat{\Theta}_{\alpha\beta}^b - \hat{\mu}_\alpha^a \hat{\mu}_\beta^b + \frac{1}{3} \hat{\Theta}_{\alpha\beta}^a \hat{q}^b) + \dots ] \end{aligned} \quad (2.5)$$

where  $\hat{q}^a$ ,  $\hat{\mu}^a$ , and so on represent the operators for the charge and dipole moments etc at site  $a$ . The subscripts  $\alpha$ ,  $\beta$  etc indicate the required element of the vectors/tensors expressed in cartesian space - i.e they can each be  $x, y$ , or  $z$ . There is a further implicit sum - indicated by the Einstein notation [82] - across the the different  $\alpha, \beta$  etc. The bracketed calculations in each term of the summation become increasingly complex for the terms involving higher-ranked multipoles. The  $T_\alpha^{ab}$  etc are tensor terms acting on the vector  $\mathbf{R}$  between the sites [82].

These tensor terms take the general form:

$$T_{\alpha\dots}^{ab} = \frac{1}{4\pi\epsilon_0} \nabla_a \nabla_{\dots} \frac{1}{\mathbf{R}} \quad (2.6)$$

where  $\mathbf{R}$  is vector  $\mathbf{a} - \mathbf{b}$ . [82]

The practicalities of implementing these calculations accurately and efficiently is a complex topic, and is not discussed in depth in this thesis.

## 2.2.7 Summarising the Rigid-CSP Workflow

In short, the Rigid-CSP workflow involves, taking a single molecular conformation optimised at DFT level, creating trial crystal structures of that molecule via quasi-random sampling of unit cell parameters and asymmetric unit positioning/orientation, before optimisation of trial structures using interatomic forcefields and atom-centered multipoles to find the nearest local minima on the forcefield level structure-energy landscape. The workflow is summarised in Figure 2.2



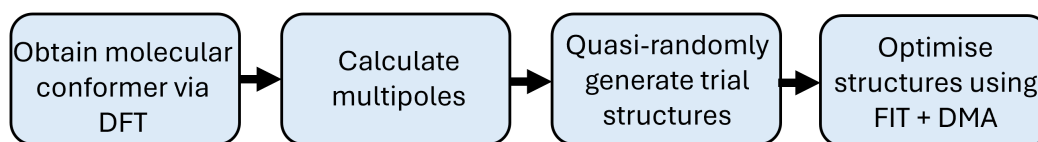


Figure 2.2: Summary of the Rigid-CSP workflow

### 2.2.8 Flexible Crystal Structure Prediction

Where the underlying molecule has flexibility, however, the process is more complex. As it can no longer be assumed that the molecule will adopt a single optimal conformation, sampling must also account for intramolecular degrees of freedom. Further, as intermolecular forces may impact the conformation - additional geometry optimisation steps, allowing flexibility of the molecule must be added. Such re-optimisation steps can also be used after Rigid-CSP, but are more crucial to Flexible-CSP.

Figure 2.3 shows an overview of the Flexible-CSP workflow, which can be viewed as largely using the underlying workflow from Figure 2.2, with additional processes and adaptations.

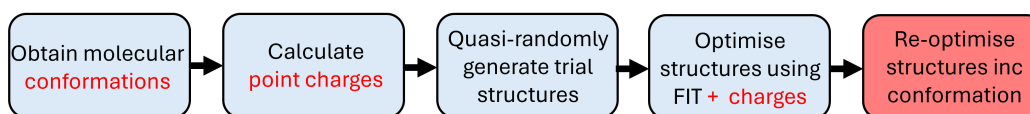


Figure 2.3: Summary of the Flexible-CSP workflow. Differences to the Rigid-CSP approach are highlighted in red

### 2.2.9 Sampling Multiple Conformations

The first step used when sampling molecular conformations in this work is to consider that a molecule may even have several stable **conformers** - i.e there may be several minima on the conformational energy landscape.

There are several ways in which these conformational minima can be sought. One approach which has been used in this work is to attempt to map the conformational landscape by systematically sampling many points on that landscape and calculating the energies. This can be performed using software such as Gaussian09. From the resulting energy surface, the conformers (the local minima) can then be identified. There are also alternative programs designed to identify all conformers. One example is the program CREST [83], which samples conformational space using meta-dynamics

runs which explore perturbation of the molecular geometry, beginning from a single starting conformation. This can be further expanded upon using a developed approach mCREST - which runs multiple CREST searches, from different starting points, to ensure more thorough sampling [84]. mCREST searches have also been used in this work.

Sampling of stable conformers remains insufficient for effective crystal structure prediction of flexible molecules, however. This is because intermolecular forces can influence the molecular conformation, leading molecules to adopt in-crystal conformations which are not conformers of the isolated molecule. The impact of this is such that, even if re-optimisation steps are later applied, deriving an initial landscape that only sampled stable conformers will lead to missed crystal structures in the prediction set [85]. To address this, in this work, conformational sampling is extended using an in-house code MOLDIS, which perturbs conformers by user-determined increments to sample the conformational energy surface around each conformer. The point charges and/or multipole expansions corresponding to each generated conformation are also calculated and stored.

To incorporate this conformational sampling into generation of trial structures, the asymmetric unit molecules are randomly selected from all generated conformations lying within a chosen energy range from the global minimum conformation.

### **2.2.10 Differences in Initial Optimisations**

Due to the additional degrees of freedom in the search, Flexible-CSP workflows require larger numbers of trial structures for effective sampling. This means that steps to reduce computational cost are warranted. Because of this, initial optimisations used in Flexible-CSP in this work used only point charges to calculate the electrostatic contribution to the intermolecular energy. This was assumed to be sufficient for initial optimisations, as more accurate geometries and energies could be obtained from subsequent reoptimisation steps.

A difference in implementation worth noting here is in the derivation of point charges. Simply deriving point charges via a truncated series of multipole moments would result in poor accuracy. Instead, an external program MULFIT is used to obtain Point charges that best reproduce the charge distribution that would be described by a higher-order multipole expansion. [86, 87].

Another key difference in initial optimisations is that, in order to obtain reasonable energetic rankings of the structures, lattice-energy alone is insufficient as the intramolecular energy contribution

- which is no longer consistent - must be accounted for. To do this within CSPy, the intramolecular energy relative to the global minimum conformation ( $U_{intra}^{Rel}$ ) is calculated and this is added to the lattice energy ( $U_{inter}$ ) to provide a final measure of the energy of the crystal structure:

$$U = U_{intra}^{Rel} + U_{inter} \quad (2.7)$$

### 2.2.11 Re-optimisation Steps

As discussed, especially when running Flexible-CSP workflows, further re-optimisation steps are required after initial forcefield level optimisations in order to obtain accurate predictions. These re-optimisation methods should allow changes to the in-crystal molecular conformation during optimisation.

In this work, re-optimisations were typically performed in two steps. First using a cheaper, less accurate method (DFTB or MACE) and then using a more costly but highly accurate method (Periodic DFT).

### 2.2.12 Periodic DFT

The final step of re-optimisations for Flexible-CSP workflows in this thesis uses periodic-DFT. Periodic DFT relies upon the same principles as DFT for molecules (Section 2.2.2), but there are important differences in its implementation. While in theory DFT methods for crystal structures could be made intractable by the infinite size and number of atoms in the crystal structure, fortunate simplifications allow the total energy to be derived by performing calculations only over a single repeating unit.

This arises from the fact that, in a truly periodic crystal, the wavefunctions at a given point and at an equivalent point in the next unit cell must be equal - save for a phase factor - as  $|\Psi|^2$  must be maintained. Bloch's theorem provides a definition of the wave-function that satisfies this is:

$$\psi_k(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} u_k(\mathbf{r}) \quad (2.8)$$

where  $k$  is the 'crystal momentum' and  $u_k(\mathbf{r})$  is some function with same the periodicity as the lattice. It therefore suffices to only perform calculations to find each  $u_k(\mathbf{r})$  in order to derive the wave function across the whole crystal [88].

However, a dependency on  $k$  has been introduced. Each  $k$  can be considered to be a point in the reciprocal lattice and the final overall wave-function is the sum over all unique  $k$ . That is, all  $k$

in the unit cell of the reciprocal lattice - called the first Brillouin zone. There are potentially infinite  $k$  for which to solve this.[88] However, for molecular crystals, the localisation of electrons onto molecules and the large size of the unit cells means that they can be considered insulators and so the important  $u_k(\mathbf{r})$  are only weakly dependent on  $k$  and the first Brillouin zone can be effectively sampled using only a few representative  $k$  points [88, 89].

The construction also suggests an alternative type of basis set to be used in calculations for periodic systems. The term  $e^{i\mathbf{k}\cdot\mathbf{r}}$  defines a plane wave. It is then convenient to also use a basis set composed of plane waves to define each  $u_k(\mathbf{r})$ . Such  $u_k(\mathbf{r})$  can satisfy the periodicity requirements of the wave-function [90].

The geometry optimisation process can proceed much as with DFT for molecules, using self-consistent field and gradient descent methods. In this work, all Periodic DFT calculation were implemented in Vienna Ab Initio Simulation Package (VASP) - a widely used external software designed for ab initio calculations in materials modelling [91–93]. Most used the PBE exchange correlation functional - with dispersion corrections - which has been shown to facilitate high-quality geometry optimisations of molecular crystals with plane-wave basis sets, leading to calculated energy values and unit cell lengths close to experiment [94].

### 2.2.13 DFTB

Density Functional Tight Binding (DFTB) [95, 96] is a method designed to approximate the results of Periodic DFT, at a lower computational cost by incorporating additional parameterisation and tight-binding principles [84]. In this work DFTB calculations are conducted using the advanced implementation of the approach DFTB+ [97].

### 2.2.14 MACE

Recently, use of machine-learned force-fields to approximate the results of high-level calculations has proved successful [29–31]. MACE [98] is one such machine-learning approach. The architecture employs message passing neural networks to train machine learned force fields from many-body atomic environment descriptors. In this work, a pre-trained model MACE-OFF [99], which is trained to reproduce DFT energies of molecules, was employed.

### 2.2.15 Duplicate Removal Methods

The CSP workflows discussed thus far are designed to ensure that all truly possible crystal structures are identified within the prediction set by using extensive sampling. The high number of generated structures leads, in most cases, to production of many duplicate structures after optimisation - i.e each basin on the landscape may have been sampled multiple times [20].

This means that methods must be applied to remove duplicate structures from the set, and obtain the set of **unique** predicted crystal structures, i.e those that are not considered to be ‘approximations’ to the same structure. In this sense, there is not a single rigid definition defining duplicate or unique crystal structures, but rather structures are classified as unique/duplicate based upon varying duplicate removal methods that have been developed in the field to serve the pragmatic purpose of reducing sets of crystal structures to the key representative structures. Many methods have been devised in the field of CSP to perform this duplicate removal.

In this work, a two-step process to duplicate removal was enacted:

1. Removal of the closest duplicates by comparison of predicted Powder X-Ray Diffraction (pXRD) patterns
2. Additional, more effective, duplicate removal based upon geometric overlays

In this thesis, duplicate removal steps may be referred to as ‘clustering’, as is an adopted use of the language within the research group. However, it is important to note that the algorithms involved are not strictly clustering in the traditional sense - such as the HDBSCAN clustering algorithm [100] is - but are simply methods to compare structures one pair at a time - identifying and removing duplicates.

The first step of duplicate removal is a quick, low cost method - appropriate for application to large databases of CSP structures. However, it can only identify a small subset of the present duplicates. In this step, structures are compared based upon their simulated pXRD patterns - predicted using external software package PLATON [101]. After initial filtering based on energy (Usually selecting structure pairs within 1.0 kJ/mol of each other), density (Usually selecting structure pairs within 0.05 g/cm<sup>3</sup>) and cosine similarity a varying parameter - see specified details in Chapter 4 and discussion in Section 4.6) to get the most similar structures, their pXRD patterns (which have been normalised such that the area under the curve is equal to one) are compared using a constrained Dynamic Time Warping (cDTW) method [102, 103]. This revolves around comparing the

peak heights between the two patterns at each position. However, this is performed flexibly, akin to allowing some warping of the patterns to be compared. This essentially allows for shifting of the peak positions so as to increase similarity. This is necessary as pXRD patterns - especially those that are simulated - have very sharp narrow peaks, and so comparisons that do not allow distortion would be overly sensitive to minute differences between the patterns [84]. The method is only known to be reliable when used within tight tolerances, and so can only be used to identify the closest duplicates. In this work, unless otherwise specified, structures were considered duplicates if their respective pXRD patterns had a cDTW distance  $\leq 10^\circ$ . Some further discussion of the thresholds applied in pXRD clustering, especially with regard to comparison to the COMPACK clustering method (see below), is given in Section 4.6

The second, more thorough, stage of duplicate removal uses geometric overlaying of crystal structures to identify duplicates - inspired by the COMPACK algorithm [104]. Structures are compared if suitably similar in energy - and in some cases density. Then the *CrystalPackingSimilarity* feature in an external software the CSD API [105] generates representative clusters of 30 molecules from each of two crystal structures to be compared. It then attempts to overlay these clusters, and if the clusters can be overlaid such that 30/30 molecules overlay within reasonable tolerances (usually 0.2 Å allowed deviation in interatomic distances and 20° deviation in interatomic angles), then the structures are declared to be duplicates. Hydrogen positions are usually ignored in these searches. Whilst it is possible in cases of  $Z' > 1$  for such comparisons to be influenced by the selection of an initial molecule at the centre of the cluster, this effect should be minimal when using suitably large clusters. The results of geometric overlays, i.e the COMPACK algorithm, can also depend upon the size of the cluster used. Testing and experience within the Day group has identified 30 molecules as a suitable cluster size, with increasing cluster size having minimal impact upon results that does not justify the increased cost. 30 molecules could additionally be considered a high-standard for molecular similarity within the field of molecular CSP, being used to classify structures as ‘matches’ for instance in recent CSP blind tests [17, 18] It is important to note that, in this work, duplication removal was enacted where structures presented a 30/30 overlay regardless of the Root Mean Square Distance (RMSD) value defining the imperfection of that overlay - a slightly different implementation to that used in the public software mol-CSPy [69]. In this thesis, such duplicate removal may be referred to as the COMPACK method.

In all cases of identified duplicates, the higher energy duplicate was removed from the structure set and the lower energy structure retained.

## 2.3 Landscape Analysis Methods

### 2.3.1 Overview

As discussed in Section 1.2.2, a typical CSP landscape contains a large number of local minima crystal structures and the prediction set must be filtered to identify the most likely synthesisable structures. There are several approaches that can be taken to do this, but the most pertinent to this thesis are using an energy window/energetic ranking and convex hull methods.

### 2.3.2 Energetic Ranking

Determining synthesisable structures based upon their relative energetic ranking is straightforward. Relative energies for each structure in a prediction set are calculated by subtracting from that structure's energy the energy of the global energy minimum structure in the set:

$$E_x^{Rel} = E_x - \min\{E_i | i \in \text{structure set}\} \quad (2.9)$$

The likely synthesisability of each structure can then be determined by its position in the relative energetic rankings for the structure set - with structures with a lower relative energetic ranking (lower relative energy) being more likely to be synthesisable. In this work, the global energy minimum structure in the set is said to have the rank of 1 and other structures are then ranked in ascending order with increasing relative energy.

### 2.3.3 Convex Hull Methods

As discussed in Section 1.2.2, convex hull based methods identify synthesisable structures from prediction sets based on proximity to a 'convex hull' on the landscape of predicted structures - plotted according to their energy and some structural descriptor(s). Often, especially in inorganic materials discovery, the structural descriptor used is composition [57]. But other constructions, used in this work, employ density as a descriptor or can use machine-learned descriptors as in the Generalised Convex Hull method (Section 2.4).

Figure 2.4 shows a conceptual example of a convex hull. A 'hull' is drawn across the bottom of the set of points. Then, the structures corresponding to points that are vertices of the hull are classed as synthesisable, in that they should be stabilisable under some conditions. The remaining structures are meta-stable with structures closer to the hull being more likely to be stabilised under some conditions than structures further from the hull [1, 57].

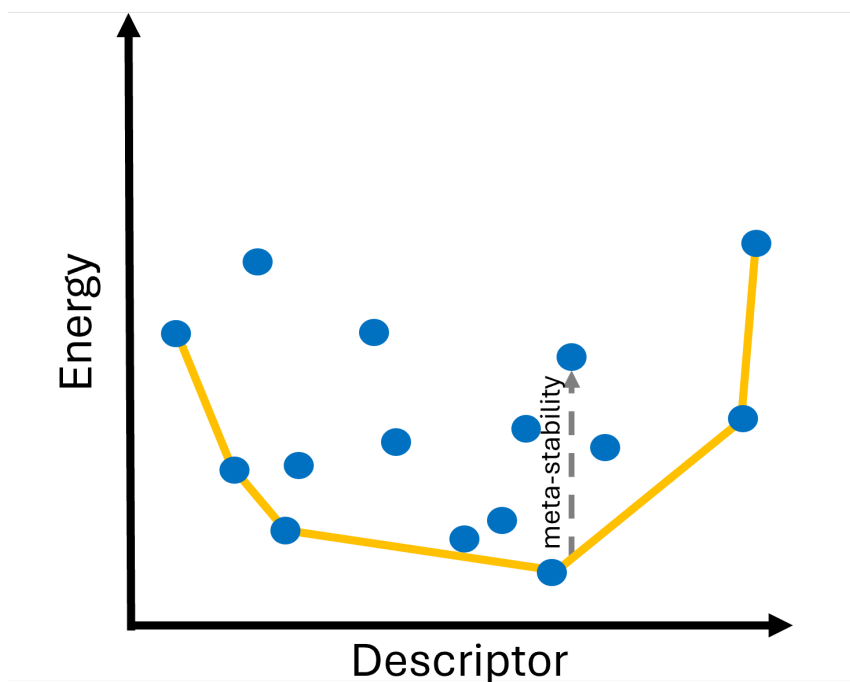


Figure 2.4: A hypothetical example of a convex hull used in materials discovery. Each point represents a predicted crystal structure, and the orange lines indicate the hull. The meta-stability of a predicted structures is given by its height above the hull

The following sections outline how such a convex hull can be calculated from the CSP landscape and how the hull can then be used to analyse stabilisability of the structures.

#### 2.3.4 Defining the Convex Hull

Mathematically, the term ‘convex hull’ describes the smallest convex region of space that can contain a set of points. Often, however, the term is often used to refer just to the boundary of that volume[106]. This differs from that discussed in the context of materials discovery, in which, the chemically useful convex hull is the part of the boundary across the ‘bottom’ of the set of points. This chemical definition will be discussed further in Section 2.3.6 after explanation of the mathematical derivation of the hull.

There are several common algorithms that may be used to calculate the convex hull of a set of points. In this work, convex hulls have been calculated via the SciPy ConvexHull class [107] which uses the Quick Hull algorithm [108]. This approach constructs a hull by iteratively rejecting sets of points that can be confidently classified as non-hull points - until all points have been tested. This set of hull points, alongside related attributes, can be used to define the hull.



The convex hull can be identified for any set of points in any number of dimensions. Only the simplest case - a set of points in two dimensions - is explained in detail here, but the algorithm follows a similar process in higher dimensions. The two-dimensional Quick hull algorithm (Figure 2.5) works as follows [106, 108]:

- The data is partitioned into two sets by the establishing of a line between two extreme points. (The points with the lowest and highest value along one axis). These extreme points are geometrically guaranteed to lie on the hull. Each set can then be searched separately and equivalently, using the process below
- A third most-extreme point is identified- forming a triangle with the previous two extreme points. (Here, the point must be extreme along the orthogonal axis). This point is also guaranteed to lie on the hull - and all points falling within the triangle are guaranteed to be non-hull points and are removed from the search
- The search then continues with the construction of a further two triangles each formed from the previous 'third' extreme point, one of the pair of initial selected extreme points, and another new extreme point - again chosen to be the most extreme along the orthogonal direction to the previous step. The corners of the triangles are defined as hull points and points within the triangles are rejected.
- The search continues iteratively in this way. At each step two new triangles are formed incorporating extreme points from the previous two steps and a third point that is most extreme along alternating axes. This continues until all points have been searched or rejected

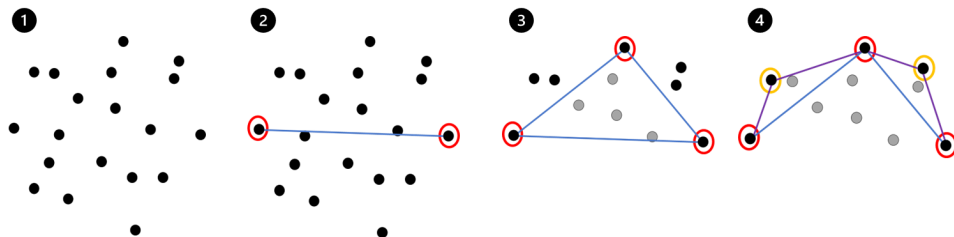


Figure 2.5: Graphic of the quickhull algorithm for a two-dimensional set of points, based upon an iterative process of defining triangles with 'extreme' vertices - and thereby selecting hull points (red) and non-hull points (grey)

### 2.3.5 Vertices, Facets and Equations

A derived convex hull of a set of points in  $d$  dimensions has several elements within its construction. These elements can serve different purposes in analysis of CSP landscapes and can be obtained as attributes of the SciPy ConvexHull class. Examples of these elements for a set of points in two dimensions are shown in Figure 2.6.

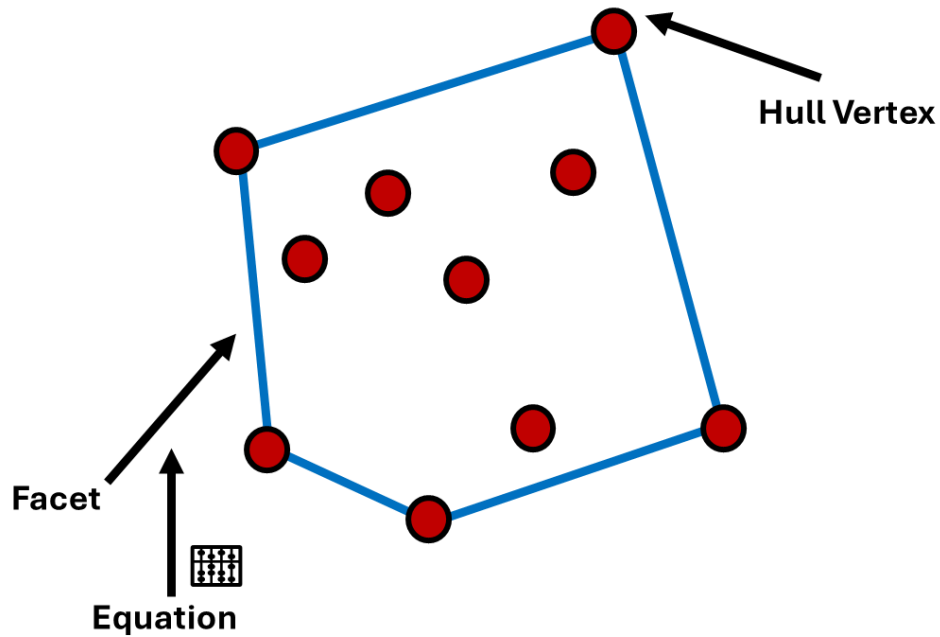


Figure 2.6: Examples of important attributes of the convex hull - as they relate to the hull on a 2D set of points

The vertices of a convex hull are the extreme points lying on the boundary of the convex region containing all the points. The facets of the hull are the  $d - 1$  dimensional simplices defining the hull boundary. Each of these is a  $d - 1$  dimensional simplex defined by  $d$  hull vertices. The easiest case to imagine here is the case of the two-dimensional set of points as in Figure 2.6. In such an instance the facets are the edges joining neighbouring hull vertices. Another attribute is the equations, which are the hyperplane equations corresponding to each facet. That is that they are the equations of the  $d - 1$  planes in which each facet lies. In the case of a 2D set of points, such equations would give the line defining each facet - extended infinitely. To be precise, the equation attribute returned by the SciPy ConvexHull class is defined by the normal vector to the hyperplane and the offset. [57, 109].

### 2.3.6 The Chemically Relevant Convex Hull

In the context of chemistry, the convex hull is most often defined as being the part of the hull which is low in energy. This is the part of the hull boundary that goes across the bottom of the set of points when the vertical axis defines energy (Figure 2.4). Such a hull is formed of a subset of the true hull facets. This is the subset of facets for which the component of the normal (as directed out of the hull) parallel to the energy axis is negative. The chemically relevant part of the hull can therefore be obtained from the dataset using the Scipy ConvexHull class and its attributes. Unless otherwise specified, all ‘convex hulls’ used and discussed in this work refer to the chemically relevant convex hull.

### 2.3.7 Dressed Energies

The stabilisability of a structure, as defined by convex hull methods, is measured by its energy relative to the hull. This is called the dressed energy. The dressed energy corresponds to the ‘vertical height’ above the hull, where vertical is defined as being parallel to the energy axis [57, 110].

In this work, the dressed energies are calculated using code adapted from the Directional Convex Hull (DCH) class [110] in the scikitlearn library [111]. This implements functions within SciPy’s ConvexHull class [107]. It takes all the facets making up the chemically relevant hull and the equations of the hyperplanes that define those facets. For each structure, it then calculates the vertical distance between that structure and each hyperplane - taking the dressed energy to be the minimum of those distances. This gives the shortest vertical distance between each point and the defined convex hull (Figure 2.7).

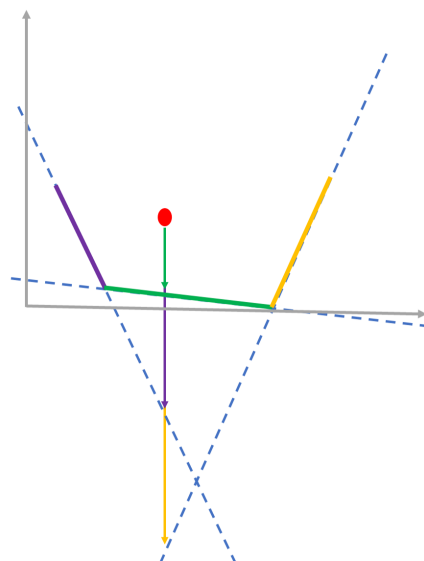


Figure 2.7: Diagram showing how dressed energies are calculated for a hypothetical hull using energy (vertical axis) and one structural descriptor. The equations, shown by dashed lines, corresponding to the facets of the hull are taken, the distances from a given point to these planes are calculated and the minima of the distances are taken. Colour coding matches the measured distance to the corresponding facet

There are multiple structures with a dressed energy of zero -i.e the structures corresponding to hull vertices. Therefore, when ranking structures by dressed energy, it is important to define what is meant by the ranking. In this work, all hull vertex structures are said to share the rank of 1 and other structures are then ranked in ascending order with increasing dressed energy.

### 2.3.8 Candidate Pools

To prove useful for materials discovery, an approach for identifying synthesisable structures should pick out those structures within as small as possible a pool of candidate structures - in order to limit the time/resources needed to test the candidates. Therefore, an important factor in the effectiveness of a convex hull construction is the size of this candidate pool. For simplicity, in this thesis, the size of the candidate pool will be referred to simply by the term ‘candidate pool’. For the purposes of testing on known systems the candidate pool for a convex hull construction is, unless otherwise stated, taken to be the number of structures in the dressed energy window required to contain all known polymorphs. The equivalent candidate pool for approaches using simple energetic rankings would be the number of structures in the lattice/total energy window needed to capture the polymorphs.

## 2.4 Generalised Convex Hull Methods

### 2.4.1 Overview

The principal convex hull based approach used in this thesis is the Generalised Convex Hull. As discussed in Section 1.2.2, this method determines the stabilisability, of predicted crystal structures based upon their proximity to a convex hull on a landscape of structures defined by their energy and some machine-learned structural descriptor(s). This is designed to provide a convex-hull approach to identifying stabilisable structures, without the researcher bias of using manually-selected descriptors. The identified structures are considered stabilisable with regard to some experimental constraint related to the descriptors used to construct the hull. That is, that if the necessary constraint were controlled, it is suggested that the identified structures may be selectively stabilisable under some value of that constraint. In this thesis, when discussing convex hull based methods and their results the term 'stabilisable' therefore refers to these structures, which are those considered likely synthesisable when using the GCH approach.

### 2.4.2 The Generalised Convex Hull Workflow

The GCH workflow takes the following basic steps, also summarised in Figure 2.8:

1. Compute a similarity kernel matrix defining the pairwise similarity of all structures in the prediction set - in this work SOAP kernels are used
2. By performing kernel Principal Component Analysis (kPCA) using this kernel, derive optimal machine-learned (ML) descriptors
3. Obtain the dataset in which each sample corresponds to a predicted structure and the features are that structure's energy and its value(s) for one or more of the top-ranked ML descriptors
4. Compute the chemically relevant convex hull of that dataset
5. Calculate the dressed energies of each predicted structure relative to the computed convex hull

[1, 57]

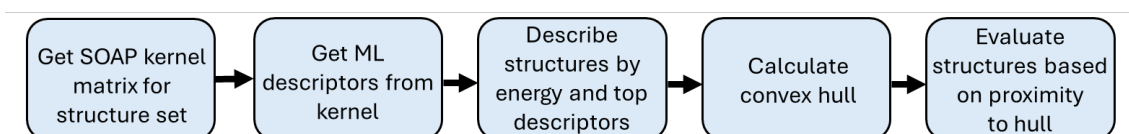


Figure 2.8: Workflow used in the GCH approach

### 2.4.3 Smooth Overlap of Atomic Positions (SOAP) Descriptors

The first step of the GCH process, as performed in this work, requires computation of a SOAP similarity kernel. This is a construction that compares the similarity of each pair of structures based upon their SOAP **descriptors**. It is therefore pertinent to begin discussion of the GCH implementation by introducing these descriptors.

SOAP descriptors define a structure by its local atom-centred environments. The basis of this is to treat each atom as being at the centre of a sphere (of a given cut-off radius) that defines the important ‘local environment’ to the atom. The positions and species of atoms within that sphere determine that local environment.

However, whilst conceptually it appears sensible to define a local environment in this way, in practice the construction is not useful. This is because the atomic positions are tightly defined. This would lead to very sharp peaks of atomic density in the local environment. This level of precision is not reasonable, given the uncertainty in atomic positions, and would make useful comparison of atomic environments difficult. Therefore, a smoothing process is used, in which the local environments are defined not merely by a list of atomic positions within the cut-off, but using Gaussian functions centered on each atom in the environment. (Figure 2.9) The summing of these Gaussian smoothed atomic densities within the cut-off can then reasonably describe the local environment.

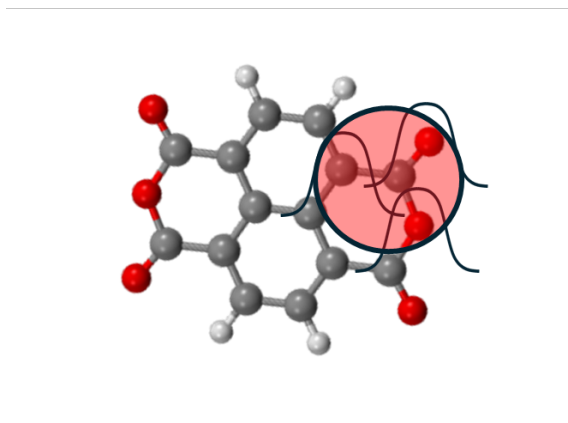


Figure 2.9: Conceptual graphic of analysis of a structure via SOAP descriptors. Red circle represents the selected local environment of the central atom - and all neighbouring atoms within the environment have been decorated with a Gaussian function

The SOAP descriptor for a single local environment in the simplest case - where only one neigh-

bouring atom species is present- is given by:

$$\rho_x(\mathbf{r}) = \sum_{i \in x} e^{-\frac{\|\mathbf{r}-\mathbf{r}_i\|^2}{\sigma}} \quad (2.10)$$

where  $x$  is the local environment,  $\mathbf{r}$  defines position,  $\sigma$  is the width of the Gaussian, and the index  $i$  specifies the atom within the local environment. [57, 68, 112]

However, there are of course usually multiple atomic species in a given crystal structure. The information of the species of atoms within the local environment is retained when calculating SOAP descriptors by essentially ‘tracking/labelling’ each Gaussian function that contributes to the local environment by the corresponding atomic species. This allows for separate calculation of the smoothed atomic densities of each species within the environment. The end construction then computes the smoothed atomic density contributions arising from each  $\alpha$ - $\beta$  species pair in the environment. Those contributions combined define each local environment [112].

Practically, speaking, to define the SOAP descriptors in their most convenient form, they are expanded in a basis of spherical harmonics and radial basis functions, facilitating their expression as power spectrum vectors, where each component corresponds to a term in the power spectrum of the SOAP descriptor. Initially, one power spectrum vector is obtained per  $\alpha$ - $\beta$  species pair. These are called **partial power spectra**. This vector has components corresponding to pairs of basis functions in the respective expansions. Each component is given by:

$$\rho(x)_{b_1 b_2 l}^{\alpha \beta} = \pi \sqrt{\frac{8}{2l+1}} \sum_m ((c_{b_1 l m}^{\alpha})^{\dagger} c_{b_2 l m}^{\beta}) \quad (2.11)$$

where  $b_1, b_2, m$ , and  $l$  index basis functions of the expansion and the corresponding expansion coefficients  $c_{b l m}^s$  are taken from the description of the atomic density of species  $s$  in the environment. [112]

The full derivations have not been provided here, but can be seen in references [112] and [57]. The partial power spectra are then concatenated, to form one SOAP vector  $\boldsymbol{\rho}(x)$  defining the entire environment of the central atom. [112]. The length of a SOAP vector will therefore depend upon the number of pairs of atom species present across a set of structures for which the descriptors are being calculated. In this work, each partial power spectra will describe 448 co-ordinates, this length being determined by the other parameters chosen. These can be seen in Appendix A.

### 2.4.4 SOAP kernels

SOAP descriptors are most often used not in isolation, but rather as a means by which to determine the similarity of structures, via a SOAP kernel. This is what is required for the GCH approach.

Luckily, defining the SOAP descriptors as power spectra is in Equation 2.11 greatly simplifies the similarity calculations. The local similarity kernel defining the similarity of any two **atomic** environments  $A_s$  and  $B_t$  can be given by the simple linear kernel (i.e the dot product) of the corresponding power spectra vectors:

$$k(A_s, B_t) = \boldsymbol{\rho}(A_s) \cdot \boldsymbol{\rho}(B_t) \quad (2.12)$$

This is conventionally normalised such that the dot product between two identical environments is equal to one. [57, 112]. Within the kernel calculation code used in this work, as the power spectrum vectors have been arranged via consistently ordered concatenation of the partial power spectra, the dot product local kernel only considers the similarity of matching  $\alpha$ - $\beta$  environment densities between atomic environments to be compared [112, 113]. This results in a ‘species-aware’ comparison of local environments. However, it is important to note here that this restriction extends only to the compared terms of the respective environments’ power spectrum vectors and does not inherently enforce any limitation upon comparison of environments in which the central atom species differ.

Whilst the definition of the similarity of two **atomic** environments (the local SOAP kernel) is simple, complexity is introduced when extending the concept to compare structures composed of multiple atoms - such as molecules or crystals. In such cases, the similarity is defined via a **global kernel** - formed by combining local kernels. Conventionally, the global kernel between two crystal structures A and B is some function of all the local kernels between the atoms of structure A and the atoms of structure B.

There are several conventionally defined global kernels, which implement different functions to combine the local kernels. The simplest of these is the average global SOAP kernel, in which the function is merely the arithmetic mean.:

$$K(A, B) = \frac{\sum_{s=0}^{S-1} \sum_{t=0}^{T-1} k(A_s, B_t)}{ST} \quad (2.13)$$

normalising according to:

$$K(A, B) = \frac{K(A, B)}{\sqrt{K(A, A)K(B, B)}} \quad (2.14)$$



where  $s$  and  $t$  denote atom indices within the respective unit cells,  $S$  is the number of atoms in unit cell of A, and  $T$  is the number of atoms in the unit cell of B [1, 112].

The average kernel is a somewhat naive measure of the similarity - however it is a very quick and efficient measure. Particularly given the following identity:

$$K(A, B) = \frac{\sum_{s=0}^{S-1} \sum_{t=0}^{T-1} \boldsymbol{\rho}(A_s) \cdot \boldsymbol{\rho}(B_t)}{ST} = \left( \frac{1}{S} \sum_s \boldsymbol{\rho}(A_s) \right) \cdot \left( \frac{1}{T} \sum_t \boldsymbol{\rho}(B_t) \right) \quad (2.15)$$

which allows for the SOAP descriptors to be calculated once for each atom in the structure, those descriptors to be averaged - and for the global average kernel to be calculated using a single dot-product operation [57].

Not all global kernel constructions are so simplistic. For instance the BestMatch and ReMatch kernels, implement functions incorporating identification of the best 'matching' of atomic environments between two structures - and combining only the corresponding local kernels to form the final global kernel. The ReMatch kernel then adds an additional contribution from the average kernel.[112]

The choice of global kernel to use is not always obvious. Selecting an appropriate global kernel to be used when comparing molecular crystals is the main work of this thesis. Discussion of the relative benefits and drawbacks of different kernel constructions begins in Chapter 3.

Once the chosen global SOAP kernel has been computed for every structure pair in a prediction set of  $N$  structures, the  $N \times N$  pairwise kernel matrix can be constructed - in which each matrix entry  $ij$  gives the similarity of structure  $i$  to structure  $j$ .

### 2.4.5 Kernel Principal Component Analysis

Once a global SOAP kernel matrix has been obtained, the next step in the GCH approach is to obtain ML descriptors of the structures via kernel Principal Component Analysis. The method is an extension of Principal Component Analysis.

Principal Component Analysis (PCA) is an unsupervised machine learning technique for dimensionality reduction (See Section 1.2.2). In essence, it aims to create a smaller number of new dimensions to describe a dataset by forming linear combinations of the initial dimensions. (Figure 2.10) These new dimensions (principal components) are constructed so as to maximise variance

of the dataset across each component. It also ensures that all principal components are orthogonal to one another - and so can form a new coordinate system to define the data [114].

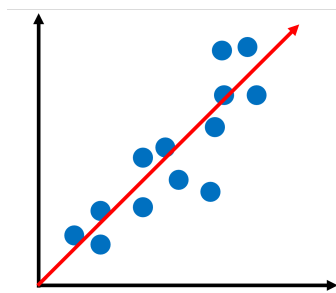


Figure 2.10: Conceptual example of ‘new descriptors’ that may be found via PCA approaches. The red arrow indicates a new descriptor, capturing much of the variance of the original dataset

Mathematically, finding these orthogonal linear combinations of the original variables is equivalent to finding the eigenvectors of the covariance matrix of the data. The corresponding eigenvalues quantify the variance associated with the principal component they define. Thus solving the eigenvector/eigenvalue problem and selecting the vectors with the highest eigenvalues finds the most useful principal components [115].

However, it may not always be effective to define principal components in this way as the important data may not always lie on a linear manifold -i.e linear combinations of the variables may not be able to capture the key variance in the dataset. In these cases, more useful principal components could be derived using the ‘kernel trick’. The crux of the kernel trick is capturing non-linear relationships in the data by exploring the information that **could** have been described by a linear relationship had the data been projected into a higher dimensional space.

KPCA is a PCA approach that employs the kernel trick. To achieve this, the eigenvectors/eigenvalues defining the principal components can be derived from a valid kernel matrix for the data in place of the covariance matrix.[1, 57, 116]. In this work, for implementing the GCH, the kernel matrix used is the global SOAP kernel matrix for the data.[1, 57].

The principal components corresponding to the highest eigenvalues include the maximum structural variance - i.e the range of values seen for a given component is greater for components with greater eigenvalues. The components can be ranked by their eigenvalues (and their relative importance compared) by looking at the series of eigenvalues across the set - called the eigenspectrum. This allows for PCA/kPCA approaches to be used for dimensionality reduction because the data

can be represented -with little information loss- using only the top/highly-ranked components. It is these top principal components that are used as ML descriptors in the GCH approach.[57, 115]

Whilst the eigenvectors defining the principal components can be obtained from the kernel matrix, this does not complete the transformation of the data. The principal components can be thought of as simply the new directions or axes - and the values for each point along these directions need to be found. The set of datapoints along the principal components is called the projection.

Obtaining the projection from kPCA is a mathematical transformation using the eigenvectors, eigenvalues and the original kernel matrix (Equation 2.16). First, the top  $d$  eigenvectors must be selected, where  $d$  is the desired dimensionality of the projection. Then, each eigenvector in that set must be multiplied by the reciprocal of the square-root of its eigenvalue. This is performed by multiplication of a vector containing the reciprocal square root values and a matrix containing the eigenvectors. Lastly the resulting matrix front multiplies the original kernel matrix to give the projection:

$$\text{Projection} = \boldsymbol{\lambda}^{-\frac{1}{2}} \mathbf{U}_d \mathbf{K} \quad (2.16)$$

where  $\boldsymbol{\lambda}^{-\frac{1}{2}}$  is a vector containing the reciprocals of the square roots of the  $d$  eigenvalues,  $\mathbf{U}_d$  is the matrix formed of the  $d$  eigenvectors, and  $\mathbf{K}$  is the Kernel matrix [57].

#### 2.4.6 Constructing and Using the Hull

Lastly, the GCH approach requires construction of a convex hull on a landscape of the structures defined by their energy and their projection onto the chosen principal components. This is performed using the approach discussed in Section 2.3.4. Once the generalised convex hull has been constructed, the stabilisability of structures - with regard to some constraint - can be determined.

Strictly speaking, the GCH analysis of stabilisability of structures relies upon the assumption that the relationship between the descriptor used to construct the hull and the stabilisation term to the energy under the related constraint is linear i.e:

$$E' = E + Q(x)D \quad (2.17)$$

where  $E$  is the original energy,  $E'$  is the altered energy,  $D$  is the descriptor, and  $Q(x)$  is a term dependent on the value  $x$  of a constraint coupled to the descriptor. In these circumstances, the change to the stability will scale linearly with the descriptor  $D$  on which the hull is constructed. It is worth noting here that despite its naming here as the ‘stabilisation term’, the contribution  $Q(x)D$  can be

stabilising or destabilising depending on the signs of  $Q(x)$  and  $D$ .

Then, it can be said that the structures corresponding to hull points are each stabilisable by controlling the constraint. A full proof is not provided here, but the fact that the hull points are **edge structures** leads to the phenomenon that any hull point can become the global minimum in  $E'$  at some value of  $Q(x)$ . One way to picture this is to imagine the set of datapoints as movable. Applying a stabilisation term that scales linearly with the descriptor(s) would induce energy changes on a ‘gradient’ - i.e structures with high descriptor values will be (de)stabilised more than those with low descriptor values or vice versa.

With regard only to the points position along the energy axis, the application of the constraint would have an impact akin to tilting the set of points. It can then be imagined that for hull points, some degree of tilt could bring each point to be the global minimum along the energy axis.

However, there is no stabilisation term  $Q(x)D$  at which non-hull points can become the most stable. A proof for the case using a single structural descriptor is straightforward. First consider that all non-hull points in such a case must lie energetically above some line joining two hull points. The three points therefore form a triangle as illustrated in Figure 2.11, where  $E_i$  represents the energy of the corresponding point, and  $D_i$  represents its value of descriptor  $D$ .

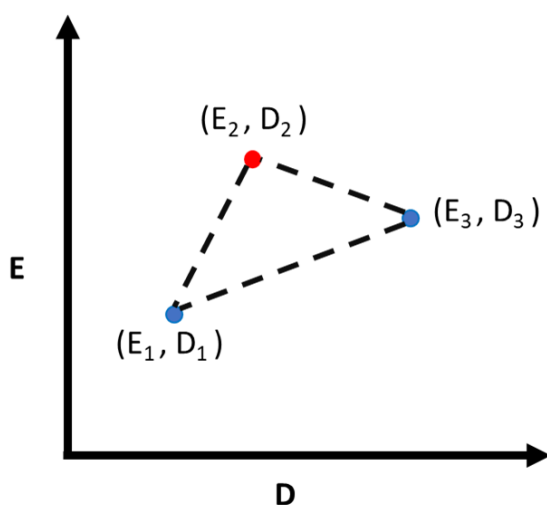


Figure 2.11: Example of a ‘triangle’ formed of non-hull and hull points

We label the points such that:

$$D_1 < D_2 < D_3 \quad (2.18)$$

A necessary but not sufficient condition for a point to become the global minimum would be:

$$E'_2 < E'_1 \textbf{ and } E'_2 < E'_3 \quad (2.19)$$

Therefore if it can be proven that if, for a given point, there exists such a triangle which cannot satisfy this condition, then that point cannot be stabilised.

Let us first define  $E'_i$  for the points of such a triangle:

$$\begin{aligned} E'_1 &= E_1 + QD_1 \\ E'_2 &= E_2 + QD_2 \\ E'_3 &= E_3 + QD_3 \end{aligned} \quad (2.20)$$

Therefore:

$$\begin{aligned} E'_2 < E'_1 &\Rightarrow Q < \frac{E_2 - E_1}{D_1 - D_2} \\ E'_2 < E'_3 &\Rightarrow Q > \frac{E_2 - E_3}{D_3 - D_2} \end{aligned} \quad (2.21)$$

and so :

$$\frac{E_2 - E_3}{D_3 - D_2} < Q < \frac{E_2 - E_1}{D_1 - D_2} \quad (2.22)$$

There exist two cases:

1.  $E_2 > E_1$  **and**  $E_2 > E_3$
2.  $E_3 > E_2 > E_1$  **or**  $E_1 > E_2 > E_3$

Case 1 is the simplest to prove. Here:

$$\begin{aligned} E_2 > E_1 \textbf{ and } D_2 > D_1 &\Rightarrow \frac{E_2 - E_1}{D_1 - D_2} < 0 \\ E_2 > E_3 \textbf{ and } D_3 > D_2 &\Rightarrow \frac{E_2 - E_3}{D_3 - D_2} > 0 \end{aligned} \quad (2.23)$$

This contradicts Equation 2.22. Therefore it is not possible to stabilise the structure in this case.

Case 2 is more complex to prove. First we must introduce an additional constraint, defining that energy  $E_2$  must lie above the line joining  $E_1$  and  $E_3$

$$E_2 > \frac{E_3 - E_1}{D_3 - D_1}(D_2 - D_3) + E_3 \quad (2.24)$$

Therefore:

$$\begin{aligned}
 \frac{E_2 - E_1}{D_1 - D_2} &= \frac{\left(\frac{E_3 - E_1}{D_3 - D_1}(D_2 - D_3) + E_3 + a\right) - E_1}{D_1 - D_2} \\
 &= \frac{E_3 - E_1}{D_3 - D_1} \frac{D_2 - D_3}{D_1 - D_2} + \frac{E_3 - E_1}{D_3 - D_1} \frac{D_3 - D_1}{D_1 - D_2} + \frac{a}{D_1 - D_2} \\
 &= -\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_1 - D_2}
 \end{aligned} \tag{2.25}$$

where  $a$  is strictly positive.

Further:

$$\begin{aligned}
 \frac{E_2 - E_3}{D_3 - D_2} &= \frac{\left(\frac{E_3 - E_1}{D_3 - D_1}(D_2 - D_3) + E_3 + a\right) - E_3}{D_3 - D_2} \\
 &= -\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_3 - D_2}
 \end{aligned} \tag{2.26}$$

Therefore, according to Equation 2.22:

$$\frac{E_2 - E_3}{D_3 - D_2} = -\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_3 - D_2} < -\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_1 - D_2} = \frac{E_2 - E_1}{D_1 - D_2} \tag{2.27}$$

However:

$$\begin{aligned}
 D_2 > D_1 &\rightarrow \frac{a}{D_1 - D_2} < 0 \\
 D_2 < D_3 &\rightarrow \frac{a}{D_3 - D_2} > 0
 \end{aligned} \tag{2.28}$$

and so:

$$-\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_3 - D_2} > -\frac{E_3 - E_1}{D_3 - D_1} + \frac{a}{D_1 - D_2} \tag{2.29}$$

which is a contradiction - thereby proving stabilisation of the point is not possible in this case.

If it can be assumed that a linearly-scaled stabilisation term cannot alter the points which lie on the hull, then the proof can be extended to higher dimensions. This consistency of hull points allows us to treat the contributions to the energy change entirely independently. Then, the above proof must hold in an  $n$ -dimensional case, as the prior proof can simply be used along each two-dimensional slice (of energy vs one descriptor) of the landscape at a time.

In principle, this theory states that only crystal structures corresponding to hull points can be stabilised, and that the relationship between the constraint and the stabilisation term must be linear. However, in reality a wider pool of crystal structures may be stabilisable. In much the same way as proposed stabilisable structures are traditionally drawn from an energy window rather than being limited to the global minimum structure alone, metastable structures - i.e those close to the hull could also be found. Further, possible errors in the calculated energies and/or deviation from perfect linear behaviour may allow some near-hull structures to be stabilised.

## 2.5 Supervised Machine Learning Methods

### 2.5.1 Overview

In addition to the use of unsupervised machine learning to derive ML descriptors for GCH construction, work in this thesis has also employed **supervised** machine learning methods to assess the utility of different kernel constructions. The two key methods used were Support Vector Classification (SVC) and Gaussian Process Regression (GPR).

### 2.5.2 Support Vector Classification

Support Vector Machines are a supervised machine learning method. They can be used for both classification and regression, but the use-case in this thesis is for binary classification.

Support vector classification for binary classification predicts the class of a sample in a dataset based upon its position relative to a ‘separating hyperplane’. The simplest SVC formulation is hard-margin linear SVC. For a dataset with  $d$  features - i.e that would be plotted as a landscape of points in  $d$  dimensions - the separating hyperplane is a  $d - 1$  dimensional hyperplane that perfectly linearly ‘divides’ the landscape so as to separate samples of different classes. An example of this, for a dataset with two features, is shown in Figure 2.12

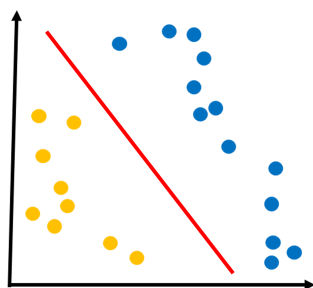


Figure 2.12: Example of a separating hyperplane in a simple SVC model

In the purely hypothetical and unconstrained case, there can be infinite separating hyperplanes (Figure 2.13) - but the algorithm must select a single definition by which to classify structures.

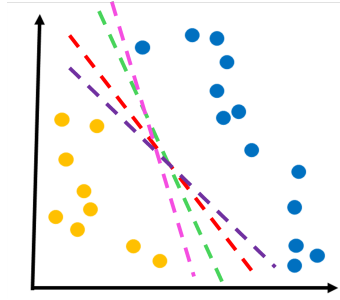


Figure 2.13: Examples of different possible separating hyperplanes to separate classes in a simple SVC model

This definition of the separating hyperplane is determined by the training of the model. The objective function to be maximised during training relates to the defined ‘margins’ - the distance of the nearest samples in the space from the separating hyperplane (Figure 2.14). The reasoning for this relies upon the concept that the distance of a point from the separating hyperplane relates to the certainty with which it can be classified.

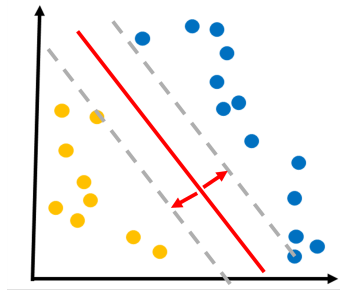


Figure 2.14: Example of a separating hyperplane in a simple SVC model and the corresponding margins - the distance from the hyperplane to the nearest data point

For cases such as Figure 2.12 the value of this method can be seen intuitively. However, in more realistic cases such as Figure 2.15 a hyperplane designed to ensure perfect linear separation of classes will not exist. To manage this, support vector machines incorporate an allowance for miss-classification by introduction of ‘Soft-Margins’. These allow for the definition of a hyperplane such that some samples may be lie on the wrong side of the hyperplane for their respective class, or lie within the margin region. A measure of this miss-classification is added to the objective function in order to penalise it and therefore minimise its extent.



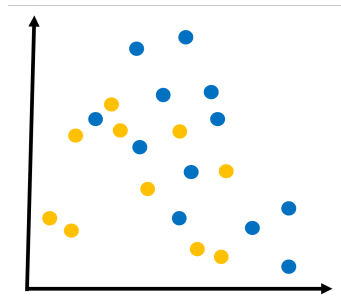


Figure 2.15: Example of a set of points of two classes that is not perfectly linearly separable

For cases in which a linear hyperplane - even with allowed miss-classification - would be ineffective, the ‘kernel trick’ can also be employed. This allows for classifying samples based on how they could be linearly separated if the landscape were transformed into a higher dimensional kernel space. [117]

### 2.5.3 Gaussian Process Regression

Gaussian process regression is a supervised machine learning method that uses Bayesian Linear Regression. In short, these models attempt to find mathematical functions that fit the training data and can then be used to predict target values for unseen data. However, crucially, instead of determining a single function, the models include contributions from many functions that could fit the data.

To make predictions, GPR models take the mean prediction for a given sample over a set of predictive functions. These functions are determined such that they are reasonable in two regards:

1. They account for knowledge of the similarity of samples and ‘correlation’ of features
2. They fit the training data

The approach utilises a **prior** - which is an (infinite) set of functions that have been determined to be reasonable by incorporating prior knowledge. In the case of GPR, the prior functions are a theoretically infinite number of functions sampled from a Multi-Variate Normal Distribution (MVN) of functions. This distribution incorporates prior knowledge about the correlation between variables based on a provided kernel. Incorporation of this knowledge controls the shape of functions in the prior, such that points that are similar according to the kernel have similar outputs from the functions. This smooths the functions in a reasonable and meaningful manner.

Initially, the prior functions are just a set of hypothetical functions and are not directly influenced by observed data points. Constraining this set to include only those functions that fit the

observed/training data leads to derivation of the **posterior** - the set of prior functions that also fits the data. The final predictive function is taken to be the mean function over the posterior [117, 118].

## 2.6 Additional Tools Used in this Work

### 2.6.1 Overview

In addition to the most crucial methods and aspects of underlying theory that have been discussed - there are many additional tools and resources used to achieve the work in this thesis. To end discussion of the theory and methods, a few of the most important are briefly outlined here.

### 2.6.2 ASE

Atomic Simulation Environment (ASE) [119] is a Python package providing important functionalities for materials modelling at the atomistic scale. This includes constructions to define molecular or crystal structures - such as 'Atoms' objects. Atoms objects have been used extensively in this work to provide structural data inputs for kernel calculations. The primary software used in this work for performing SOAP descriptor and kernel calculations, librascal, interfaces with ASE. Additionally, the package provides tools for energy calculations and geometry optimisations, which have been used for MACE-level crystal structure reoptimisations in this project.

### 2.6.3 Librascal

Librascal [113] is a python library developed by the COSMO group[120] to provide functionality for calculating structural representations and descriptors - primarily for machine learning purposes. Librascal has been used extensively in this work, for calculation of SOAP descriptors and within kernel calculations.

### 2.6.4 PYMATGEN

Pymatgen [121] is an open source python library for analysis of the structures of materials, version 2022.1.7 has been used in this work to identify symmetry operators for molecule objects.

### 2.6.5 Molecular Graphs

Molecular graphs are mathematical graph constructions that can be used to describe the connectivity - but not usually the 3D geometry of a molecule. In these graphs, nodes are used to represent atoms, and edges of the graph denote bonds connecting the atoms [122]. Molecular graphs can be created and analysed using packages such as networkx [123], and have been used in this work primarily for identification of isomorphisms of the graphs.

### 2.6.6 CSD

The Cambridge Structural Database is a databank, managed by the Cambridge Crystallographic Data Centre (CCDC). It contains over 1.35 million experimentally reported molecular crystal structures. It is widely used in the molecular CSP community as a reputable source of reference data and has been used in this work as a source of experimental data used to assess the quality of predictions. Deposited crystal structures can be identified using assigned alphanumeric 'refcodes' - which are used in this thesis to declare the structure data used [124].

### 2.6.7 Mercury

The Mercury software and Graphical User Interface (GUI) is a visualisation software provided by the CCDC. It provides functionality for visualisation of crystal structures - which has been used in production of this thesis. It also provides analysis tools - particularly designed for molecular crystals [125].

### 2.6.8 CSD API

The Cambridge Crystallographic Data Centre provides a Python Application Programming Interface (API) to provide users with a library of code to implement much of the same functionality as the Mercury Software. The library contains many useful classes such as those designed to handle crystals and molecules as well as classes to handle geometric and molecular descriptors. This allows the user to write scripts to automate tasks and run them from the command line or on a worker node -with greater efficiency than manually performing tasks via the Mercury GUI. The majority of work in this thesis implements version 3.0.0 of the Python API.[105]

## Chapter 3

# Constructing a Kernel Suited to Molecular Crystals

### 3.1 Overview

This chapter introduces work performed to propose a new global SOAP kernel. This is designed to provide a more theoretically reasonable definition of the similarity of a pair of molecular crystal structures than provided by conventional global SOAP kernels. The new global kernel is proposed as a more appropriate ‘molecular crystal average kernel’ - being, conceptually, a constrained version of the simple average global SOAP kernel.

This work developed a wrapper around an existing open-source kernel calculation code to alter the final global kernel constructed. The developed code acted to restrict the atom-atom comparisons made, i.e to restrict the local kernels that contributed to the global kernel. These constraints were selected to ensure that the resulting global kernel meaningfully described the similarity of molecular crystals.

This chapter acts to explain the new kernel construction, the assessment of its utility being investigated in later chapters. The discussion here introduces the limitations of conventional global SOAP kernels and explains the conceptual alterations made to the kernel construction, the intricacies of implementing those changes, and the limitations of the approach.

### 3.2 The Problem with Conventional Kernels

As discussed in Section 2.4.3, conventional global SOAP similarity kernels, including those used for the original implementation of the GCH [1], define similarity between two crystal structures as some function of **all pairwise atom-atom similarities** between atoms in the unit cells of the two structures - i.e all pairwise similarities between atoms in the unit cell of structure A and atoms in the unit cell of structure B. Some of these constructions are simple - such as the average SOAP kernel, in which the function used is merely the arithmetic mean. Others are more complex, such as the best-match or ReMatch kernels, in which the functions incorporate taking a maximal similarity value across different sets of atom-atom comparisons [1, 112]. Crucially, though, **all** atom-atom comparisons are processed by the function used to derive the final global kernel.

The issue with this is most clearly demonstrated by considering the **average SOAP kernel** - which treats the global similarity between two crystal structures A and B as the average of the local similarities between all pairs of atoms in the unit cells of A and B. Recall that:

$$K(A, B) = \frac{\sum_{s=0}^{S-1} \sum_{t=0}^{T-1} k(A_s, B_t)}{ST} \quad (3.1)$$

normalising according to:

$$K(A, B) = \frac{K(A, B)_{norm}}{\sqrt{K(A, A)K(B, B)}} \quad (3.2)$$

where  $s$  and  $t$  denote atom indices within the respective unit cells,  $S$  is the number of atoms in unit cell of A,  $T$  is the number of atoms in the unit cell of B, and  $k(A_s, B_t)$  is the local similarity between atom  $s$  of the unit cell of A and atom  $t$  of unit cell of B and  $K(A, B)$  [112].

Such definitions are not always chemically reasonable, especially for molecular crystal structures. Though they include reasonable atom-atom comparisons - such as that indicated by the **green** arrow in Figure 3.1 they also include comparisons that are not meaningful for defining the similarity, such as that indicated by the **red** arrow in Figure 3.1.

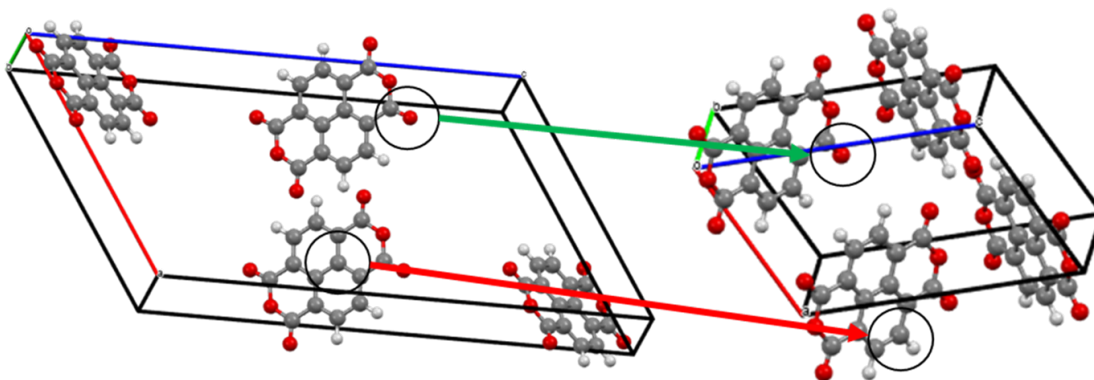


Figure 3.1: Example of two molecular crystal structures and hypothetical atom-atom comparisons between them. The green arrow signifies a reasonable comparison, while the red gives an example of an unreasonable comparison.

If two atoms in crystal structures A and B respectively correspond to different atoms of the underlying molecule - i.e they would not have identical intramolecular environments in any conformation of the isolated molecule - then the local kernel between those two atoms  $k(A_s, B_t)$  provides an unreasonable contribution to the global similarity of the crystals. This is because the similarity value is not solely defined by the differences in the intermolecular contributions to the atomic environment. It is also impacted by the differences simply arising from the two environment centres corresponding to different atoms of the molecule. Even when comparing identical structures, such atomic environments would be expected to be different. Further, if two structures happened to have a high similarity for such an atom-atom comparison, this would not correspond to actual similarity of the structures. As such, these comparisons should not be considered as meaningful contributions to the assessment of the global similarity.

The simplest average SOAP kernel will even include contributions from comparison of atomic environments in which the environments centres are of different species. For example, comparing the environment of a carbon in one structure to the environment of an oxygen in another structure. This is the implementation used in librascal [113]. This was the underlying kernel calculation package used in this thesis. Given its correspondence to the ‘true’ simple average kernel, and it representing an accurate view of the average kernel as would be calculated by the base package without adaptation, this is the definition of the average kernel used throughout this thesis. However, it is acknowledged that it is possible to restrict average kernel construction such that only local kernels between atomic environments with centres of the same species are included. A functionality [110] in a later codebase, also developed by the COSMO group [111], now incorporates

this limitation, though unreasonable comparisons between atoms of the same species remain a problem.

Other, more complex, definitions of the global SOAP kernel still fail to fully account for this issue [112]. The best-match and ReMatch kernels restrain the final global kernel to focus upon the combination of atom-atom comparisons between two crystal structures that results in the highest similarity value. However, all mathematically possible combinations - including those that are theoretically unreasonable - are considered. For example, both the reasonable (a) and unreasonable (b) sets of atom-atom comparisons shown in Figure 3.2, among others, would be trialled as possible means to define the similarity of the two crystals.

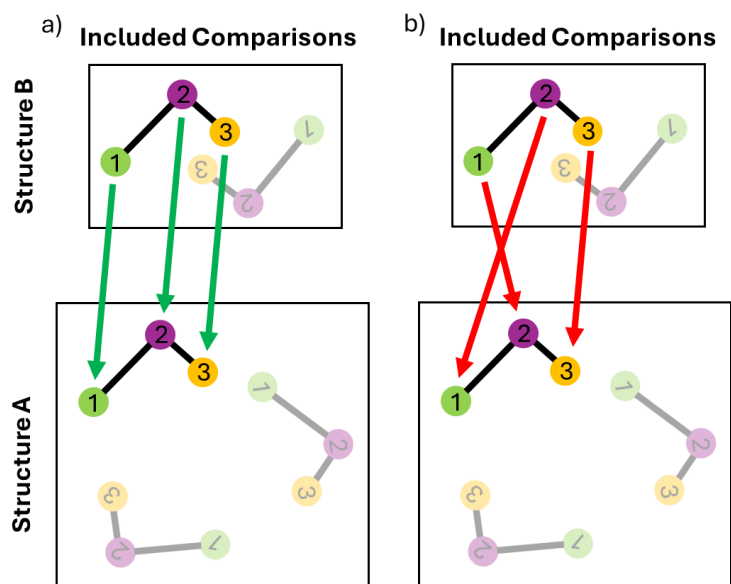


Figure 3.2: Examples of combinations of atom-atom comparisons that would be trialled in calculating the global best-match or ReMatch kernels. One example a) is a theoretically reasonable combination while the other, b), is not. Note in practice, these comparisons would likely be carried out across the entire unit cell, and not just asymmetric units

Therefore, given simply a final global kernel between two crystal structures, it is not possible to know if it corresponds to a meaningful definition of the similarity. That is, it is not possible to know if the set of comparisons included would be expected by chemical/physical reasoning to provide a good description of the similarity of the crystal structures. It is not intrinsically guaranteed that the most theoretically reasonable combination of atom-atom comparisons will lead to the highest similarity value - as the intermolecular contributions and contributions due to confor-



mational differences also exhibit an influence on the atomic environment. Though it is not known how frequent unreasonable cases will be - work in this thesis sought to directly exclude the possibility. The best-match and rematch kernels in this respect could also be said to be computationally wasteful, as they dedicate resources to considering combinations of atom-atom comparisons that simple chemical reasoning could have excluded.

### 3.3 Adaptation Concept

#### 3.3.1 Overview

In light of these issues, work aimed to develop a new global SOAP kernel by adapting the simplest conventional kernel - the average SOAP kernel. To do this, constraints were applied to the global kernel calculation - restricting the included local kernels. This was done such that all included local kernels corresponded to comparisons where the environment centres could be said to represent the same atom of the underlying molecule. The atom pairs corresponding to these reasonable comparisons can be called ‘analogous atoms’.

#### 3.3.2 Crystals of Asymmetric Molecules

For the simplest cases (crystal structures of asymmetrical molecules) the definition of reasonable comparisons is conceptually simple. In these cases, each atom of the molecule has a unique intramolecular environment, regardless of the intramolecular conformation. Therefore each ‘atom of the underlying molecule’ is uniquely defined. For example, in Figure 3.3, there can be no question as to which atoms in the second crystal represent the same atom of the underlying molecule as the highlighted atom in the first crystal. Such an atom in the second crystal is also highlighted.

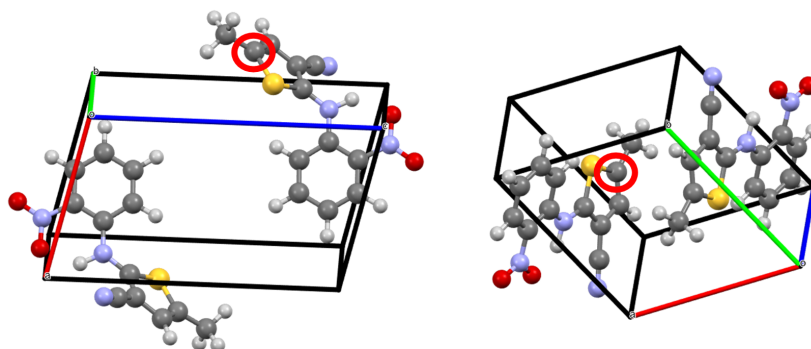


Figure 3.3: Example of two predicted crystal structures of ROY [55]. The circled atoms represent examples of analogous atoms between the two structures

Given this, each atom can then be uniquely identified by its molecular atom index - its place in the declared order of atoms within the molecule. This indexing arises from labelling or ordering within the structure files. (See Figure 3.4)

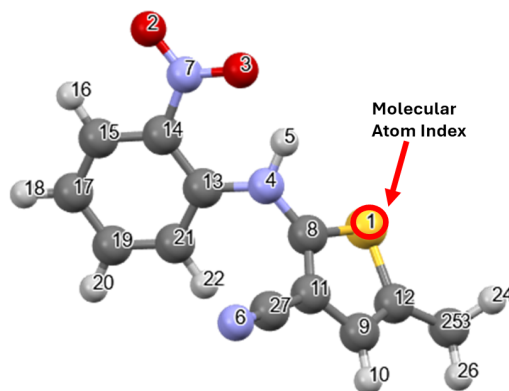


Figure 3.4: Example of a ROY molecule - numbered according to its indexing as derived from a structure file formed during predictions [55]. The numbers represent the determined 'molecular atom indices'.

Therefore, the reasonable atom-atom comparisons to include when comparing two crystal structures of an asymmetric molecule are those between analogous atoms that - assuming consistent indexing of the structures - will share the same molecular atom index. These reasonable comparisons can then be averaged to give a global view of the similarity between the crystal structures.

Consider the simple case of a structure set containing only  $Z'=1$  crystal structures. Recall that the  $Z'$  value denotes the number of formula units in the asymmetric unit. In all cases in this thesis, this is given simply by the number of **molecules** in the asymmetric unit. For  $Z'=1$  crystal structures of asymmetric molecules the kernel  $K(A,B)$  between two structures  $A$  and  $B$  is given by:

$$K(A,B) = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} k(A_x, B_y) \times f(x,y)}{uvN} \quad (3.3)$$

$$f(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

Where  $x$  and  $y$  index the atom within the underlying molecule,  $N$  is the number of atoms within the molecule, and  $u$  and  $v$  denote the number of copies of the asymmetric unit within crystal structures  $A$  and  $B$  respectively. As all copies of the asymmetric unit are in the same environment, the comparisons need only be made between one copy of each asymmetric unit, as further comparisons would return the same result. Then,  $u$  and  $v$  can be treated as being equal to one. In this simple instance, the kernel is normalised by construction - as the kernel between identical structures,  $K(A,A)$ , will be equal to one.

The kernel construction applied to  $Z'=1$  crystal structures of asymmetric molecules is demonstrated in Figure 3.5.

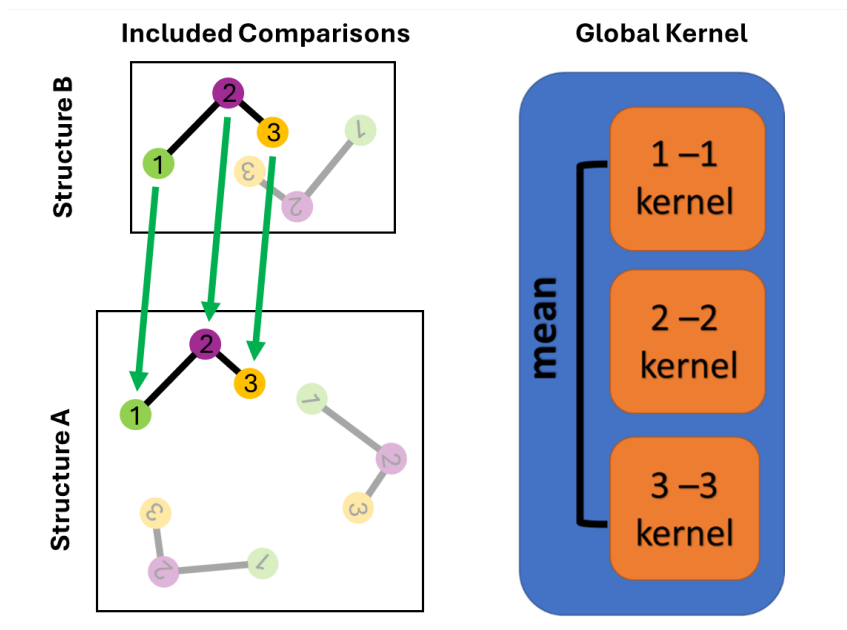


Figure 3.5: Example construction of the adapted kernel for  $Z'=1$  crystals of asymmetrical molecules

In a crystal structure of  $Z'>1$ , each molecule of the asymmetric unit may be in a different environment. Thus, comparisons should not arbitrarily consider only one particular molecule of each asymmetric unit. Instead comparisons between all molecules of the respective asymmetric units should be considered. There are two clear approaches to combining these comparisons to form a final kernel between crystal structures. These are to use a best-match scheme or an averaging scheme. These schemes are indicated in Figure 3.6.

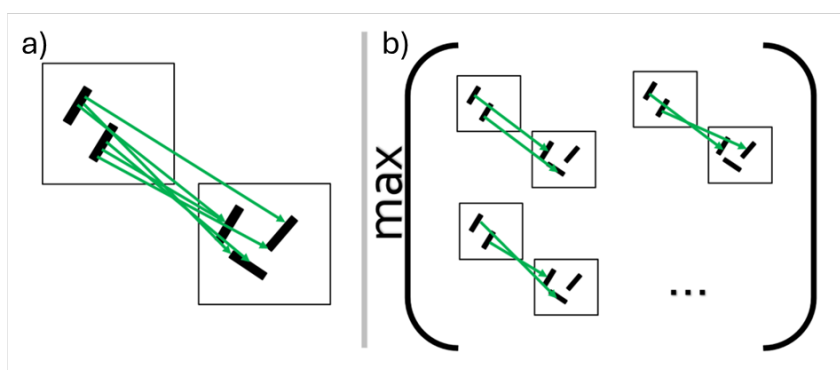


Figure 3.6: The molecule-molecule comparisons included in the final kernel construction when comparing crystal structures with  $Z'>1$  using the a) averaging and b) best-match schemes.

A best match scheme would separately consider all possible sets of molecule-molecule compar-

isons for the asymmetric units of a given structure pair and take the maximum of these similarity values. This has the advantage that in each case conflicting comparisons are not included. That is, with careful implementation, for each molecule in one crystal only comparisons between that molecule and a **maximum** of one molecule in the other crystal will be included. The self-similarities will also be defined to be one - making the scheme appear more reasonable. However, it is complex and costly. Further, any given case may neglect structural information. For example in comparing  $Z'=1$  structures to  $Z'=2$  structures, the resulting similarities will be based upon comparisons that exclude one molecule of the  $Z'=2$  asymmetric unit - so key structural information is excluded from the assessment of similarity.

Using an averaging scheme, every molecule of the asymmetric unit in structure A is compared to every molecule of the asymmetric unit in structure B. The resulting molecule-molecule comparisons are then averaged to derive the final kernel. This approach is simple, generalisable, and utilises all unique structural information from the structures - i.e no molecule of the asymmetric units is neglected in comparisons. However, it may not represent the best definition of similarity, for instance under this definition the self similarity is not equal to one and 'conflicting' molecule-molecule comparisons are included simultaneously.

This work uses an averaging scheme due its conceptual simplicity and comparative low cost. The definition of the kernel then becomes:

$$K(A,B) = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} k(A_x, B_y) \times f(x,y)}{uUvVN} \quad (3.4)$$

Where  $U$  and  $V$  denote the  $Z'$  of crystals  $A$  and  $B$ . Here, the kernel is no longer normalised by construction - and must be normalised as in Equation 3.2

### 3.3.3 Accounting for Symmetry

Where the underlying molecule in a crystal is symmetrical, the process is more complex. If a molecule 'has symmetry' there will exist symmetry operations - e.g mirror planes - that map each atom in the molecule to an atom of the same element within that molecule. This means that the molecule can be 'transformed' by the operation to return an identical molecule. The set of symmetry operations that apply to any given molecule will form a mathematical group [7] - called the molecular point group [126].

Due to molecular point group symmetry, there can exist multiple valid sets of analogous atoms and corresponding reasonable comparisons. This is because, in a given conformation of the isolated molecule, there may be atoms - those that can be mapped to one another under a symmetry operation - that have identical intramolecular environments. In these cases, distinguishing between two such atoms using the molecular atom index, and restricting the atom-atom comparisons accordingly, introduces an arbitrary factor based upon the way in which the atoms have been labelled. That is, the ‘unique’ identity of an atom within a molecule is only meaningful up to a transformation by a molecular point group operator. Distinction between atoms that are symmetry equivalents under any given operator should not be used to restrict the atom-atom comparisons included within a similarity kernel. To demonstrate this issue, consider two crystal structures  $A$  and  $B$  of a symmetric molecule. Then, consider a third structure - an adaptation of structure  $B$  ( $B'$ ) in which a molecule has been acted upon by a molecular point group operator  $q$  - mapping every atom in the molecule to an atom of the same element within that molecule. Structures  $B$  and  $B'$  are thus identical. However, if the definition of the kernel given in Equation 3.4 were implemented, kernels  $K(A, B)$  and  $K(A, B')$  may differ because the local kernels  $k(A_x, B_x)$  or  $k(A_x, B'_x)$  may refer to a comparison between different pairs of atomic environments (See Figure 3.7).

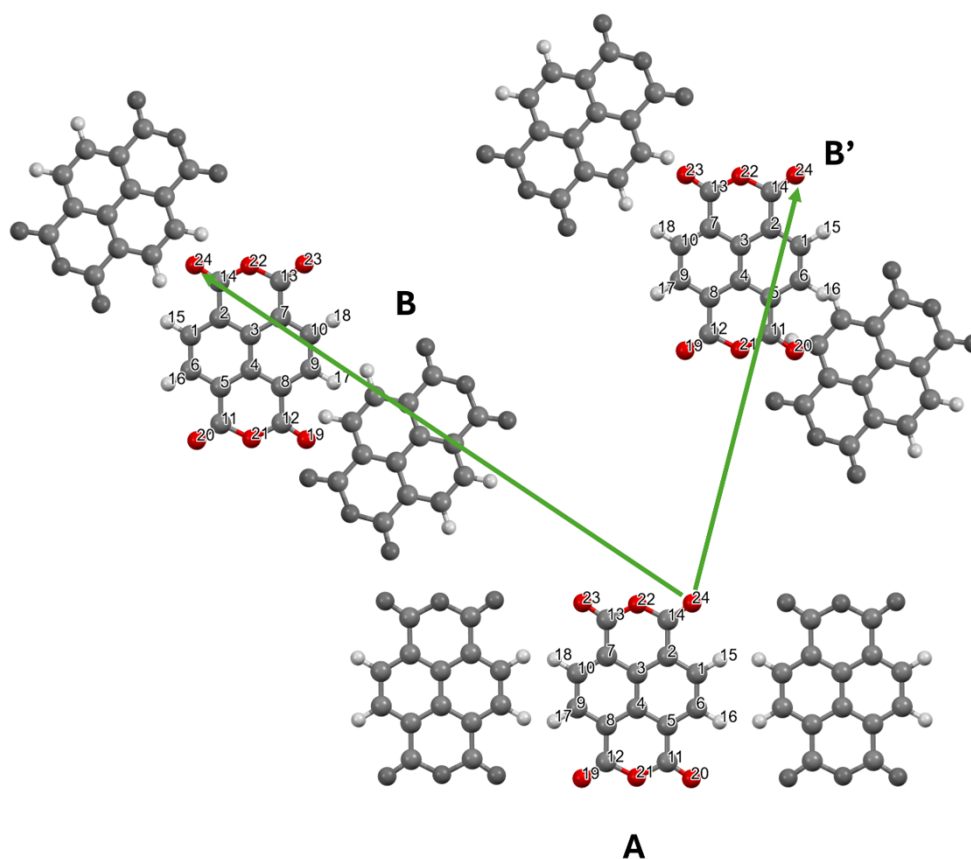


Figure 3.7: Hypothetical crystal structures A, B, and B'. B and B' are identical, however a molecular point group operator has transformed the asymmetric unit of B - forming B'. The arrows indicate a specific atom-atom comparison (index 24- index 24) that would be included in  $K(A,B)$  and  $K(A,B')$  demonstrating the difference in the local environment comparisons that would be included.

This ambiguity in the definition of the similarity between the two crystal structures is problematic. Neither  $K(A,B)$  or  $K(A,B')$  provide an inherently more correct option for defining the similarity. A valid kernel for comparing crystal structures of symmetric molecules should consider  $K(A,B')$  for all possible  $B'$  and define a final global kernel that is deterministic.

For  $Z'=1$  crystal structures, each kernel  $K(A,B')$  can be separately calculated by considering the action of each molecular point group operator  $q$  in turn. Then kernels  $K_q(A,B)$  can be constructed that include atom-atom comparisons for pairs where the given operator  $q$  maps one of the corresponding atoms of the underlying molecule to the other. These kernels  $K_q(A,B)$  will be referred to

as 'possible mapping kernels':

$$K_q(A, B) = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} k(A_x, B_y) \times g(x, y)}{stN} \quad (3.5)$$

$$g(x, y) = \begin{cases} 1 & \text{if } q(x) = y \\ 0 & \text{otherwise} \end{cases}$$

Here, it is sufficient to consider only

$$[K(A, B'), \forall B'] \quad (3.6)$$

That is, it is sufficient to define the possible  $K(A, B)_q$  as in Equation 3.5 and not consider:

$$K(A, B)_q = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} k(A_y, B_x) \times g(x, y)}{stN} \quad (3.7)$$

$$g(x, y) = \begin{cases} 1 & \text{if } q(x) = y \\ 0 & \text{otherwise} \end{cases}$$

This is because for any molecular point group operator there must be an inverse operator in the group. Applying Equation 3.7 for a given molecular point group operator  $q$  is equivalent to applying Equation 3.5 for a given operator  $q'$  - where  $q'$  is the inverse of  $q$ . Therefore, the construction in Equation 3.5 is sufficient. This is demonstrated for a hypothetical 3 atom 'molecule' in Figure 3.8, the result being that it is sufficient either to consider all transformations of the left-hand structure or all transformations of the right-hand structure.



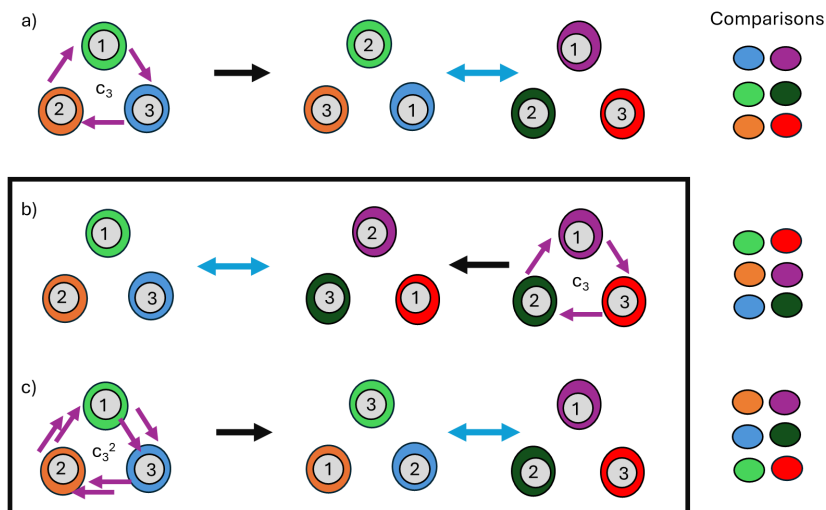


Figure 3.8: Hypothetical 3-atom molecule within crystal structures. Coloured ellipses behind each atom represent the surrounding local environment. The molecule in one of the structures has undergone transformation by a molecular point group operator prior to crystal structure comparison. In the case of a) and b) this is the operation  $C_3$ , and in c) it is the inverse operation  $C_3^2$ . The local environment comparisons that would be induced by comparing atoms of matching indices is shown in each case. This demonstrates that whilst the cases of a) and b) differ - c) is equivalent to b). Therefore both cases a) and b) can be covered simply by considering all transformations of the left-hand structure, including the inverse operations.

To construct a final kernel  $K(A, B)$ , the contributions  $K_q(A, B)$  are averaged. That is, the kernel between any two crystal structures is taken to be the mean of the individual similarity values for that structure pair across the possible mapping kernels:

$$K(A, B) = \frac{\sum_{q=1}^Q K(A, B)_q}{Q} \quad (3.8)$$

Where  $Q$  is the total number of point group operators. In this way, the adapted kernel can be thought of as an average of possibilities kernel. This kernel construction is demonstrated in Figure 3.9. Again, this kernel is no-longer normalised by construction - and must be appropriately normalised.

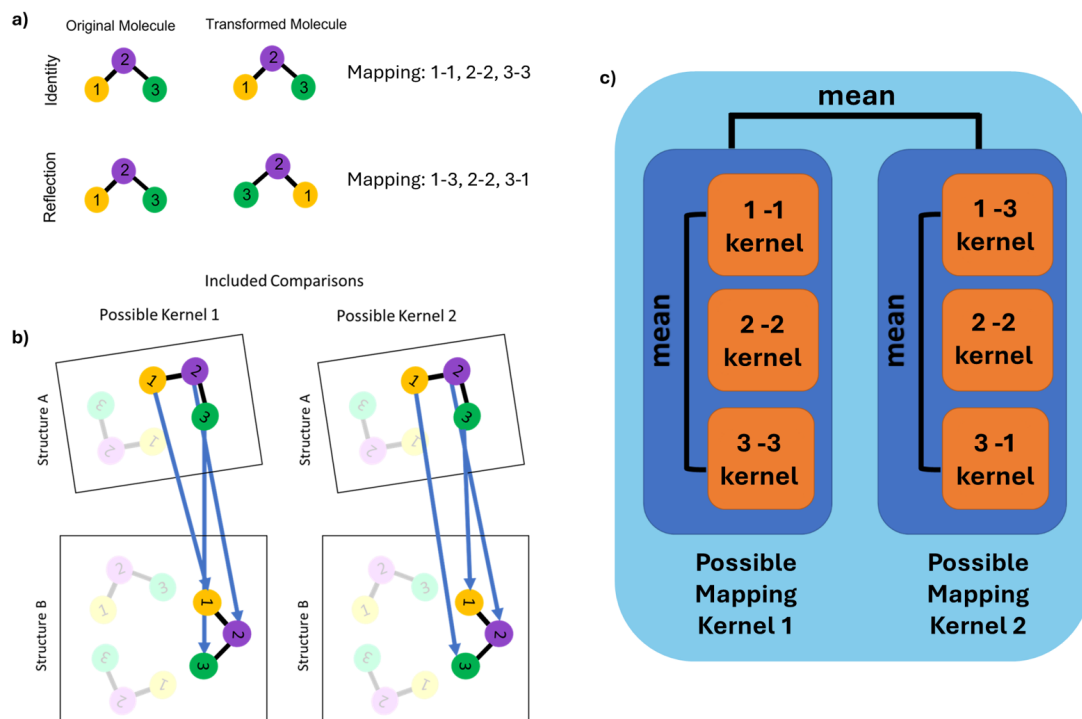


Figure 3.9: Example adapted kernel construction in the case of  $Z'=1$  crystal structures of a symmetrical molecule. a) shows the symmetry mappings relevant to the molecule, b) shows the atom-atom comparisons included in the corresponding possible mapping kernels, and c) indicates how this information is combined to form the final kernel.

In this work, the ‘symmetry of a molecule’ in a given conformation was determined by the point group of the molecule in that conformation, so as to incorporate geometric considerations when deriving the respective set of atom-atom mappings. An alternative would be to determine atom-atom mappings using merely the connectivity information of the molecule but this could in some cases classify as analogous pairs of atoms that are not actually interchangeable under symmetry in any of the structures present in the set.

When the structure set contains structures with  $Z' > 1$ , this introduces further complexity. This work attempted to construct a reasonable kernel  $K(A,B)$  for such cases by considering not just the action one point group operator at a time, but one point group operator per molecule of the asymmetric unit of  $B$  at a time. That is that each possible mapping kernel contributing to the final adapted kernel would not be a single  $K_q(A,B)$  but would itself, under the  $Z'$  averaging scheme, be the average of  $K_M(A,B_m)$ . Here,  $m$  indexes the molecules of the asymmetric unit of  $B$ , and  $M$  indexes the symmetry operation that determines comparisons between the molecules of the asymmetric unit of  $A$  and molecule  $m$  of the asymmetric unit of  $B$ . The number of possible mapping

kernels here is then equal to the number of combinations of the molecular point group operators across the molecules of the asymmetric unit of  $B$ . In practice, kernel calculations were performed such that crystal  $B$  of any  $A, B$  pair was the one with the largest asymmetric unit. Therefore, the number of possible mapping kernels - averaged to construct the final adapted kernel - was equal to  $N^{Z'_{\max}}$ , where  $Z'_{\max}$  is the highest  $Z'$  of the structure pair and  $N$  is the number of molecular point group operators. An example is given in Figure 3.10 for the case of  $Z'=2$  crystals of a molecule with three point group operators - including the identity.

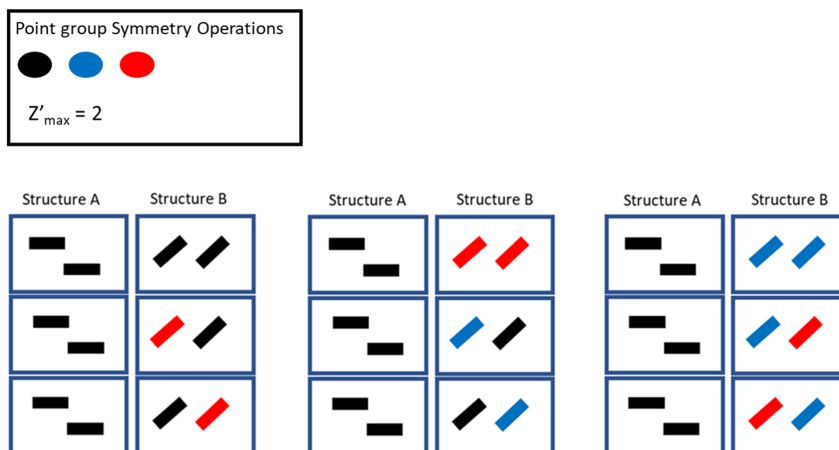


Figure 3.10: An example of the possible combinations of molecular point group operators for which kernels would need to be generated in the case of three operators and a largest  $Z'$  of two - showing that the number of kernels to create to cover all possibilities grows quickly.

However, this construction later proved problematic (see Section 3.3.4). These issues could not be resolved on the timescale of the work. To generate useful results, only  $Z'=1$  crystal structures of symmetric molecules and crystal structures of asymmetric molecules with any  $Z'$  were considered going forward. Therefore, this aspect of the kernel construction is not discussed further here.

### 3.3.4 Why Average the Possibilities?

Initially, work in this thesis planned not to form an adapted kernel by averaging across  $K_q(A, B)$ , but by taking the **maximum** across that set. This would still give a consistent global kernel between two structures, but would act as perhaps a more theoretically reasonable kernel - not including contribution from any conflicting comparisons and setting the self-similarities to one by construction.

Unfortunately, such a construction is not useful in practice. The kernel matrix of the ‘maximum kernel’ across a structure set was not positive semi-definite (PSD). Being positive semi-definite is an important property for the utility of a kernel, particularly for use in ML applications such as

Gaussian process regression [127].

Following largely trial-and-error investigation as to the origins of and solutions to this problem, it was found that taking the average across the set of  $K_q(A, B)$  resulted in a PSD kernel construction. This is a mathematical concern, and not a chemical one as, in order to ensure the PSD property, the group of symmetry operators considered need not be truly representative of the point group of the relevant molecule, but **must** be a true mathematical group.

However, where a structure set contained crystal structures of symmetric molecules with  $Z' > 1$ , this ‘average of possibilities’ construction still lacked the PSD property - and so such structure sets were not explored further in this work.

The need for the set of symmetry operators used to form a mathematical group introduces a further consideration in some cases as a suitable group of operators must be selected. For crystal structure sets featuring only a single conformation of the underlying molecule, the choice here is self-evident. The conformation will have a fixed point group defining its symmetry. This point group fulfils the mathematical criterion and is of course relevant to the system at hand. By the closed property of groups, the point group contains not only all possible symmetry operators for the molecule - but also all combinations of those symmetry operators. Inclusion of these combinations is vital to consider all valid  $K_q(A, B)$ .

Where the underlying molecule is flexible, however, a prediction set may contain multiple in-crystal molecular conformations - each of which could be defined by a different point group and corresponding set of symmetry operations. To encompass all valid possibilities, all symmetry operators applying to a present in-crystal conformation, as well as all combinations of these operators must be considered. A mathematical construction that achieves this, and formulates a mathematical group [126], is the direct product of groups of symmetry operators. This is discussed further in Section 3.4.3.

### 3.3.5 Summarising the Construction

In short, the adapted kernel defines the global kernel as the average across all the local kernels that could meaningfully contribute to the similarity of two molecular crystals. This is done first by using knowledge of the molecule to identify different sets of analogous atoms of the underlying molecule - considering the action of one symmetry mapping at a time. Then calculating a con-

strained average kernel - considering only the corresponding local kernels - for each set, before finally averaging across those possible mapping kernels and normalising the final matrix (Figure 3.11).

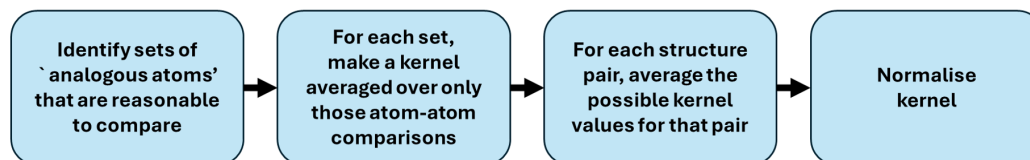


Figure 3.11: Flowchart indicating the conceptual process for constructing the adapted kernel

## 3.4 Implementation

### 3.4.1 Overview

The concept behind the adaptation having been discussed, this section outlines the computational workflow and tools used to enact the changes.

### 3.4.2 Re-indexing

The first concern for implementation is that identifying reasonable atom-atom comparisons via molecular atom indices requires the indexing to be consistent across the structure set. Specifically, it requires that the atoms of the underlying molecule will be indexed equivalently across all structures in the set and that where there are  $Z' > 1$  structures, atom indexing of the subsequent molecules of the asymmetric unit will follow the same order as that for the first molecule. For example, in a  $Z'=2$  crystal structure of a three atom molecule, the first molecule in the asymmetric unit will be indexed 0-2, and the second will be indexed 3-5 following the same order - such that 3 labels the analogous atom to that that 0 labels.

This feature holds true by default for several of the datasets explored in this work. However, for other structure sets, a re-indexing script was used to pre-process the structure set before calculation of the kernel. This script extracts all asymmetric unit molecules from a given structure. It then re-indexes each molecule to match the ordering of a reference molecule - taking the index for each atom to be that of the corresponding atom in the reference molecule, and reconstructs the crystal structure from the reindexed asymmetric unit.

Corresponding atoms between asymmetric unit molecules and the reference molecule are identified by comparing molecular graphs of the two molecules. For each isomorphism of the two graphs, the molecules are overlaid - the overlay being defined so as to reduce the RMSD between identified pairs of corresponding atoms. The final overlay is then selected to be the overlay from the isomorphism which resulted in the lowest achievable RMSD. In this way it considers all reasonable overlays - matching atoms between molecules in valid ways accounting for molecular symmetry - and finds the overlay that best aligns the two molecules. The set of corresponding atom pairs from that isomorphism is then taken to be the correct set and the target molecule is re-indexed accordingly. This ensures consistent ordering up to an isomorphism - transformation under a molecular point group operator. This is sufficient, as the impact of such transformations is accounted for by the handling of symmetry (see Section 3.3.3).

### 3.4.3 Accounting for Symmetry

In order to implement the adapted kernel for crystal structures of symmetric molecules, the relevant set of symmetry operators - or rather the atom-atom mappings under those operators - must be identified.

This is performed using the *PointGroupAnalyser* functionality in the *symmetry* module of Py-matgen [121]. This is used to return the symmetry operators of the centered molecule as rotational matrices accompanied, if necessary, by translational vectors. Then, the corresponding atom-atom mappings are obtained using a script that applies each symmetry operator in turn to the centered molecules, and compares the structural data before and after. If an  $a, b$  pair of atoms from the before and after files respectively share the same atomic species and ,to within tolerances, the same cartesian co-ordinates - then it is determined that the relevant symmetry operator maps atom  $a$  to atom  $b$ .

As discussed, where the underlying molecule has flexibility, the approach is more complex. It is insufficient simply to collate the set of all possible mappings across the included conformations as such a set may not include all **combinations** of mappings that need to be considered. Further, they may not form a mathematical group, which is a requirement to retain the PSD property of the kernel.

A solution to this problem is to construct a group from the available mappings. A useful fact here is that the direct product of two mathematical groups is itself a group [126]. Therefore, an appropriate mathematical group of mappings by can be constructed by taking the set of mappings corresponding to the direct product of two or more groups of mappings. This will by construction include all possible combinations of the relevant mappings, as required.

For a given set of included in-crystal conformations, the point groups and corresponding groups of mappings can be obtained as explained previously. Some of these groups of mappings will be subgroups of other groups in the set. These subgroups do not need to be considered when forming the direct product, as the mappings will already be included via use of the corresponding supergroup in forming the direct product. Therefore, to gather the required groups of mappings a process was used in which:

1. The groups of symmetry mappings for each in-crystal molecular conformation are identified
2. The groups of mappings are filtered to obtain only those representing supergroups of other

groups - and which are not themselves subgroups of any other group

3. The direct product of those supergroups of mappings is taken

The direct product between two groups of mappings is then obtained by calculating the overall mapping the would result from the consecutive application of each pair of mappings between the groups. The direct product of more than two groups can then be taken by performing this process consecutively, forming first the direct product of two groups and then the direct product of that result with another group - continuing in a like manner until all groups of mappings have been combined. It is not necessary to be concerned with the order of the process or the consecutive application of mappings, as the resulting direct products will be equal up to an isomorphism [128].

It was necessary here to consider ‘groups of mappings’ as distinct from molecular point groups as any two molecular conformations could share a point group - but have different corresponding groups of mappings. Therefore, the ‘filtering’ stage to identify the supergroups cannot be performed by exploring point groups alone - as this may be arbitrarily restrictive. This is a downside of the approach - which therefore requires calculation of the mappings across all in-crystal molecular conformations. This can potentially carry a large computational cost. For large structure sets, either the large set of conformations must first be filtered to identify unique conformations, or a large number of searches for the mappings must be performed.

The result of the process to identify the required group of mappings is a set of lists of ordered pairs. Each mapping in the final group is represented by a single list. Ordered pairs within the list represent the mapping, giving the molecular atom indices of the atoms that map to one another.

#### **3.4.4 Making the Kernel**

Once a consistently indexed structure set and an appropriate group of symmetry mappings has been obtained, the process of kernel construction can begin.

Both calculation of SOAP descriptors and the base-code for kernel calculation are implemented in librascal. The functionalities take as inputs ASE ‘Atoms objects’, which must be constructed from the structure files. Molecular atom indexing is maintained upon this conversion.



Then, to separate the required calculations, the set of atoms objects is split into subsets according to the  $Z'$  value of the structures, such that each different subset contains only structures of a single  $Z'$ . Sub-matrices of the full unnormalised kernel matrix are then calculated by separately performing the kernel calculation between the structure sets corresponding to each pair of  $Z'$ .

For each  $Z'$  pair kernel and corresponding structure subsets, a series of nested loops is enacted. The process proceeds as follows:

1. For each atom index  $a$ :
  - (a) For each mapping:
    - i. The secondary molecular atom index  $b$  to which  $a$  maps is extracted from the mapping data
    - ii. For each A,B structure pair in the set
      - A. Those molecular atom indices are translated to identify all the corresponding indices (within the asymmetric units only) of the ASE atoms objects of A and B
      - B. Boolean masks are applied to the atoms objects to 'select' those indices within the corresponding structures
    - iii. SOAP descriptors (with chosen parameters) are calculated for each structure - the boolean mask ensuring that descriptors are only calculated for selected atoms
    - iv. The constrained kernel matrix is calculated across the structure set. The matrix entry for each structure pair corresponds to one local kernel contribution to the relevant possible mapping kernel sub-matrix.
    - v. This 'local kernel contribution' is **added** to the relevant possible mapping kernel sub-matrix.
2. Each possible mapping kernel sub-matrix is multiplied by  $\frac{1}{\text{mol size}}$ , where mol size is the number of atoms in the molecule. This results in the set of final possible mapping kernels
3. The final adapted kernel sub-matrix for that  $Z'$  pair is obtained, by taking the kernel value for each structure pair to be the average of all the possible mapping kernel values for that pair

In the current code implementation, calculations are parallelised across mappings - i.e steps i-iv are performed for all mappings simultaneously, with the process for each mapping being executed on a different core. Additional parallelisation, potentially across atom indices or structure pairs

may be worth exploration in future work.

The SOAP-cut off radii for descriptor calculations, being a parameter the effects of which were investigated in this work (See Chapters 5, 6, and 7), was specified separately for each full kernel construction - being supplied as an argument when beginning the overall process. Additional parameters, which control factors such as the smoothing of Gaussians, and the terms considered in the spherical harmonic expansion, were maintained across all SOAP descriptor calculations. These are specified in Appendix A.

### 3.4.5 Gathering Kernels

Lastly, the kernel sub-matrices from each  $Z'$  pair must be recombined to form the full kernel matrix. This is done by concatenating the sub-matrices in order such that the final complete kernel compares structures in order (from left-right/top-bottom) of increasing  $Z'$ . For a structure set containing structures with  $Z'=1$  and  $Z'=2$ , for example, there are in theory four parts to the final kernel ( $K(Z'=1, Z'=1)$ ,  $K(Z'=1, Z'=2)$ ,  $K(Z'=2, Z'=1)$ ,  $K(Z'=2, Z'=2)$ ). In practice, however only three kernel sub-matrices need be calculated as:

$$K(Z' = 2, Z' = 1) = K(Z' = 1, Z' = 2)^T \quad (3.9)$$

These three sub-matrices, along with the transposed sub-matrix, would then be concatenated as shown in Figure 3.12.

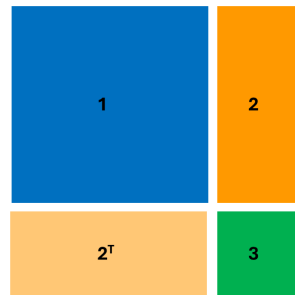


Figure 3.12: Conceptual demonstration of a full kernel matrix formed by concatenation of  $Z'$  pair sub-matrices.

Finally, the resulting kernel must be normalised such that the self similarities are equal to one - as in Equation 3.2.

### 3.4.6 Summarising the Implementation

The full process is summarised in Figure 3.13

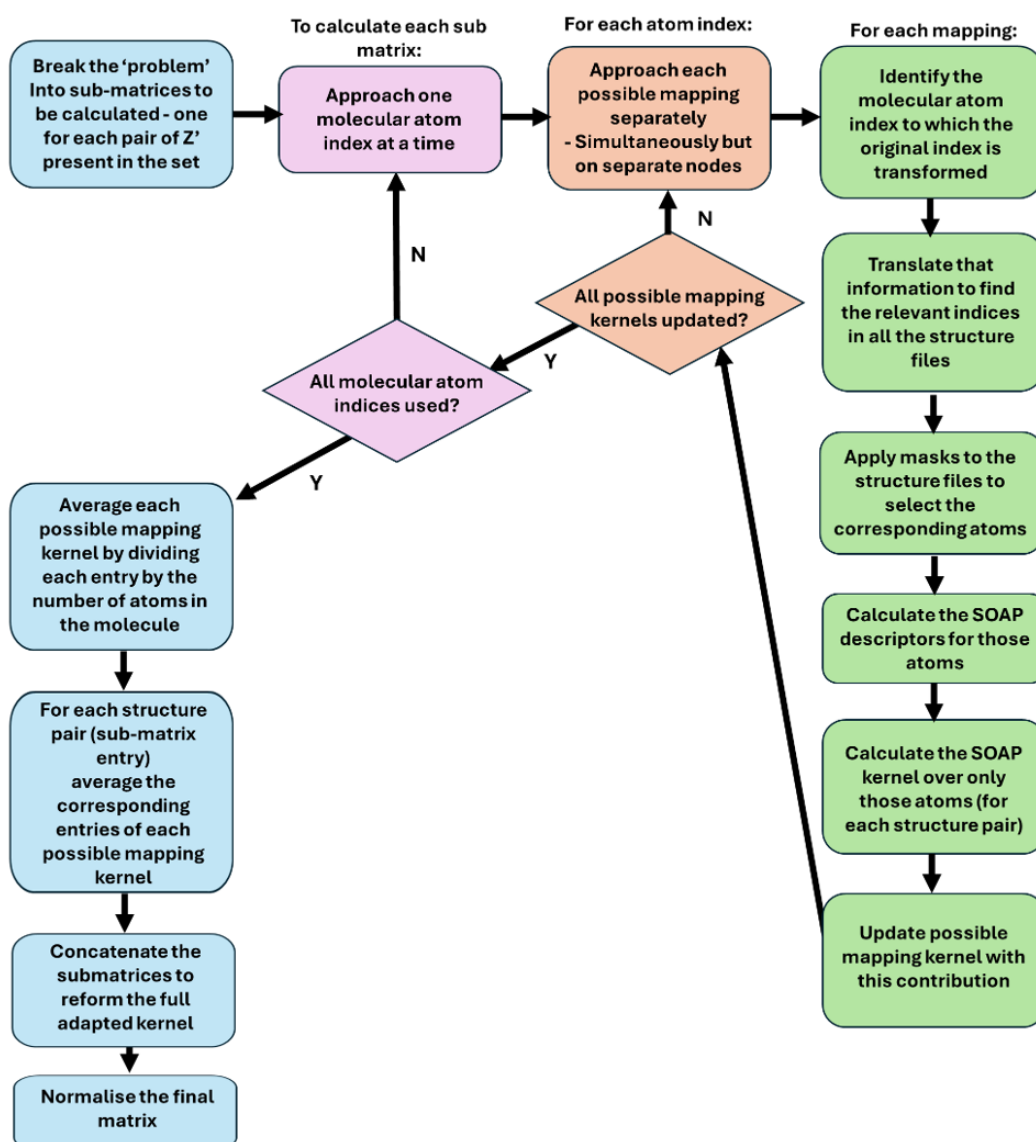


Figure 3.13: Flowchart of the full process of constructing the adapted kernel.

### 3.4.7 Forming a Cohesive Codebase

For the purposes of efficiently automating the kernel calculation process, a cohesive script was developed to conduct the process from calculation of mappings through to final kernel construction in the case of systems with a single in-crystal molecular conformation. Whilst the script has not been finalised to the point of release level code, a copy of the working script is provided in Appendix B.2. Due to time constraints, a fully combined automated script was not developed. A

script to obtain a group of mappings in the case of multiple in-crystal molecular conformations is provided in Appendix B.3. The re-indexing of structure sets was also performed separately, and a script is available in Appendix B.1.

### **3.4.8 Facing Large Datasets**

In some instances, the structure sets were large, either containing a high number of structures or underlying molecules with a large number of atoms. This leads to excessive computational cost. One possible way of alleviating this is use of a more efficient method of calculating the underlying average kernel than that within librascal. Another is to calculate the kernel in multiple parallelised tasks - each deriving the kernel for a small subset of the structures. The results can then be re-combined into a final kernel, similarly to the process in Section 3.4.5. Both of these approaches have been used on occasion for handling larger datasets in this work, to expedite calculation of the average and adapted kernels respectively. The alternative to librascal calculations is discussed further in Section 3.5.5.

## 3.5 Considerations and Concerns

### 3.5.1 Overview

Though the adapted kernel has been developed to more meaningfully define the similarity of molecular crystals, there remain further considerations to be made -potentially of interest for future work. This section discusses the key additional concerns.

### 3.5.2 The Asymmetry of Possible Mapping Kernels

One point to note with regard to the adapted kernel construction is that for each considered mapping, the kernel matrices ( or rather the corresponding sub-matrices) must be calculated completely in order to obtain accurate matrices corresponding to that mapping. The useful ‘trick’ of calculating only the values  $K(A,B)$  and then recovering the values of  $K(B,A)$  by assuming symmetry of the full kernel matrix cannot be used here. Whilst the final definition of the average kernel is symmetrical i.e  $K(A,B) = K(B,A)$ , this is not necessarily true for individual mapping kernels because

$$k(A_x, B_y) \equiv k(B_y, A_x) \not\equiv k(B_x, A_y) \quad (3.10)$$

For mappings that are their own inverse, this does not impact the symmetry of the kernel, as the final  $K_q(A,B)$  will be formed by averaging across a set that includes  $k(A_x, B_y)$  and  $K(A_y, B_x)$ , for all  $x$  and corresponding  $y$  under the mapping. For other mappings, this is not the case. However, due to the principle described in Section 3.3.3, the values of  $k(A_x, B_y)$  and  $k(A_y, B_x)$  for all  $A, B, x$  and corresponding  $y$  under some mapping will be included in the construction of the **final** adapted kernel. When the **whole** range of possible mapping kernels is considered,  $k(A_y, B_x)$  will be included in the overall final kernel due to its inclusion in  $K_{q'}(A,B)$ , where  $q'$  is the inverse of  $q$  - with  $q$  for which  $k(A_x, B_y)$  is included in  $K_q(A,B)$ .

In essence, the set of mapping kernels is a set of possibly asymmetrical kernels - for each of which the ‘other matching half’ will be found in another mapping kernel in the set - the one corresponding to its inverse (Figure 3.14).

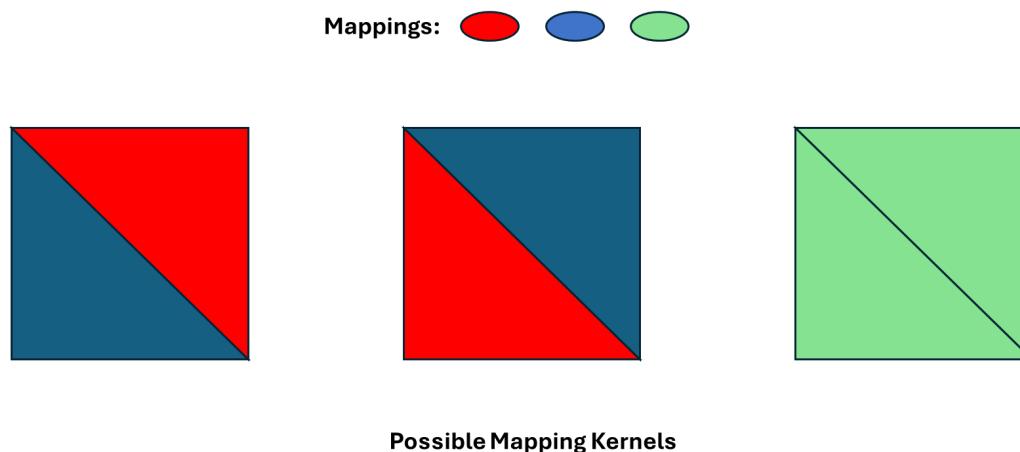


Figure 3.14: Hypothetical example of a set of possible mapping kernels for a system with three relevant mappings. Here green would represent the identity mappings and the blue and red mappings would be the inverse of one another.

This leads to a final adapted kernel that is symmetrical, but for which the underlying individual mapping kernels cannot be assumed to be symmetrical. This does not pose a theoretical problem, but does increase the cost of calculations.

### 3.5.3 The Importance of Local Symmetry

A concern with the adapted kernel construction is that, whilst the selection of included local kernels did consider comparisons of atoms analogous due to molecular point group symmetry, the impact of local symmetry was not accounted for.

If an underlying molecule has local symmetry - i.e multiple atoms that could be interchanged by rotation of bonds without altering the molecule - then this introduces an additional ambiguity in the assignment of molecular indices. There could be further reasonable atom-atom comparisons that should be included in the adapted kernel construction as a result. This consideration was left to future work, with two suggested key areas for investigation:

1. Is the consideration of locally symmetric atom pairs significant?
2. Can such consideration be incorporated without breaking the PSD property of the kernel?

### 3.5.4 Ideality

Another concern regards the extent to which reasonable definition of similarity has been lost upon changing from the ‘maximum’ to the ‘average of possibilities’ approach to forming the adapted kernel. This was briefly explored, also providing insight into comparison of the final adapted and average kernels, by exploring the ideality of the kernels. Two metrics were tested. For efficient use of research time and resources these were only applied to a single example. The test explored the similarity values of the unnormalised kernels - to explore the raw data from the nature of the structure-structure comparisons used.

One metric tested was the ‘Fraction Ideal’, which is the fraction of structures in the set for which the highest similarity between that structure and any structure in the set was the self-similarity. If the most sensible ideal comparisons had been made, then the self-similarity should be the highest similarity for every structure - even before normalisation. The second metric tested was the ratio between the average self-similarities and the average similarity between unique structures in the set. Ideally, the self-similarities should be higher than the similarity between unique structures. A greater positive ratio corresponds to a greater ability of the kernel - in terms of the included set of atom-atom comparisons - to discriminate between similar and dissimilar pairs. Table 3.1 shows these metrics for each kernel type for the NTCDA structure set - using a 4 Å SOAP descriptor cut-off.

Kernel	Mean Self Similarity	Mean Similarity between Different Structures	Ratio	Fraction Ideal
Maximum	1.0	0.94928	1.0534	672/672
Adapted	0.97385	0.94537	1.0301	422/672
Average	0.44507	0.43818	1.0157	16/672

Table 3.1: Metrics for each kernel construction showing how theoretically reasonable the results of the underlying unnormalised kernels are.

This data suggests that the maximum kernel would, in theory, be a more reasonable kernel than the adapted kernel. However, it further demonstrates that the adapted kernel - even in its average of possibilities- formulation may be more reasonable than the simple Average SOAP kernel.

### 3.5.5 Computational Costs

One limitation of the research conducted is that little investigation has been made into the comparative computational cost of different kernel calculations. This was not investigated because continual changes were being made to the conceptual approach and implementation of the adapted kernel and to the implementation of the average kernel throughout the work. Therefore, a reliable measure of the relative cost was not available.

Particularly, a key change related to the use of kernel calculations performed via librascal. This code as-is calculates average kernels sub-optimally, performing all atom-atom local kernel calculations individually [113]. This is not necessary (See Section 2.4.3), and, following averaging of the SOAP descriptors across the unit cell atoms, the average kernel for any given structure pair could be calculated with a single dot-product calculation between the two average SOAP vectors. The latter approach would be a ‘well-implemented’ kernel calculation.

Whilst quantitative comparison is not provided in this work, it can be expected that, with ideal implementation, the computational cost of the adapted kernel calculation would be higher than that of the average kernel, but there are many avenues for reducing its real-time speed. A well-implemented adapted kernel calculation would require fewer calculations of SOAP descriptors than an average kernel calculation - as descriptor calculations need only be performed for one copy of the asymmetric unit in each structure. However, when comparing well-implemented calculations, there would remain a greater cost to the adapted kernel due to a greater number of separate local kernel (i.e dot product) calculations being required.

However, qualitatively, it would appear that the cost of ideally implemented calculations would not be excessive for either kernel. Further, the real-time cost of the adapted kernel calculations can be improved as there are several avenues for potential parallelisation. By exploiting parallelisation, it is possible that the real-time cost of the key kernel calculation steps could rival that of the average kernel. However, the initial time-cost of calculating mappings must also be considered.

### 3.5.6 Choosing an ‘Original’ Kernel Construction

Following construction of an adapted kernel, it is necessary to assess the performance of this new construction. To do this, work in subsequent chapters compares -via various metrics - the utility of the adapted kernel to that of the simple average kernel. The average kernel was selected for comparisons because it was that most analogous to the adapted kernel - the adapted kernel serving



as an average kernel albeit with constraints. Another theoretically suitable choice for comparisons would have been the ReMatch kernel. However, due to time constraints and concerns over implementing the ReMatch kernel, work focussed instead on development of the adapted kernel and straightforward comparison to the average kernel. Investigation of the utility of both kernel constructions relative to that of the ReMatch kernel remains of interest however, and could be pursued in future work.

## 3.6 Concluding Remarks

Work in this chapter conceived and implemented construction of a new SOAP kernel designed to better describe the similarity of molecular crystals.

This was performed by creation of a wrapper around an existing average kernel calculation code, by applying restrictions upon the local kernel calculations included. Restrictions for local kernel inclusion were determined based upon knowledge of the underlying molecule - only comparing atomic environments between structures if the corresponding atoms could reasonably be thought to represent the same atom of the underlying molecule. This construction included taking account of point-group symmetry of the underlying molecule. This should lead to a more theoretically reasonable measure of similarity for molecular crystals. It is this adapted kernel construction that will be assessed in following chapters - by comparison to the average SOAP kernel.

However, there remain limitations. For example, the neglect of local symmetry in determining the inclusion of local kernels and the lack of investigation into the relative computational cost of kernel calculations. Addressing these areas may be of interest to future work, as may extending the subsequent assessment of adaptations by comparison to the ReMatch kernel.

## Chapter 4

# Crystal Structure Prediction

### 4.1 Overview

This chapter presents work on the crystal structure prediction of several systems, with potential applications in organic electronics, medicine, gas storage, and defence. The resulting CSP landscapes were used as datasets for later work in landscape analysis, and the prediction work is discussed here for information purposes on the derivation of the data. Prediction workflows in each case incorporated some steps of the accepted in-house approach discussed in Section 2.2, which aims to predict possible crystal structures of a given molecule, based upon thermodynamic considerations. However, work on the cases of Primidone and CL-20 deviated from this approach. In part, this was due to limitations encountered, and both the issues and the alternative approaches are discussed.

For each case of CSP performed, the workflow and computational details are discussed, and the performance of the CSP assessed via comparison of predicted structures to known experimental crystal structures of the relevant molecule.

Related work also investigated optimal parameters for removal of duplicate structures via pXRD clustering, proposing new defaults for such clustering when applied to structures optimised at the DFTB+ level of theory. This work is discussed at the end of the chapter.

## 4.2 Crystal Structure Prediction of Semi-conductors

A set of four small molecules: 1,4,5,8-naphthalene-tetracarboxylic dianhydride (NTCDA), N,N'-dimethyl-1,4,5,8-naphthalenediimide (MeNTCDI), Perylene-3,4,9,10-tetracarboxylic dianhydride (PTCDA), and N,N'-dimethylperylene-3,4,9,10-bis(dicarboximide) (MePTCDI) with potential applications as organic semiconductors [129–131] and pigments [132] was explored. The structures of these molecules can be seen in Figure 4.1.

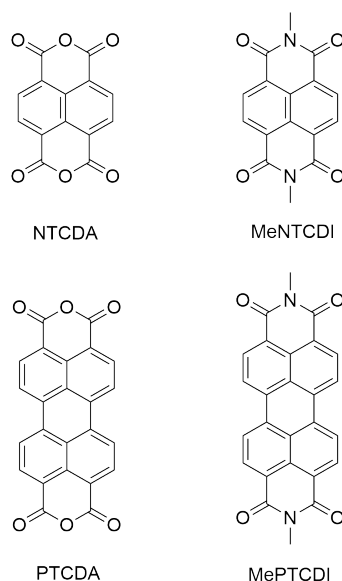


Figure 4.1: Structures of the four semiconductor-type molecules

Prediction work in these cases proved straightforward. For each system, a  $Z'=1$  rigid-molecule quasi-random CSP search was performed, with optimisation of crystal structures conducted at the interatomic forcefield level of theory.

The workflow used for each system was as follows:

1. The gas-state conformer of the molecule was derived via optimisation at DFT-level (B3LYP [133–135] + GD3BJ/6-311G\*\*), implemented in Gaussian09.
2. Distributed atomic multipoles for the molecule, in its optimised conformation, were then calculated at DFT level (PBE0/6-31G\*\* [136–138]), via GDMA.
3. Using CSPy, trial crystal structures were quasi-randomly generated and lattice energy minimised using pairwise interatomic forcefield FIT, with additional handling of permanent electrostatic using the derived distributed atomic multipoles (FIT+DMA), until 10,000 fully optimised crystal structures had been generated in each of the ten most common space groups for organic molecules [124] ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ).

4. Duplicate removal was performed via pXRD clustering, according to the in-house ‘default’ approach (identifying duplicates that had cosine similarity  $\geq 0.8$  and cDTW distance  $\leq 10^\circ$ ).
5. Additional duplicate removal was performed via the in-house default approach to ‘COMPACT’ clustering.

It is important at this point to recall as an aside that the ‘clustering’ used here and in subsequent CSP work in this thesis refers simply to a duplicate removal process and not to the application of traditional clustering algorithms.

Each landscape was then searched for predicted structures that corresponded to known experimental crystal structures of the molecule. ‘Matches’ to experimental structures were declared in cases for which 30/30 molecules of representative clusters of the experimental and predicted structures could be overlaid within tolerances of 0.2 Å and 0.2 ° using the *CrystalPackingSimilarity* functionality in the CSD API. Hydrogen Positions were ignored. This is akin to COMPACT searching previously discussed (Section 2.2.15) and such matches may be referred to as COMPACT matches in this thesis. The experimental structures used for comparison can be found in the CSD [105], using the reference codes as follows : NTCDA - KENDEM [139] , MeNTCDI - DAHMUX [140] , PTCDA ( $\alpha$ ) - SUWMIG02 [132] , PTCDA ( $\beta$ ) - SUWMIG03 [132], MePTCDI - DICPAG01 [130]. Matches to the known experimental structures were found in all cases, yielding the results shown in Table 4.1.

System	No. Unique Structures	Energetic Ranking of Match	RMSD <sub>30</sub> of Match (Å)
NTCDA	672	1	0.336
PTCDA	1186	1 ( $\alpha$ )/2 ( $\beta$ )	0.376 ( $\alpha$ )/0.360 ( $\beta$ )
MeNTCDI	2084	1	1.063
MePTCDI	4667	1	0.270

Table 4.1: Results of crystal structure prediction on 4 organic semi-conductor molecules

Successful recovery of a known experimental structure as the global thermodynamic minimum on the predicted landscape for small rigid molecules such as these is positive, but expected [54]. Figure 4.2 gives an example of one of the predicted landscapes, for PTCDA, with the well-ranked matches to experimental structures indicated.

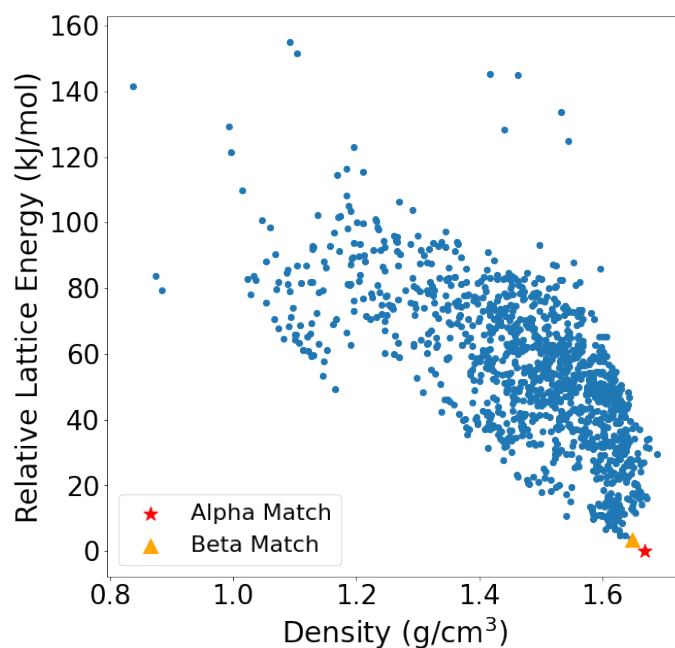


Figure 4.2: Example CSP landscape, here of predicted crystal structures of PTCDA. Matches found to the experimental  $\alpha$  and  $\beta$  polymorphs are shown.

Surprising, however, is the poor quality of the match in the case of MeNTCDI, which is visualised in Figure 4.3. There is no one-to-one correspondence between the RMSD value and the precise similarity of crystal structures. However, a useful guideline is that an overlay with an  $\text{RMSD}_{30} \leq 0.3 \text{ \AA}$  should correspond to a strong match as such variation is on a similar order of magnitude to the level of variation in experimental crystal structures at room temperature and at low temperatures [54]. The reason for this poor overlay was not fully investigated, though one possibility is that the packing is influenced by rotational freedom of the methyl groups - which was ignored in predictions.

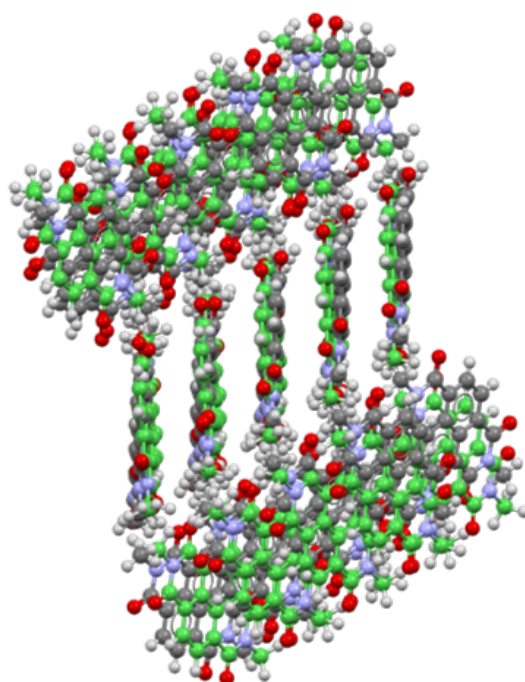


Figure 4.3: Overlay of 30-molecule clusters of the experimental structure of MeNTCDI (DAHMUX)(element-Colour) and the predicted global minimum (green)

## 4.3 Crystal Structure Prediction of Primidone

### 4.3.1 Overview

2-desoxyphenobarbital (primidone) (Figure 4.4) is an anticonvulsant drug used to treat epilepsy [141] and essential tremor. Its active metabolite is a similar molecule, phenobarbital [142]. There are four known polymorphs of primidone. Two of these ( $\alpha$  [141] and  $\beta$  [143]) are well-known and fully characterised. The remaining two are novel polymorphs, recently synthesised through supercritical anti-solvent precipitation. These are not fully characterised but pXRD patterns are available [144].

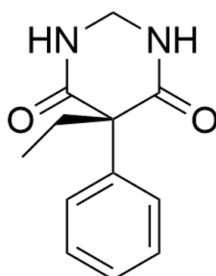


Figure 4.4: Molecular structure of primidone

Primidone provided an interesting case for producing data to test the GCH due to its flexibility, multiple polymorphs, and potential for further as yet undiscovered polymorphism. It had further potential as a test to flexible CSP in investigating whether predictions could suggest structures for the uncharacterised polymorphs - although this has only been partially investigated here.

### 4.3.2 Structure Prediction Process

#### Conformational Search

The primidone molecule is similar to phenobarbital albeit with additional flexibility in the heterocyclic ring. There are a total of three important intramolecular degrees of freedom - the heterocyclic ring conformation, the phenyl torsion angle ( $p$ ) and the ethyl torsion angle ( $e$ ). These are seen in Figure 4.5.



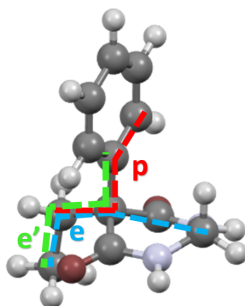


Figure 4.5: Primidone molecule with important flexible torsions indicated

Due to these conformational degrees of freedom, an effective CSP search for primidone requires use of flexible-molecule CSP. The first step in conformational sampling was to identify stable conformers of the molecule. The approach to this was inspired by a similar study of phenobarbital [145]. The minima on the conformational energy surface were found using two-dimensional torsional scans. These were performed using DFT (PBE0/6-311G\*\*) in Gaussian09 as a function of  $p$  (Range:  $\pm 90^\circ$ , Step Size:  $30^\circ$ ) and  $e'$  (Range:  $180 - 0^\circ$ , Step Size:  $30^\circ$ ). Here,  $e'$  was used in place of  $e$  for functionality of the scans. To then derive conformational energy surfaces of primidone as a function of  $p$  and  $e$ , it was assumed that  $e = e' + 180$ . A separate energy surface was derived for each of three key heterocyclic ring conformations (two boat and one chair conformation) which had been identified via preliminary conformer searching. These are indicated in Figure 4.6.

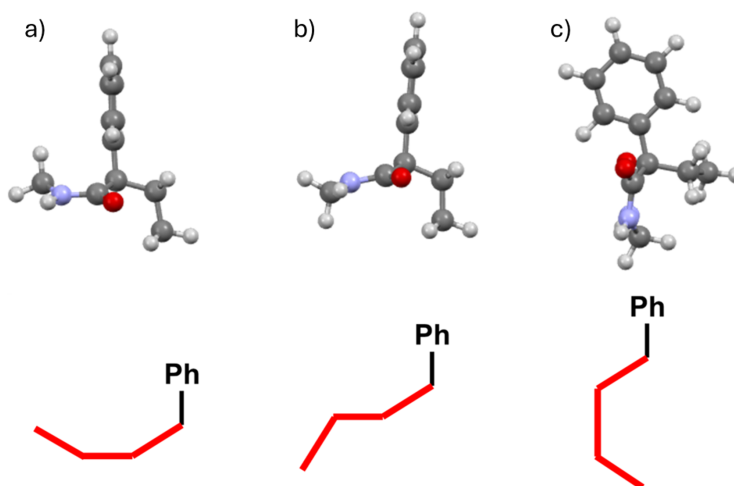


Figure 4.6: Enter Caption

The resulting conformational energy surfaces are shown in Figure 4.8. For the cases of the ring

conformations *b* and *c*, geometry optimisations within the scans were constrained - with some atoms frozen to forbid ring-flipping.

The optimised structures corresponding to each local minimum on the conformational energy surfaces were then extracted from the torsional scans. For cases of pairs of conformers that are enantiomers - only one enantiomer was taken. For the case of ring conformation *b*, it was thought that the lack of symmetry of the conformational energy surface (i.e the presence of minima at  $(\pm 60^\circ, \pm 120^\circ)$  and  $(\pm 90^\circ, \mp 150^\circ)$  rather than at  $(\pm 90^\circ, \pm 150^\circ)$  and  $(\pm 90^\circ, \mp 150^\circ)$ ) was likely an artefact of the scanning process and that the conformers at  $(\pm 90^\circ, \pm 150^\circ)$  represented more reasonable minima. This resulted in a set of initial conformers shown in Figure 4.7.

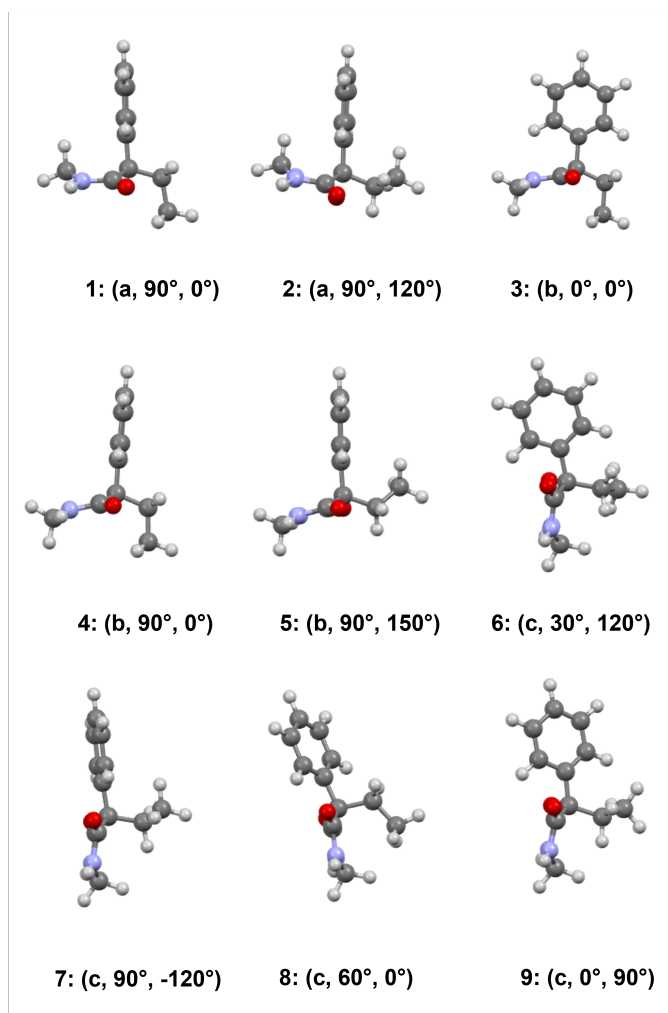


Figure 4.7: Conformers of primidone extracted from torsional scans with torsions indicated - (heterocyclic ring conformation, phenyl torsion angle, ethyl torsion angle)

It was then required to obtain a set of locally distorted conformations, for the CSP search to

sample the conformational energy surface around the basins. These were obtained using MOLDIS [146] - allowing variations of  $\pm 90^\circ$  around the phenyl angle and  $\pm 60^\circ$  around the ethyl angle with a step-size of  $30^\circ$  for each conformer. The resulting set of conformations - mapped on to the conformational energy surface- is shown in Figure 4.8, displaying the final conformational sampling of the surface used for CSP. A conformer search was also attempted via use of CREST conformer sampling, as is an accepted step within the in-house approach to flexible CSP, but this did not recover all of the expected conformational minima.

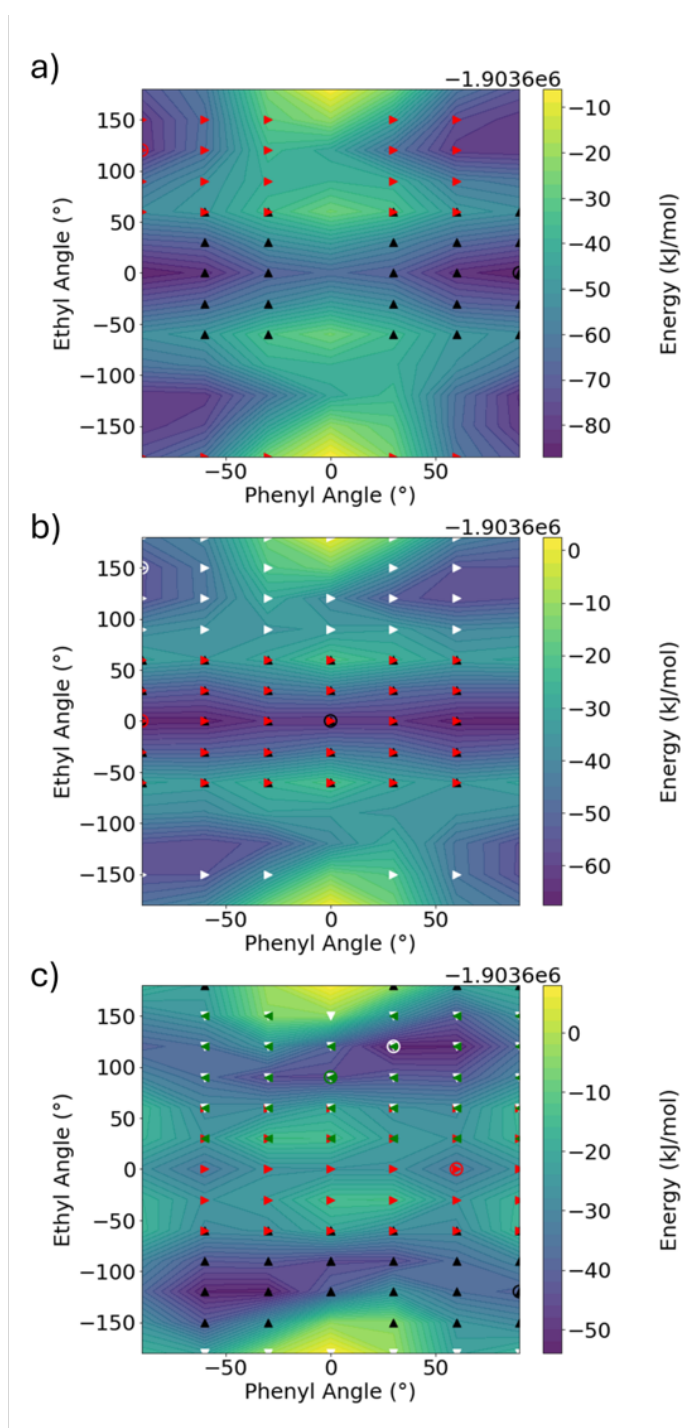


Figure 4.8: Conformational energy surfaces of primidone with *a*, *b*, and *c* heterocyclic ring conformations and MOLDIS conformational sampling shown as overlaid points. Colours of points separate the original conformers on each surface from which a distorted conformation was created, with the original conformers indicated by rings of the same colour

### Initial CSP Search

A quasi-random CSP search was then performed allowing use of all conformations with a relative energy of  $\leq 30$  kJ/mol on the conformational energy landscape. The search was performed across 26 space groups with  $Z'=1$  ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ,  $P1$ ,  $Pbcn$ ,  $P1c1$ ,  $P2_12_12$ ,  $Fdd2$ ,  $Pccn$ ,  $P12/c1$ ,  $I4_1/a$ ,  $R\bar{3}$ ,  $P4_1$ ,  $P4_32_12$ ,  $P4_12_12$ ,  $P4_3$ ,  $P3_2$ ,  $P3_1$ ,  $P6_1$ ) and 13 space groups with  $Z'=2$  ( $P\bar{1}$ ,  $P12_1/c1$ ,  $P12_11$ ,  $P2_12_12_1$ ,  $P1$ ,  $Pca2_1$ ,  $Pna2_1$ ,  $C12$ ,  $c1$ ,  $Pbca$ ,  $C12_1$ ,  $C1c1$ ,  $P1c1$ ,  $P2_12_12$ ). These groups were chosen such that all space groups comprising  $\geq 0.05\%$  of known organic molecular crystals [124] of the corresponding  $Z'$  were searched. In the  $Z'=1$  search, 100 000 successfully lattice-energy minimised structures were generated in each space group, with the search being extended to 200 000 structures in space groups  $P12_1/c1$  and  $C12/c1$  due to known difficulty in sufficiently sampling those spacegroups. The sampling was doubled in the  $Z'=2$  search.

Trial structures were initially lattice-energy minimised in CSPy, implementing the FIT [78] force-field and permanent electrostatic contributions from only point charges calculated at DFT level (PBE0/6-311G\*\*), under the philosophy that such calculation was sufficient to derive an initial landscape to be re-optimised.

At this stage, due to complexities of implementation, only partial duplicate removal was performed. Duplicate structures were removed from the set of structures in each spacegroup for each  $Z'$  by pXRD clustering across the corresponding individual sets. This pXRD clustering stage used the default thresholds previously discussed.

### Re-Optimisation

Re-optimisation of the important region of the landscape was performed in two stages - DFTB+ re-optimisation and pDFT re-optimisation.

DFTB+ re-optimisation was performed on all structures falling within 25 kJ/mol of the global minimum on the forcefield-level landscape. All DFTB+ re-optimisations were performed using the 3OB parameter set [147].

A slightly altered approach to duplicate removal was then performed. Initially, again across individual sets of a given  $Z'$  and spacegroup, pXRD clustering was performed. However, it had been identified via a previous attempt at CSP for primidone that the default approach to pXRD

clustering was insufficient for even coarse duplicate removal on landscapes calculated at DFTB-level. Work to identify effective, but risk-averse thresholds for clustering (Discussed in Section 4.6) identified ideal thresholds for such cases. pXRD clustering was then performed on each set, such that structure pairs whose pXRD patterns had a determined cosine similarity  $\geq 0.4$  and a cDTW distance  $\leq 10^\circ$  were considered duplicates.

Finer duplicate removal was then performed, by use of the in-house default approach to COMPACK clustering across the entire structure set.

Further re-optimisation was then performed using periodic Density Functional Theory, on all structures lying within 12 kJ/mol of the global minimum on the DFTB-level landscape. pDFT optimisations was itself a two-step process:

1. Geometry optimisation was performed on each structure, allowing relaxation of only atomic positions within a fixed unit cell
2. Further geometry optimisation was performed on each structure, allowing simultaneous relaxation of atomic positions and unit cell parameters

Both steps were performed using plane-wave basis sets with a 500 eV cut-off - implemented in VASP and using k-points selected to ensure a maximal k-point spacing of  $0.05 \text{ \AA}^{-1}$ . For some cases, due to an error in implementation, the second step was run initially with fixed k-points, placed by default in the VASP implementation. All final optimised structures, however, are from optimisations using a  $\leq 0.05 \text{ \AA}^{-1}$  k-point grid. As such, this error may have impacted the pathway to optimisation but is not expected to have affected the final optimised structures.

Initially, work aimed to ensure that each calculation step was converged such that the total self-consistent-field energy was consistent to within  $1 \times 10^{-7} \text{ eV}$ . However, this strict criterion proved unfeasible for some structures, and the convergence criteria were loosened as necessary. All successful calculations were converged so that, at worst, the allowed difference in energy between steps was  $1 \times 10^{-5} \text{ eV}$ . A very small number of optimisations did not converge in a practicable time period even to that minimal level of precision and as such were removed from the final set.

### 4.3.3 Restoring Symmetry

Due to time constraints, in light of encountered limitations in utilising the set of  $Z'=2$  predicted crystal structures (Section 3.3.4), full analysis of results was only conducted for  $Z'=1$  predicted crystal structures. However, the full prediction set was retained and could be of interest for future work, particularly if the aforementioned issues are resolved.

However, the investigated  $Z'=1$  prediction set is not merely that resulting from the initial  $Z=1$  quasi-random search and following (re)optimisations. Rather, it is the set of structures for which a  $Z'=1$  unit cell could be found. This is an important distinction because some structures generated as  $Z'=2$  structures in a given spacegroup could alternatively be described as  $Z'=1$  structures of a higher-symmetry spacegroup. Additionally, structures that were  $Z'=2$  could become more symmetrical under re-optimisation. Therefore, to gain the maximal  $Z'=1$  set, structures were searched for possible additional symmetry. This was important also to restore the symmetry lost in conversion of structure files to those defining the crystal structures in P1 - necessary for implementation of re-optimisation. It was important to recover this symmetry in order to obtain structures of a workable  $Z'$ .

In order to identify definitions of the crystal structures with maximal symmetry, and so minimal  $Z'$ , the *ADDSYM* functionality within PLATON was applied to each of the structure files. This is a functionality implementing the *MISSYM* [148] algorithm to detect symmetry or near symmetry elements present in the extended structure, but not defined within the structure file. When successful, PLATON returns symmetrised versions of the crystal structures, that is it returns structure files that define the same crystal, albeit using a smaller asymmetric unit and a higher symmetry space group. Of course, within the implementation, the identified symmetry is identified not absolutely, but within given tolerances. Loosening of the tolerances can lead to greater restoration of symmetry. As such, the symmetry search was run in three progressive steps, with increasingly loose tolerances. The altered parameters were the non-fit percentage - the allowed percentage of present atoms that do not-fit within the tolerances for the proposed higher symmetry- and the *TolMetric*, an angular tolerance used when searching for two-fold axes. (Table 4.2). The tested parameter sets proved sufficient, leading to successful recovery of symmetry in the vast majority of cases across the runs. However, one flaw is that the choices made were somewhat arbitrary. If necessary, further work could investigate systematic parameter testing for the use of PLATON *ADDSYM*.

Step	Parameters	
	TolMetric(°)	Non-Fit
1 (Default)	1	20%
2	3	15%
3	3	30%

Table 4.2: Parameters used in each run of the *ADDSYM* search. All remaining parameters were left at the PLATON default values

Such symmetry restoration was performed on all structures in the respective structure sets at each level of theory (Forcefield, DFTB+, and pDFT) and a resulting complete  $Z'=1$  set gathered for each level of theory. An error, noticed late on in the process, led to a small number of structures for which the symmetry had not been recovered at a given step failing to be put through to the next stage. A precise number of these cases is not known, but is expected to be very low. The impact on results is not expected to be significant as the number of wrongfully excluded structures should be small, and the issue did not lead to any inclusion of invalid or erroneous structures in the datasets.

In order to ensure that the crystal structures returned by the symmetrized representations were equivalent to those from the original representations - a  $\text{RMSD}_{30}$  search *CrystalPackingSimilarity* was run between each symmetrical representation and its corresponding original representation. Any problematic cases with  $\text{RMSD}_{30} < 0.25 \text{ \AA}$ , alongside any structures which encountered failures during the *ADDSYM* process, were removed from the datasets. The resulting  $Z'=1$  structure sets were then each COMPACK clustered to remove duplicates.

#### 4.3.4 Results

At each level of theory, the resulting  $Z'=1$  structure sets were searched for matches to the two fully characterised polymorphs ( $\alpha$ [141] and  $\beta$ [143]). The forcefield-level set used was restricted to structures falling within a 25 kJ/mol window of the global minimum. Searches for matches were performed as previously using *CrystalPackingSimilarity* feature in the CSD API - searching for 30/30 'COMPACK matches' within tolerances of 0.3  $\text{\AA}$  and 30° or 0.2  $\text{\AA}$  and 20° as stated.

Table 4.3 shows the final number of predicted structures at each stage, along with the relative



energy ranking, and overlay RMSD<sub>30</sub> of found matches. Both fully characterised known polymorphs can be defined as  $Z'=1$ , and so should be identifiable within the  $Z'=1$  prediction sets if successful. Indicated matches in each case are the lowest energy match.

Calc level	No. Unique Structures	Energetic Ranking of Match ( $\alpha/\beta$ )	Relative Lattice Energy (kJ/mol) ( $\alpha/\beta$ )	RMSD <sub>30</sub> (Å) ( $\alpha/\beta$ )	Tolerances
Forcefield	1693	1/113	0.00/13.82	0.308/0.253	0.3/30
DFTB+	1061	31/28	4.30/4.13	0.581/0.537	0.2/20
pDFT	264	6/9	2.85/4.43	0.575/0.447	0.3/30

Table 4.3: Structure sets sizes and experimental match results from each stage of re-optimisation of  $Z'=1$  crystal structure prediction of primidone

These results emphasise the impact of the chosen energy model on the energetic rankings of the resultant CSP landscapes. The ranking of the  $\beta$  polymorph is greatly improved upon re-optimisation, although the  $\alpha$  polymorph loses its status as the global thermodynamic minimum. It should be noted that as there exist two uncharacterised polymorphs it is possible that one or both of these are lower in lattice energy than the characterised polymorphs. It is therefore not a given that one should expect to find either the  $\alpha$  or  $\beta$  polymorph as the global minimum on the energy landscape. It is additionally worth noting that, whilst the energetic rankings of the found matches on the DFTB+ landscape are poor, this may be in part simply due to the population density of the landscape, as the relative energies themselves are within the important low-energy window.

Interestingly, the quality of the structures appears to have worsened, with notably higher RMSD<sub>30</sub> values of the overlays at higher levels of theory. Additionally, the predicted structures corresponding to the matches differed between landscapes at different levels of theory. One possible investigation of interest may be to perform pDFT single-point energy calculations upon the forcefield landscape to determine whether the favourable energetic rankings could be recovered without loss of structural quality.

#### 4.3.5 Uncharacterised Polymorphs

Some initial investigation into the possibility of predicted structures corresponding to the uncharacterised polymorphs was made - although work is not complete. pXRD patterns corresponding to the uncharacterised polymorphs [144] were digitized using automeris.io [149] a web-app designed to extract the data represented in images of data plots. The digitized patterns were then compared to simulated patterns for the low-energy DFTB-level predicted structures using an in-house code to compare patterns based upon a cDTW distance metric. The simulated patterns of the most promising candidates from this search were visually compared to the experimental pXRD patterns. This led to identification of one potential match to form C (Figure 4.9).

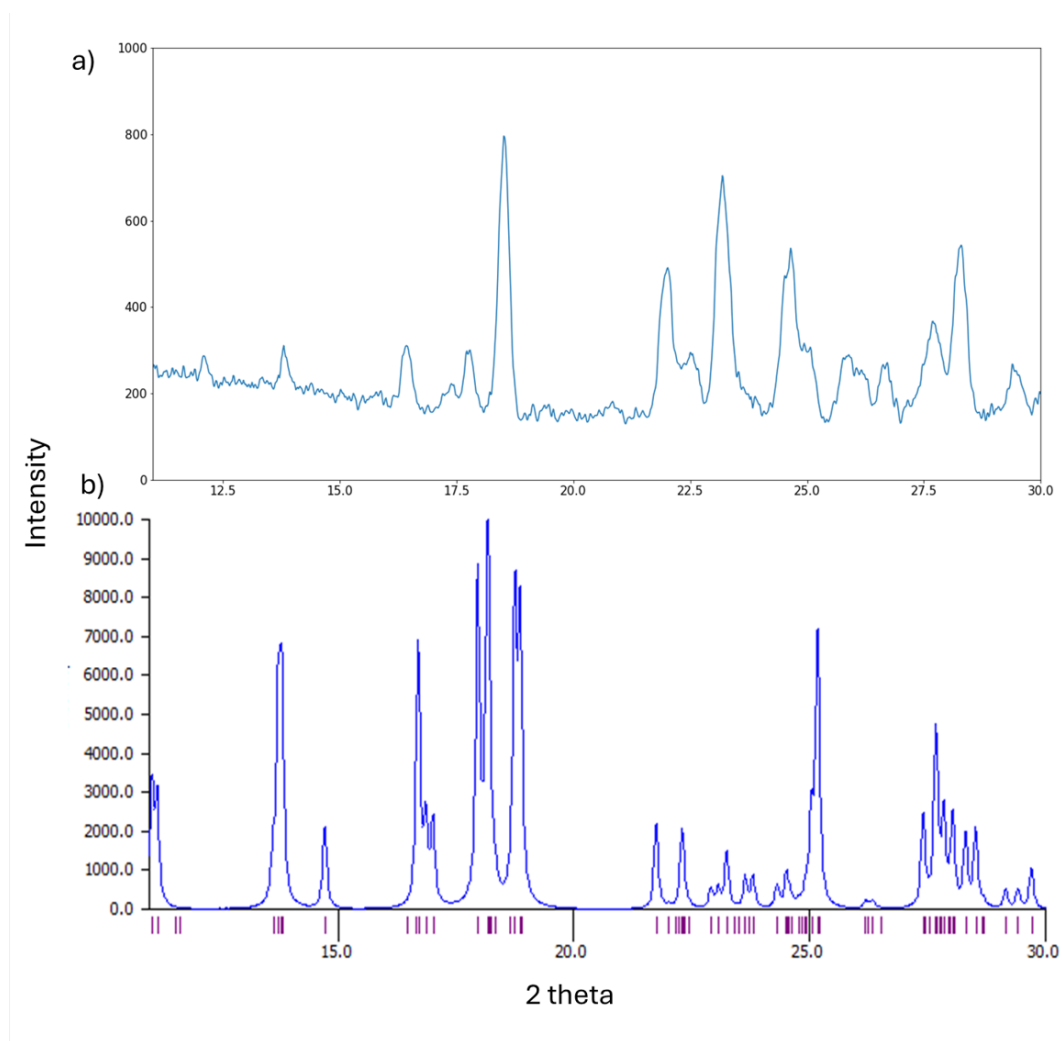


Figure 4.9: Comparison of a) pXRD pattern of uncharacterised form C of primidone - digitized from Figure 7b in ref [144] and b) simulated pXRD pattern of a predicted crystal structure

Whilst the peak heights vary significantly between the patterns, this can be influenced significantly by factors such as preferred orientation [150]. Meanwhile, the peak positions are well aligned between simulated and experimental patterns, making the structure a promising candidate for an

experimental match.

Further investigation began to explore additional testing for matches to the pXRD patterns. Full pXRD patterns for the uncharacterised polymorphs (covering  $2\theta$  range 5-50) were obtained by request[151] and attempts were made to identify possible matches to predicted structures using a variable cell powder pattern comparison algorithm (*VC-xPWDF*) [152] implemented in *critic2* [153]. However, this required prior indexing of the patterns - which proved difficult, providing many low-quality guesses for indexing and in turn large numbers of inconsistent rankings of potential matches. Thus far little has been gained from these attempts.

One approach of future interest could be to use a Monte Carlo simulated annealing approach that aims to optimise a pseudo-energy incorporating similarity of a predicted structure to experimental pXRD data. Beginning a trajectory of this algorithm from the potential match structure identified previously could lead to an optimised structure that better corresponds to the pattern [84].

## 4.4 Crystal Structure Prediction of DAP

2,6-diaminopurine (DAP) (Figure 4.10) is a small organic molecule investigated primarily for its pharmaceutical and biological potential. However, it can also adopt solid-state structure with potential porous properties, having been shown to have two experimental Hydrogen Organic Framework (HOF) structures. HOFs are extended structures of organic molecules connected by hydrogen-bonds. Whilst there exist other pure polymorphs of DAP in addition to a known hydrate structure [154], it is the porous solid state structures that were of interest in this thesis, and which were targeted for recovery in crystal structure prediction.

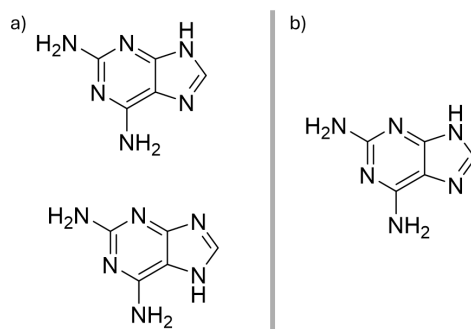


Figure 4.10: Possible tautomers of DAP (a) and the tautomer used for CSP work, being present in both experimental HOF structures (b)

CSP work in the case of DAP employed the quasi-random workflow implemented in CSPy, at interatomic forcefield (FIT + DMA) level without further re-optimisation. Two rigid-molecule searches were performed, using the optimal gas-phase conformer at DFT-level (PBE0+GD3BJ/6-311G\*\*). Multipoles were calculated at the same level of theory.

1. 10 000 successfully optimised  $Z'=1$  crystal structures in each of the 25 most common spacegroups for organic systems [124] ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ,  $P1$ ,  $Pbcn$ ,  $P1c1$ ,  $P2_12_12$ ,  $Fdd2$ ,  $Pccn$ ,  $P12/c1$ ,  $I4_1/a$ ,  $R\bar{3}$ ,  $P4_1$ ,  $P4_32_12$ ,  $P4_12_12$ ,  $P4_3$ ,  $P3_2$ ,  $P3_1$ )
2. 100 000 successfully optimised  $Z'=3$  crystal structures in spacegroup  $P\bar{1}$

The  $Z'=3$  search was restricted to spacegroup  $P\bar{1}$ , as this was known to be the  $Z'$  and spacegroup adopted by one known HOF form (HOF-2) [154]. CSP sampling in  $Z'=3$  was considered too expensive to justify a full search of all common space groups. Similarly, to restrict the search problem for feasibility, both searches used only a single tautomer of DAP, the tautomer present in both fully characterised HOF structures [154] (Figure 4.10)

The landscapes resulting from each search were analysed separately. This is likely not representative of how CSP would be applied in real word implementations, especially for assessing risk, as solid state structure adopted depends upon competition with all possible solid forms, regardless of  $Z'$ . However, it was deemed sufficient for the purposes of this thesis, i.e for deriving a potentially porous CSP landscape to be analysed via machine-learned descriptors.

The  $Z'=1$  and  $Z'=3$  landscapes were searched for matches to the known HOF-1 and HOF-2([154]) structures respectively, defining matches as previously discussed - applying tolerances of 0.2 Å and 20 °. The results of this searching are shown in Table 4.4.

Landscape	No. Unique Structures	Energetic Ranking of Match	Relative Lattice Energy (kJ/mol)	RMSD <sub>30</sub> (Å)
$Z'=1$ (HOF 1 Search)	5825	2	3.905	0.316
$Z'=3$ (HOF 2 Search)	301887	1	0.000	0.357

Table 4.4: Structure sets sizes and experimental match results from each landscape calculated in crystal structure prediction of DAP

Interesting here is the presence of a match to the porous HOF 2 and HOF 1 structures as the first and second ranked structures on their respective landscapes. Porous polymorphs often lie higher on the landscape, within high-energy low-density ‘spikes’.

Figure 4.11 shows the overlay between predicted crystal structures of DAP and the known experimental forms of DAP-HOF-1 and DAP-HOF-2, showcasing both the quality of the predicted structures and the porous properties of the solid-state forms.

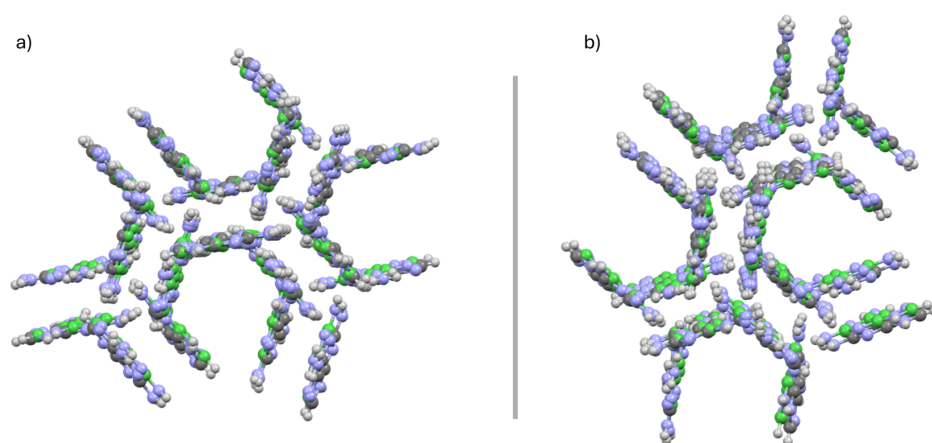


Figure 4.11: Overlays between predicted crystal structures of DAP (green) and the known experimental forms of a) DAP-HOF-1 and b) DAP-HOF-2 (element-colour)

## 4.5 Crystal Structure Prediction of CL-20

### 4.5.1 Overview

Hexanitrohexaazaisowurtzitane (CL-20) is an energetic material with four conformational polymorphs, including a high-pressure polymorph [155]. This presented an interesting case for this thesis, particularly in testing the GCH methods of identifying stabilisable structures. The uncommon molecular geometry (Figure 4.12) also acted as test of the CSP methods used.

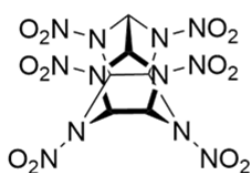


Figure 4.12: Structure of CL-20 molecule

### 4.5.2 Unsuccessful CL-20 CSP

#### Overview

An initial attempt to predict the CSP landscape of CL-20, aimed to implement a workflow similar to the approach used in the case of primidone. That is, prediction work attempted to generate molecular conformations, perform quasirandom CSP with FIT+DMA, followed by re-optimisation using DFTB+ followed by pDFT. However, significant issues were encountered at both the conformational sampling and DFTB+ re-optimisation stages. These issues are discussed here by way of explanation for the altered workflow, and for noting possible concerns that could affect future CSP work.

#### Conformational Generation

To generate conformations to be placed into trial crystals, an initial conformer was determined using minimisation with the UFF forcefield [156] implemented by the auto-optimisation feature in Avogadro [157]. Further conformer searching was not performed at this point, under the assumption that the flexible improper torsions within the CL-20 molecules (Figure 4.13) would be sufficiently handled via re-optimisation in the later stages of the CSP process, and as such would not need to be explicitly sampled. This assumption later proved to be incorrect, as attempted DFTB+ re-optimisation did not lead to changes in these improper torsions.

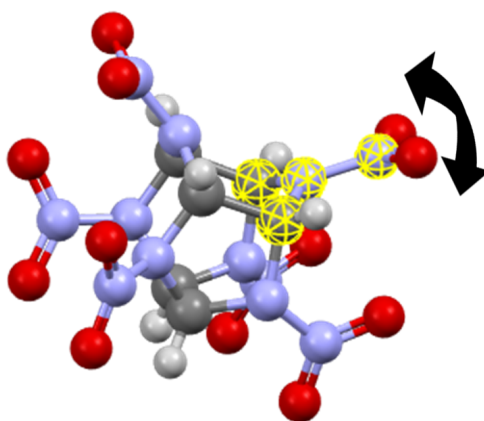


Figure 4.13: Example of 3D molecular conformation of CL-20, with one of six analogous improper torsions indicated by the yellow highlighted atoms

Some conformational sampling around this minimum was attempted however. Implementing conformer distortion using MOLDIS, work attempted to sample rotations of the nitro groups across a range of  $\pm 90^\circ$  in  $45^\circ$  steps. This, again, later proved problematic - potentially due to too large a step-size having been taken - as the global minimum conformation of those sampled lay significantly below all others. This meant that after forcefield CSP had been performed, this conformation dominated the low-energy landscape.

These two encountered issues led to unfeasibly poor prediction, as the dominant conformation was a poor match to all in-crystal conformations in the known experimental forms.

### Forcefield level CSP

CSP was then attempted using the flexible workflow in CSPy, randomly sampling conformations within a 40 kJ/mol window of the global minimum conformation and optimising structures using FIT+DMA. No issues were directly encountered during the process but, as noted, the important region of the resulting landscape was dominated by a single conformation.

### DFTB+ Re-optimisation

Prior to discovery of the extent of these limitations, re-optimisation of the lowest 30 kJ/mol of the landscape was attempted using DFTB+ with the 3OB parameter set.

A large proportion of the attempted re-optimisations resulted in non-physical structures, in which



the intramolecular bonds such as N-N bonds of the underlying molecules were broken.

In light of the issues with the conformational sampling, it was possible that the issues with DFTB+ re-optimisation were specific to the single conformation present in the vast majority of structures. To eliminate this explanation, the experimental crystal structures were also re-optimised using DFTB+ - and the same problem occurred (See Figure 4.14), thus leading to the conclusion that this method of re-optimisation would not be appropriate for CL-20 crystal structures of any form - possibly due to issues with the applicability of the parameter set.

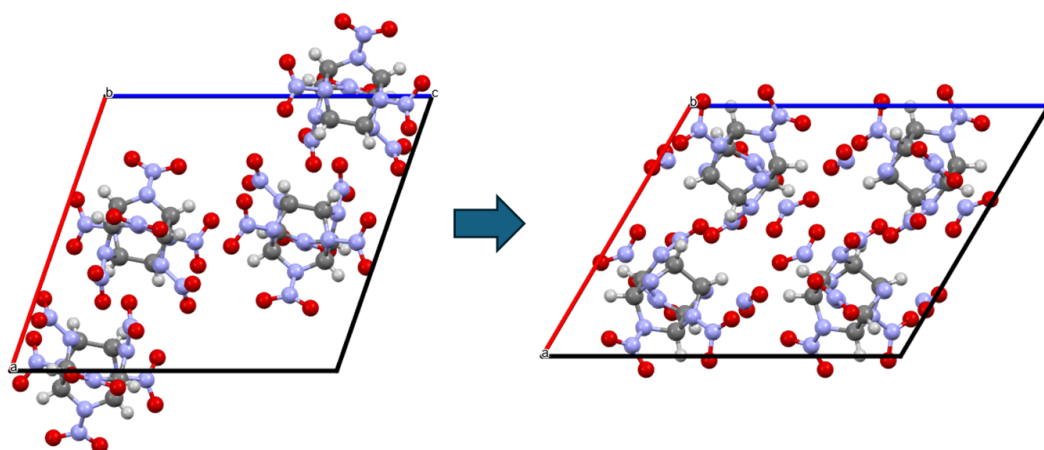


Figure 4.14: Experimental CL-20 structure example ( $\gamma$  [158]) after DFTB+ re-optimisation, showing breaking of intramolecular bonds.

### 4.5.3 Successful CL-20 CSP

#### Structure Prediction Process

Taking on board the discussed issues, a second attempt was made to predict CL-20 crystal structures. This section discusses the adjusted CSP workflow used for the system. Crystal Structure prediction was performed for CL-20 as follows:

1. An initial gas-phase conformer of the molecule was calculated at DFT-level (PBE0+GD3BJ/6-311G\*\*)
2. Conformer searching was performed from that starting point using mCREST - identified conformers being optimised at DFT-level (PBE0+GD3BJ/6-31G\*\*).
3. Conformers were distorted using MOLDIS ( $\pm 22.5^\circ$  of each nitro group in a single step each way)

4. Forcefield-level quasi-random CSP was performed in CSPy (Sampling 100 000  $Z'=1$  structures in each of the 26 most common space groups [124] ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ,  $P1$ ,  $Pbcn$ ,  $P1c1$ ,  $P2_12_12$ ,  $Fdd2$ ,  $Pccn$ ,  $P12/c1$ ,  $I4_1/a$ ,  $R\bar{3}$ ,  $P4_1$ ,  $P4_32_12$ ,  $P4_12_12$ ,  $P4_3$ ,  $P3_2$ ,  $P3_1$ ,  $P6_1$ ) - excepting 200 000 in spacegroups  $P12_1/c1$  and  $C12/c1$  and using FIT + DMA, with multipoles calculated at DFT level (PBE0/6-311G\*\*))
5. Duplicates were removed from the landscape using the default approach to pXRD clustering within each individual spacegroup set
6. Re-optimisation of predicted structures lying within 30 kJ/mol of the global minimum on the forcefield landscape was performed

Re-optimisation was, in turn, also a multi-stage process:

1. All structures within a 30 kJ/mol window of the forcefield global minimum were re-optimised using machine-learned forcefield MACE-OFF [99]
2. The re-optimised landscape was re-ranked using pDFT single-point energy calculations
3. All MACE-OFF level re-optimised structures within 15 kJ/mol of the pDFT global minimum energy were further re-optimised using pDFT.

### MACE Re-optimisation

To replace DFTB+ re-optimisation, re-optimisation using the MACE energy calculator [98] was implemented in ASE, employing the pre-trained MACE-OFF forcefield [99] to predict the energies and forces. All structures within a 30 kJ/mol window of the forcefield global minimum were re-optimised.

The optimisations were performed allowing simultaneous optimisation of cell parameters and atomic positions. The condition of this simultaneous optimisation was enforced via application of ASE functionality *FrechetCellFilter*. This filter was chosen over commonly used filter *ExpCellFilter* due to encountered inconsistencies in results from the latter. The potential of these inconsistencies was also noted in the documentation of *FrechetCellFilter* in the development version of the code at the time of work. As of an ASE release version (3.23)[159], released after this work was performed, *ExpCellFilter* is deprecated.

Lastly, duplicates were removed from the landscape using COMPACK clustering. The global minimum of the resulting set was unexpectedly low in energy. This structure also displayed an unusual

lattice energy after DFT single-point energy calculation and so was determined to be an anomaly - likely resulting from optimisation errors- and was removed from the set. The presence of the anomalous structure had not previously led to removal of any 'duplicate' structures.

### **pDFT Single-Point Energy Calculations**

The accuracy of lattice energies predicted using MACE-OFF was unknown, and could not be assumed to be reliable, as MACE-OFF used only a molecular training set, the SPICE training set[160], composed of drug-like molecules, and therefore likely dominated by molecular geometries dissimilar to that of CL-20.

Therefore, to establish a subset of low-energy structures to be further re-optimised using pDFT, a reliable energy-ranking was calculated using pDFT single-point energy calculations. These were implemented in VASP using a plane-wave basis set with a cut-off of 500 eV. All single-point calculations were converged until at least reaching minimum convergence criteria of  $1 \times 10^{-5}$  eV. Though calculations on some structures were run to higher convergence criteria of up to  $1 \times 10^{-7}$  eV - again because the computational cost of the latter proved infeasible for some structures. Adjustments were also made to increase the required precision of specified positions in the POSCAR file (*SYMPREC*) in some cases. These cases were adjusted at the recommendation of the VASP software - printed to the initial output files.

### **pDFT Re-optimisation**

On exploration of the MACE+pDFT single-point landscape, it was determined that a suitably extensive and affordable energy window for pDFT re-optimisation would be to re-optimize all structures falling within 15 kJ/mol of the global minimum. This set contained 310 crystal structures - comfortably including all good experimental matches found on that landscape.

Attempting to simplify the process used for re-optimisation of primidone structures, in light of the expected higher 'quality' of the input structures, re-optimisation here was performed in only a single step. Reoptimisations were implemented in VASP, with a basis set cut off of 500 eV, to a minimum convergence of energies of  $1 \times 10^{-5}$  eV. A small number of structures which failed

to converge were rejected. Lastly, duplicates were removed from the resulting landscape using COMPACK clustering.

### Restoring Symmetry

Both MACE and pDFT re-optimisation result in  $P1$  structures - the crystal structures being defined via unit cells with no internal symmetry. Thus, the structures would be defined as having  $Z' > 1$ , indeed,  $Z' \gg 1$  in many cases.

In order to regain structure sets of workable  $Z'$ , symmetry restoration was performed. For the most part, this process was directly equivalent to that used in the case of primidone (Section 4.3.3). However, one step differed. The RMSD criterion used to exclude failed cases of recovered symmetry at MACE level was instead  $\text{RMSD}_{15} > 0.05 \text{ \AA}$ . Whilst this is acknowledged as an inconsistency, both criteria are likely sufficient to exclude failed cases. Additionally, very few structures were excluded at this stage, and so it is unlikely for over-zealous filtering to have significantly impacted results.

### Results

After restoration of symmetry, the landscapes at each level of theory (FIT+DMA, MACE-OFF, pDFT) were searched for matches to the known pure experimental structures ( $\beta$  (PUBMUU01),  $\epsilon$  (PUBMUU02),  $\gamma$  (PUBMUU)[158], and the high-pressure polymorph  $\zeta$  (PUBMUU23 [161])). Searches were performed with tolerances of  $0.3 \text{ \AA}$  and  $0.3^\circ$ . The lowest-energy identified matches from these searches are indicated in Table 4.5.

Calc Level	No. Unique Structures	Polymorph	Energetic Ranking of Match	Relative Lattice Energy (kJ/mol)	RMSD <sub>30</sub> (Å)
FIT+DMA	2123	$\beta$	136	16.596	0.307
		$\gamma$	317	20.183	0.677
		$\epsilon$	22	8.827	0.242
		$\zeta^*$	675	23.948	0.668
MACE-OFF	1364	$\beta$	727	52.22	0.266
		$\gamma$	193	37.897	0.596
		$\epsilon$	105	32.030	0.403
		$\zeta^*$	1061	127.923	1.281
pDFT	229	$\beta$	7	2.815	0.099
		$\gamma$	6	2.789	0.607
		$\epsilon$	2	1.208	0.249
		$\zeta$	N/A	N/A	N/A

Table 4.5: Structure sets sizes and experimental match results from each landscape calculated in crystal structure prediction of CL-20. Entries marked \* required looser ( $0.4\text{\AA}/40^\circ$ ) tolerances to recover

These results are testament to the importance of re-optimisation methods, with the ranking of matches changing significantly, becoming reasonable at pDFT level. Again, as in the case of primidone, it should be noted that the structures corresponding to the low-energy matches at each level of theory differ. Unfortunately, no known structures are found to be the global minimum on the landscape even at pDFT level. Further, several of the identified matches are of poorer quality than would be desired, with high RMSD<sub>30</sub> values, and the high-pressure polymorph could not be recovered within tolerances of  $0.3\text{\AA}$  and  $0.3^\circ$  on any landscape. Matches to the  $\zeta$  polymorph could be identified on the FIT+DMA and MACE-OFF landscapes using looser  $0.4\text{\AA}/0.4^\circ$  tolerances, though again with poor overlays. No match to the  $\zeta$  polymorph could be found on the pDFT landscape, even when using  $0.4\text{\AA}/0.4^\circ$  tolerances. However, as the MACE-OFF landscape  $\zeta$  match lay outside of the feasible energy window for pDFT reoptimisations (after pDFT single points), this is not unexpected. These limitations of the results may simply be due to the complexity of the system, particularly its unusual molecular geometry. The limitations of energetic rankings predicted via MACE-OFF for systems of this kind are especially highlighted.

## 4.6 Duplicate Removal Investigations

### 4.6.1 Overview

One necessary question in crystal structure prediction is **"When are two structures the same?"**. This was a question faced in the most recent CSP blind-test [17, 18] and can be addressed in many ways, including the earlier discussed pXRD and geometric overlay comparisons, and even recent fast methods that rely upon comparison of structural invariants [162]. Particularly pertinent to this thesis though, was the resolution of this question with respect to identification of matches for duplicate removal using comparison of pXRD patterns.

As discussed in Section 2.2.15 a common first step to determine matching crystal structures is pXRD matching. This will determine two predicted structures to be duplicates of one another if their predicted pXRD patterns are suitably similar - based on a cDTW approach. This step is designed to remove 'obvious' duplicates to reduce the size of CSP landscapes such that more effective but costly duplicate removal methods become feasible.

However, when this workflow was implemented for an initial attempt at CSP of primidone, pXRD clustering using the in-house method with default parameters on the DFTB+ landscape was insufficient, leaving a landscape unfeasibly large for workflow progression. This prompted investigation into more suitable parameters.

### 4.6.2 Approach

In the in-house approach to pXRD matching, there are two key parameters:

1. A cosine similarity cut-off beneath which two structures will be assumed to differ and will not be compared
2. A cDTW distance cut-off below which compared pairs of structures will be considered to be duplicates

An optimal pXRD clustering approach should remove as many duplicates as possible whilst stopping or minimising fallacious removal of unique structures. To identify such an approach, the in-house pXRD clustering approach, using different parameter sets, was benchmarked against the more costly but effective COMPACK clustering. Here, COMPACK clustering was treated as 'correct' clustering and pXRD clustering parameters were established to replicate this as far as possible without excessive cost.

To do this, for a sample structure set (all predicted spacegroup  $P12_1/c1$  structures of primidone within a 15 kJ/mol window of the global minimum from a separate, initial CSP search), default COMPACK clustering (0.2 Å/0.2° thresholds) was performed. Then, for each performed COMPACK comparison, the pXRD similarity of the corresponding structure pair was measured analogously to the measurements performed during pXRD clustering. The cosine similarity and cDTW distance was recorded - alongside whether that structure pair corresponded to a COMPACK match or a COMPACK unique comparison.

### Testing Effectiveness

To test the effectiveness of each potential clustering threshold, the subset of all comparisons that corresponded to ‘good’ COMPACK matches (with an  $\text{RMSD}_{30} \leq 0.5$  Å) was identified. These comparisons correspond to structure pairs that can confidently be said to represent duplicate structures - i.e comparisons that would lead to removals in ideal clustering. Then, for each investigated parameter set, the proportion of those good COMPACK matches that would have been identified as duplicates during pXRD clustering was recorded.

### Identifying Risk

To test the ‘risk’ involved in pXRD clustering, for each set of thresholds the subset of comparisons meeting those thresholds was taken. This subset represents the set of comparisons which, if performed during pXRD clustering, would lead to removal of a structure from the pair. Then, the proportion of these comparisons that correspond to COMPACK matches vs the proportion that correspond to COMPACK unique comparisons was recorded.

The effectiveness and risk was then tested for the following combinations of possible pXRD clustering threshold parameters:

cDTW Distance cut-off (°)	Cosine Cut-Off		
10	0.8	0.6	0.4
15	0.8	0.6	0.4

Table 4.6: Possible pXRD duplicate removal thresholds tested

### 4.6.3 Results

Figure 4.15 shows the proportion of good COMPACK matches that would have met the thresholds for identification as duplicates using pXRD clustering with different parameter sets. Here, ‘clustered matches’ are the goal. That is, in essence, a larger blue section of the pie chart represents more effective pXRD clustering.



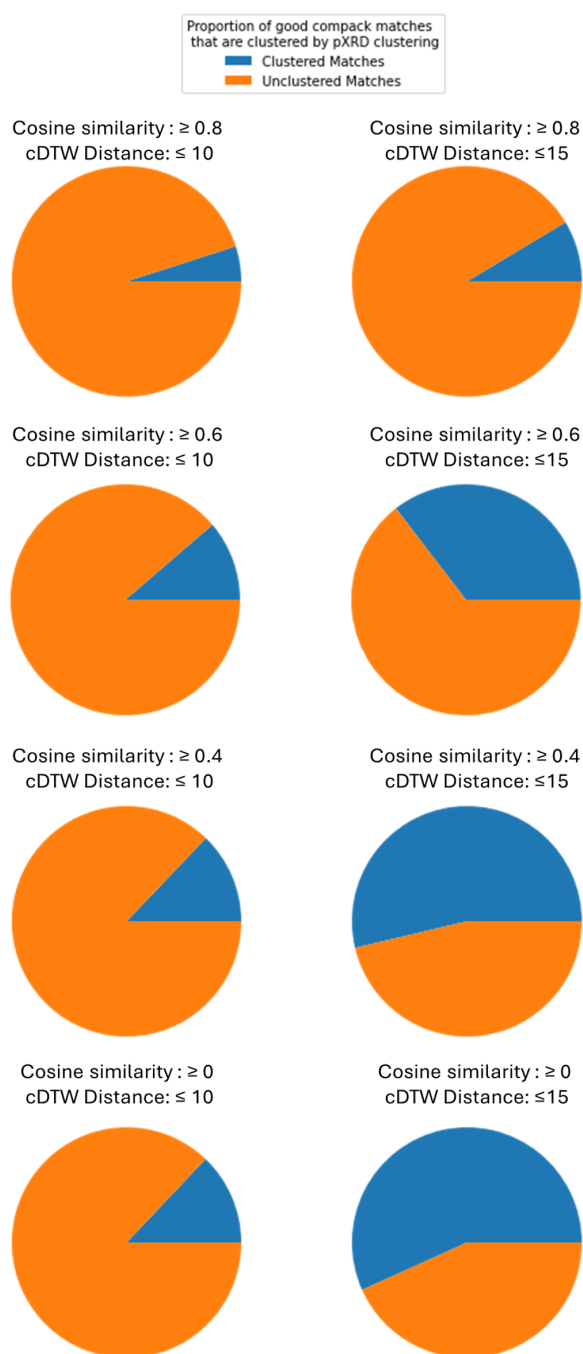


Figure 4.15: Pie charts showing the effectiveness of pXRD clustering using different parameter sets

These results indicate that the most influential factor for the effectiveness of pXRD clustering is the cDTW distance threshold. The cosine similarity cut-off also plays a significant role, however, with a notable increase in effectiveness as the cut-off is lowered from 0.8 to 0.4, though with little improvement beyond that point. Further of note is that even with the looser thresholds pXRD clustering is shown to be significantly less effective than COMPACK clustering.

Figure 4.15 shows the proportion of structure pairs identified as duplicates using pXRD clustering with different parameter sets that corresponded to COMPACK matches. Here, minimising the number of unique structures that could be removed during pXRD clustering is the goal. That is, in essence, a larger blue section of the pie chart represents safer pXRD clustering.

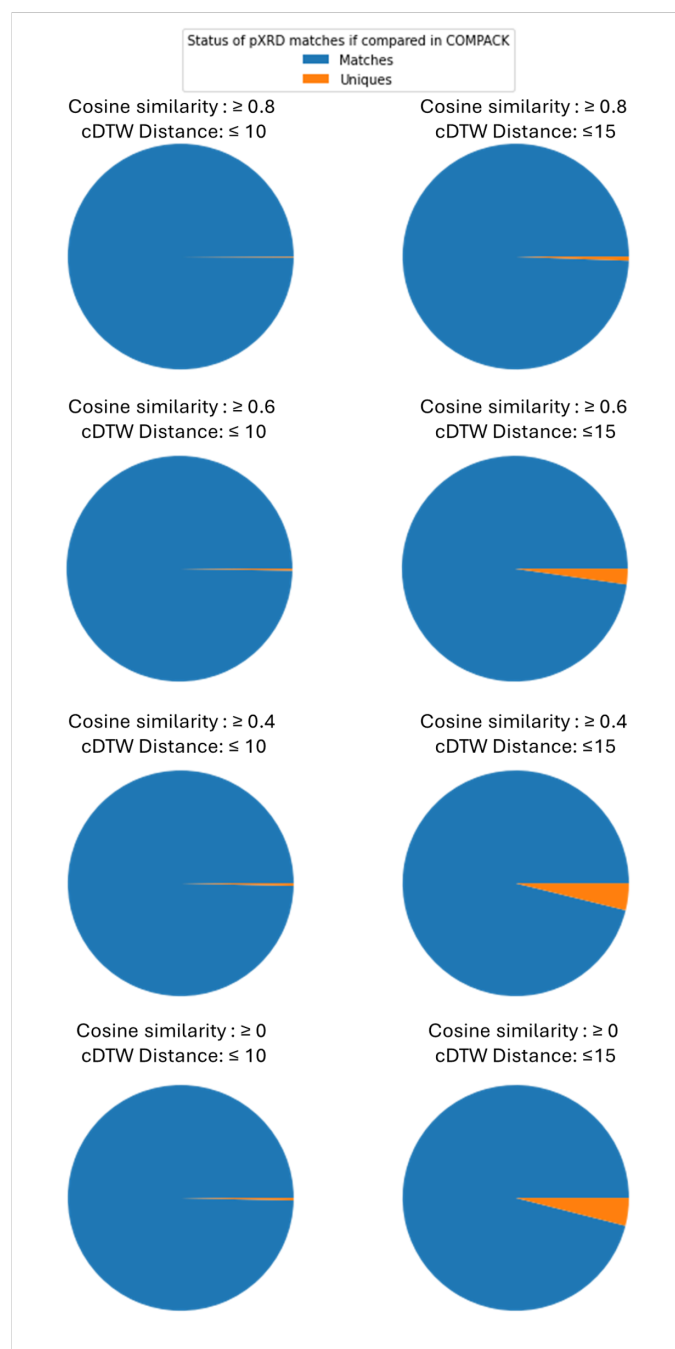


Figure 4.16: Pie charts showing the risk of fallacious removals of using pXRD clustering with different parameter sets

These results suggest that, while of significant benefit to effectiveness, increasing the cDTW distance threshold even from  $10^\circ$  to  $15^\circ$  would introduce greater risk of removing unique structures. Meanwhile, decreasing the cosine similarity cut-off has little impact on risk, and so can be safely exploited to increase effectiveness.

Considering the investigation of the risk and effectiveness of the parameter sets together, it was determined that ideal clustering thresholds, at least at DFTB+ level of theory, are a cosine similarity cut-off of 0.4 and a cDTW distance threshold of  $10^\circ$ . Further decreasing the cosine similarity cut-off would have little impact on either effectiveness or risk, so could be considered equally valid, however, this would needlessly increase the cost of clustering.

## 4.7 Concluding Remarks

Work in this chapter explored crystal structure prediction for a range of small organic molecules, including rigid semiconductor-like molecules, and more flexible molecules with known applications in medicine and defence. This produced several datasets for landscape analysis work later in this thesis. The final CSP landscapes at the highest level of theory in each case were generally, though not universally, successful in recovering known characterised polymorphs. However, the results highlighted the importance of re-optimisation of structures at high-levels of theory.

Several issues were encountered, and addressed. Most importantly, limitations in the applicability of DFTB+ re-optimisation of structures (using the 3OB parameter set), and the poor energetic ranking of structure using the MACE-OFF forcefield were identified. It is unclear how common these issues will be, but it should be borne in mind as a potential risk when applying such workflows in future.

Work also identified optimal parameters to be used in cDTW-based pXRD clustering of structures predicted at the DFTB+ level of theory as a cosine similarity cut-off of 0.4 and a cDTW distance of 10°.

Future work could explore further attempts to predict the structures of the uncharacterised polymorphs of primidone, particularly using Monte Carlo simulated annealing, or investigate the cause of inadequate energetic rankings or poor-quality matches to experimental crystal structures encountered.

## Chapter 5

# Comparing Landscape Analysis Methods

### 5.1 Overview

Initial CSP datasets having been produced and additional landscapes having been obtained from the literature, this chapter discusses work exploring the effectiveness of different landscape analysis methods in identifying the most likely synthesisable structures from the prediction sets. The main focus of the work was in comparison of the Generalised Convex Hull method using the average kernel and the adapted kernel - to assess the impact of the adaptations outlined in Chapter 3. Both implementations of the GCH, however, are also compared to identification of stable structures based solely on relative lattice energy (here called the energy cut-off method) and based upon more conventional convex hull based approaches - namely the energy-density convex hull.

The assessments made call upon systems with known experimental crystal structures, treating the identification of these structures as synthesisable as the crucial goal, and explore how well each landscape analysis method performs in doing so.

Some consideration is also given to the parameters of the GCH method, exploring the impact of the dimensionality of the hull construction and the cut-off radii of the SOAP descriptors.

## 5.2 The Rationale of Comparison

### 5.2.1 The Need for Method Selection

The initial point to be made in discussion of comparison of landscape analysis methods is that of justifying the investigation itself. That is, proving that comparison is worthwhile. Comparison is needed to identify differences in the selection of candidate structures between methods - and any indication of a superior approach - such that researchers can be aware of the dependence of their selections upon their chosen methods and choose approaches wisely. Whilst all methods explored in this work for identifying candidate structures - i.e the most likely synthesisable structures - consider only thermodynamic stability information, the methods utilise this information differently in candidate selection and so may not select the same candidates.

If all methods were to return the same selected candidates, it would suffice simply to select candidates via the least computationally expensive method. However, the selected candidates are likely to vary between approaches - thereby impacting the materials discovery workflow. This may be intuitive. However, it need not be taken as read because the demonstration is simple. Figure 5.1 shows the overlap of synthesisable candidate selections across various landscape analysis methods for a single system - ROY - using structures from the literature [55]. Each section of the Venn diagram shows the number of structures selected as synthesisable only by the corresponding method or set of methods. That is, it shows the size of the candidate pools (Section 2.3.8) for each approach - based upon the number of structures in the pools gathered to capture the known polymorphs of the system.

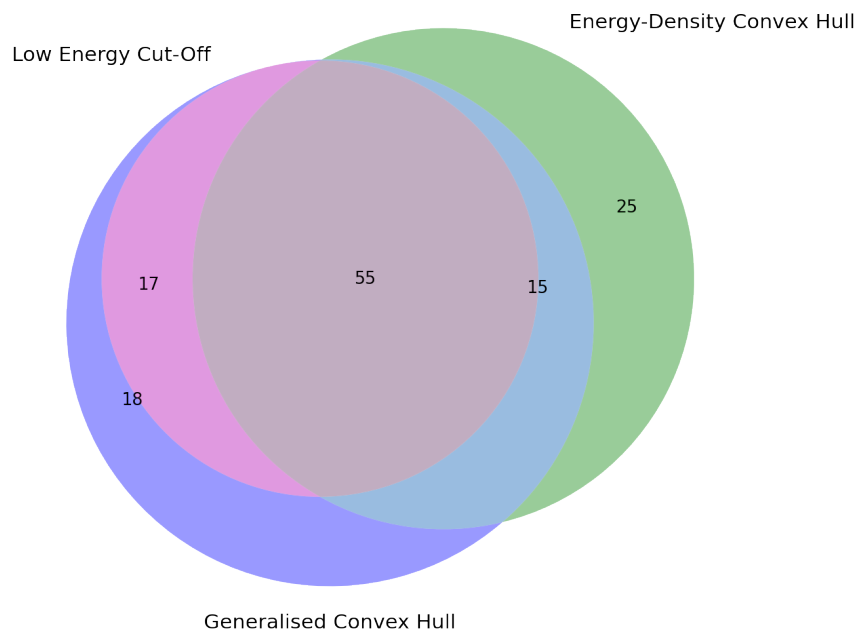


Figure 5.1: Venn diagram showing the number of structures selected as synthesisable only by the corresponding method or set of methods - indicated by the number within each section. The area of each section scales with the number of structures represented. The generalised convex hull method used in this instance is a single case, with a 4Å SOAP cut-off, an adapted SOAP kernel, and a 1D hull.

Whilst there is a significant overlap of selected candidate pools - with 55 structures being selected as synthesisable by all methods - there are also notable differences. This means that it is important to be mindful when selecting an approach -as this will influence the structures suggested as synthesisable. This instance particularly highlights the importance of comparing energy-density and generalised convex hull based approaches as there are structures uniquely identified as synthesisable by each approach, therefore it is not sufficient simply to pick the most comprehensive of the methods.

### 5.2.2 Metrics and Comparisons

To assess the effectiveness of different landscape analysis methods, it was important to select fair means of comparison.

Recall that the ‘dressed energy’ of a structure is given by its height above the convex hull on the relevant landscape. This provides an alternative energetic ranking of structures. Structures with lower dressed energies -i.e closer to the hull - are declared to be more likely stabilisable (and so



more likely to be synthesisable) than those of higher dressed energy. This measure is analogous to that of the relative energy -i.e energy or ‘height’ above the global minimum. However, these measures of energy themselves cannot be directly compared. The dressed energy of a structure must be equal to or lower than its relative energy. This is because the facet of the hull, from which the height (dressed energy) to a given structure is measured, cannot lie below the global minimum (Figure 5.2).

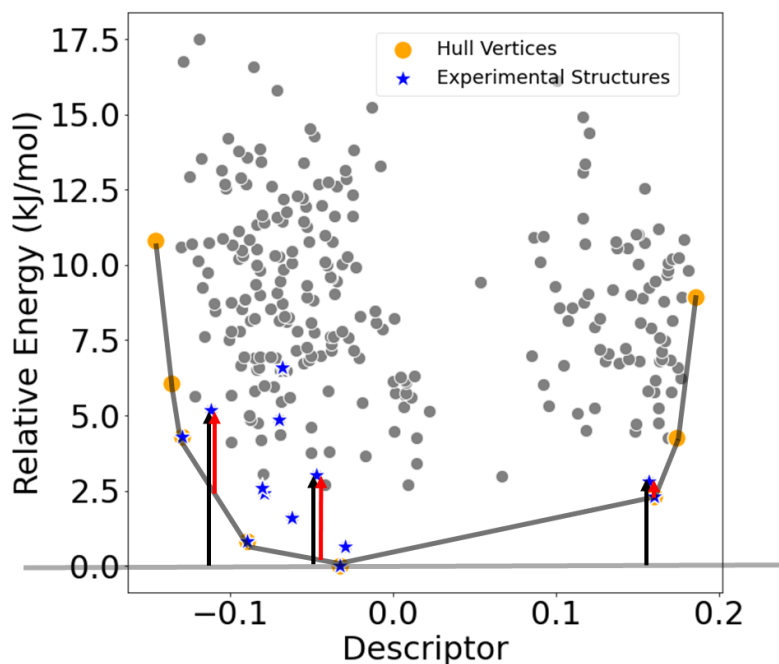


Figure 5.2: Plot of an example of a GCH landscape (here of predicted ROY structures). The determined convex hull is marked by the grey lines (facets) joining the orange vertices. The grey horizontal bar indicates the global minimum height. Black arrows represent the measure of the relative energy of some example experimental polymorphs. Red arrows indicate the corresponding dressed energies - which are lower in all cases.

Therefore, a structure having a lower dressed energy than relative energy does not necessarily correspond to a greater predicted stability. Rather, for a predicted structure present a greater stability as assessed by one analysis method over another, its energetic **ranking** must decrease.

Recall also that, for testing purposes, measure of how successfully approaches identify synthesisable structures must rely upon known data. Three metrics were therefore chosen, that explore the efficiency of approaches in identifying the known polymorphs of a system as synthesisable, based upon their ‘energetic rankings’ according to each method:

1. The candidate pool - the number of structures in the energy/dressed energy window needed to capture all known polymorphs
2. The average polymorph ranking
3. The maximum polymorph ranking

In this context the ranking of a ‘polymorph’ is determined by the ranking of the lowest-energy predicted structure that was found to match to the corresponding experimental structure. Recall that the rankings begin at 1, rather than zero. That is, that for the energy cut-off method, the lowest energy structure is of rank 1 and for the hull-based approaches the lowest-dressed energy structures (i.e the hull vertex structures) all share the rank of 1.

The distinction between the maximum polymorph ranking and the candidate pool is important when considering dressed energy rankings. Whilst the candidate pool for the energy cut-off method - i.e determined by relative energy - ( $Pool^{Rel}$ ) is given simply by the maximum polymorph ranking, the candidate pool as determined by dressed energy ( $Pool^{Dress}$ ) must account for the fact that there may be multiple structures of rank 1 - the set of hull vertices. The candidate pools were determined as indicated in Equations 5.1 and 5.2.

$$Pool^{Rel} = \max(Rank_i^{Rel} | i \in \text{polymorphs}) \quad (5.1)$$

$$Pool^{Dress} = \max(Rank_i^{Dress} | i \in \text{polymorphs}) + |\text{hull vertices}| - 1 \quad (5.2)$$

In order to limit investigation to a feasible number of systems, comparison of landscape analysis methods was restricted to the most interesting systems for which there were multiple known polymorphs successfully predicted within the CSP set. These systems are shown below (Table 5.1).

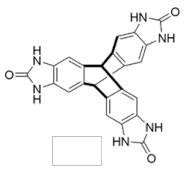
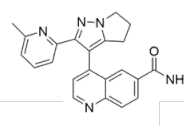
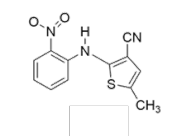
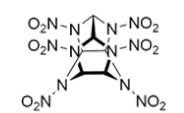
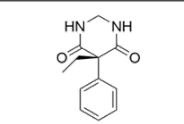
System	Diagram	Number of Pure Polymorphs
T2		5
Galunisertib		10 (9 in set)
ROY		13
CL-20		4 (3 in set)
Primidone		4/2 characterised (2 in set)

Table 5.1: Molecular structures and numbers of known polymorphs for the five systems explored in this chapter.

The T2 [12], galunisertib [15], and ROY [55] structure sets were obtained from the literature - which should be consulted for a full discussion of the prediction processes. In summary, the T2 landscape was obtained via quasi-random sampling and geometry optimisation using FIT + DMA. The galunisertib landscape was obtained via the GRACE CSP software [163] and finally optimised using periodic DFT - the lowest 10 kJ/mol of that landscape being the available set used here. The final ROY literature set used here, arises from a low-energy subset of a full landscape, re-optimised using a bespoke approach.

The primary CL-20 and primidone sets used here are the landscapes at the highest level of theory obtained (pDFT). A discussion of these sets is given in Chapter 4. As discussed, matches to the high-pressure zeta polymorph of CL-20 were not identified at this level of theory, and as such are not included in the determined polymorph rankings and candidate pools. Therefore, the results

found here could be altered had this polymorph been recovered. However, the investigations could be viewed as a reasonable exploration of the ambient pressure landscape. The galunisertib set similarly misses a known polymorph [15].

The datasets of galunisertib and ROY, and to a lesser extent of CL-20 and primidone, represent only a low-energy subset of the landscapes. This prior biasing of the landscape to low-energy structures, and the associated reduction in landscape size, could impact the results. The calculated polymorph rankings, and candidate pools could differ if calculated from a complete landscape - and this should be understood when interpreting the findings.

However, this is not an unreasonable implementation of the landscape analysis. Due to the cost of geometry optimisations, it is common in the field for the available CSP landscape to consist of a low-energy window of what would be the full landscape - re-optimised at a high-level of theory, as in several of the example sets here. The application of landscape analysis methods for identifying synthesisable structures to a low-energy landscape can be viewed as further narrowing the list of candidate structures, or providing a ranking of structures within the set.

### 5.3 Energy Cut-offs and Energy-Density Hulls

#### 5.3.1 Overview

It is sensible to begin discussion of results by returning to the need for alternatives to a simple energy cut-off in selecting candidate structures. This section compares the effectiveness of identifying synthesisable structures based solely on relative lattice energy to using their dressed energy relative to a convex hull of the **energy-density** landscape.

#### 5.3.2 Results

For each investigated system, the candidate pools, average polymorph rankings, and maximum polymorph rankings resulting from each analysis method are shown in Table 5.2.

System	Candidate Pool (Rel Eng   Hull)		Average Polymorph Ranking (Rel Eng   Hull)		Maximum Polymorph Ranking (Rel Eng   Hull)	
T2	647	48	205.8	15.4	647	44
Galunisertib	543	338	120.6	59.222	543	334
ROY	72	96	16.5	19.7	72	89
CL-20	7	13	5	3.7	7	7
Primidone	9	25	7.5	11	9	19

Table 5.2: Candidate pools, average polymorph rankings, and maximum polymorph rankings for each landscape - as determined using ranking based on relative lattice energy and based on use of an energy-density convex hull

These results show that which landscape analysis method is more efficient (i.e draws the smallest candidate pool) varies based upon the system. The systems of T2 and Galunisertib benefit significantly from consideration of the hull but otherwise the candidate pools are worsened by this, as so -generally- are the polymorph rankings. The relative performance of the landscape analysis methods is the same in most cases whether judged by candidate pool size, average polymorph rankings, or maximum polymorph rankings. These results indicate that the performance differences of the approaches are not merely due to extreme rankings of any given polymorph, but reflect overall changes in the ability to rank known crystal structures well within their respective sets.

One contrasting case worth noting, as it exemplifies the different nature of ‘rankings’ between methods, is that of CL-20. For CL-20, the maximum polymorph rankings are equal between meth-

ods, and the average polymorph ranking is lower when employing an energy-density hull (3.666 vs 5.000) . However, the candidate pool is larger. This is due to the presence of different hull vertices - i.e multiple structures of rank 1.

This finding that some landscapes benefit from consideration of an energy-density hull more so than others is not unexpected. Figure 5.3 shows the energy-density landscapes of each system, with the positions of the structures matching to experimental structures indicated.

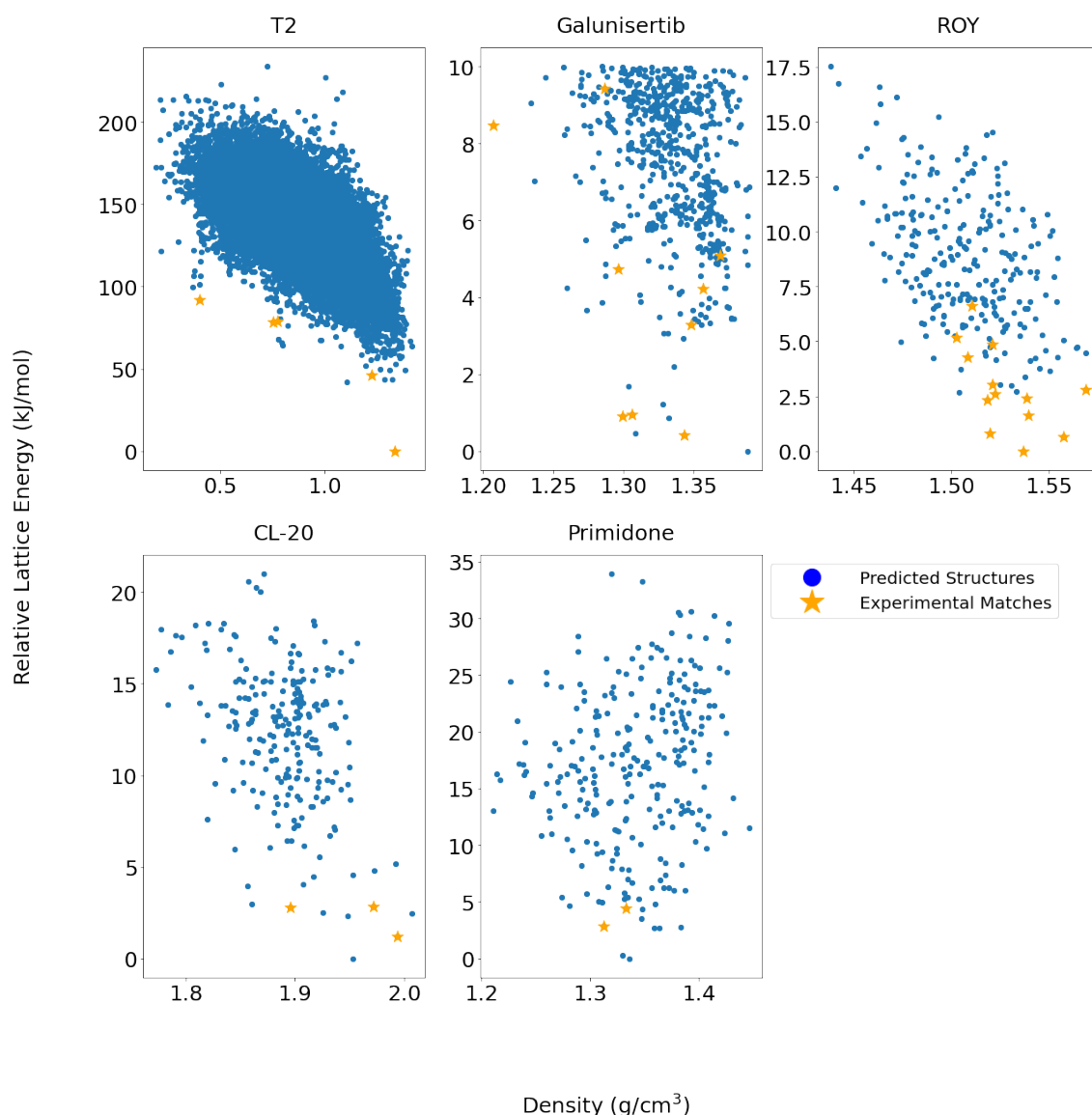


Figure 5.3: Energy density landscapes for each explored CSP set, with the structures matching to the experimental polymorphs indicated by orange stars.

It can be seen that in cases such as ROY, CL-20, and Primidone, the experimental matches lie low in energy - but show little extreme geometric behaviour (i.e do not generally display extreme den-

sities). As such, employing a convex hull based method, which in a sense here captures structures low in energy or extreme in density, is unlikely to be of benefit. For T2 and galunisertib, however, it can be seen from the energy-density plots that there are low-density structures lying higher in energy, and so it could be expected that use of an energy-density convex hull would prove useful. For T2, the high-energy polymorphs are known to be porous [10], and galunisertib has known crystal forms obtained via solvent inclusion [15], making the strong performance of the energy-density hull further expected. In this way, the results in this section do not break new ground, but do reinforce the need for researchers to be mindful of their approach to landscape analysis - i.e it is indeed important to consider whether to use a purely energetic ranking or to consider other factors. This is particularly pertinent in cases where, for example, the porosity of predicted structures may **not** be known or expected. One possible step to determining a landscape analysis approach to use may be to visualise the landscape prior to selecting synthesisable candidates - thus allowing the researcher to check for characteristic features of the landscape that imply porous character, such as ‘spikes’ on the landscape [10]

Lastly, this section has demonstrated that an energy-density hull can be used to greatly improve the ranking of experimental match structures in cases where such structures can be stabilised experimentally by density-related constraints, such as solvent inclusion. This emphasises the need to consider whether there are other instances, albeit with different structural features and related experimental constraints, for which a hull-based method could be used to identify stabilisable structures. This was a primary motivation in exploring the GCH.

## 5.4 Comparing Implemented Kernels

### 5.4.1 Overview

This section opens exploration of the effectiveness of generalised convex hull methods in identifying stabilisable structures by comparing different implementations of the GCH - one implementing an average SOAP kernel, and one implementing the adapted SOAP kernel. ‘Implementing’ a kernel here means that it is that kernel used in kPCA to derive the principal components - i.e the machine learned descriptors - used in hull construction.

It can be expected for the results of each implementation to differ, as the kPCA projections - and so the corresponding GCH landscapes differ. An example of this can be seen in Figure 5.4 which shows the 1D GCH landscapes of the ROY dataset using each implementation. ‘GCH landscape’ is used here to refer to the landscape of predicted structures defined by their energy, and their values for the relevant subset of kPCA components/ML descriptors. It is this landscape upon which the hull used in the GCH workflow is constructed.

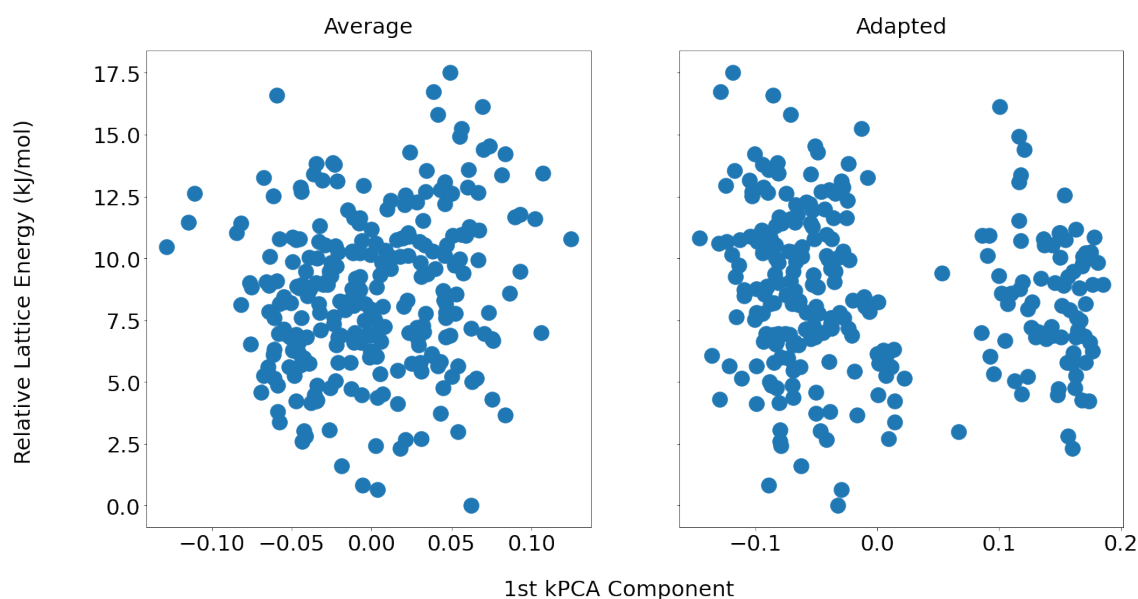


Figure 5.4: GCH landscapes for the set of predicted ROY crystal structures derived using the average and adapted SOAP kernels. Landscapes are constructed upon lattice energy and the top-ranked kPCA component from each kernel implementation

### 5.4.2 Simple Comparisons

To explore the differences between the GCH approaches implementing each kernel, the chosen metrics are first explored here for a single GCH implementation for each kernel - one using a 4 Å



SOAP descriptor cut-off and a 1D hull. It is important to recall here that the defined ‘Hull dimensionalities’ refer to the number of kPCA components/machine learned-descriptors used **alongside** energy in hull construction. That is, a 1D hull is the hull constructed on a landscape of energy and a single (top-ranked) machine-learned descriptor. These comparisons provide a simple starting point for discussion, though the multiple parameters of the GCH make the true picture more complex.

Table 5.3 shows the candidate pool, average polymorph rankings, and maximum polymorph rankings for each system - as assessed using GCH implementations (4 Å cut-off radii, 1D hull) employing the average and adapted SOAP kernels. For purposes of visualisation, the candidate pools in each case are also shown in Figure 5.5.

System	Candidate Pool ( Average   Adapted)		Average Polymorph Ranking (Average   Adapted)		Maximum Polymorph Ranking (Average   Adapted)	
T2	485	296	148.4	81.2	480	290
Galunisertib	525	557	76.667	144.556	521	550
ROY	98	105	18.077	16.769	91	98
CL-20	15	14	4.333	3.667	8	8
Primidone	21	15	10.5	7.5	15	10

Table 5.3: The candidate pool, average polymorph rankings, and maximum polymorph rankings for each systems - as assessed using GCH implementations (4 Å cut-off radii, 1D hull) employing the average and adapted SOAP kernels

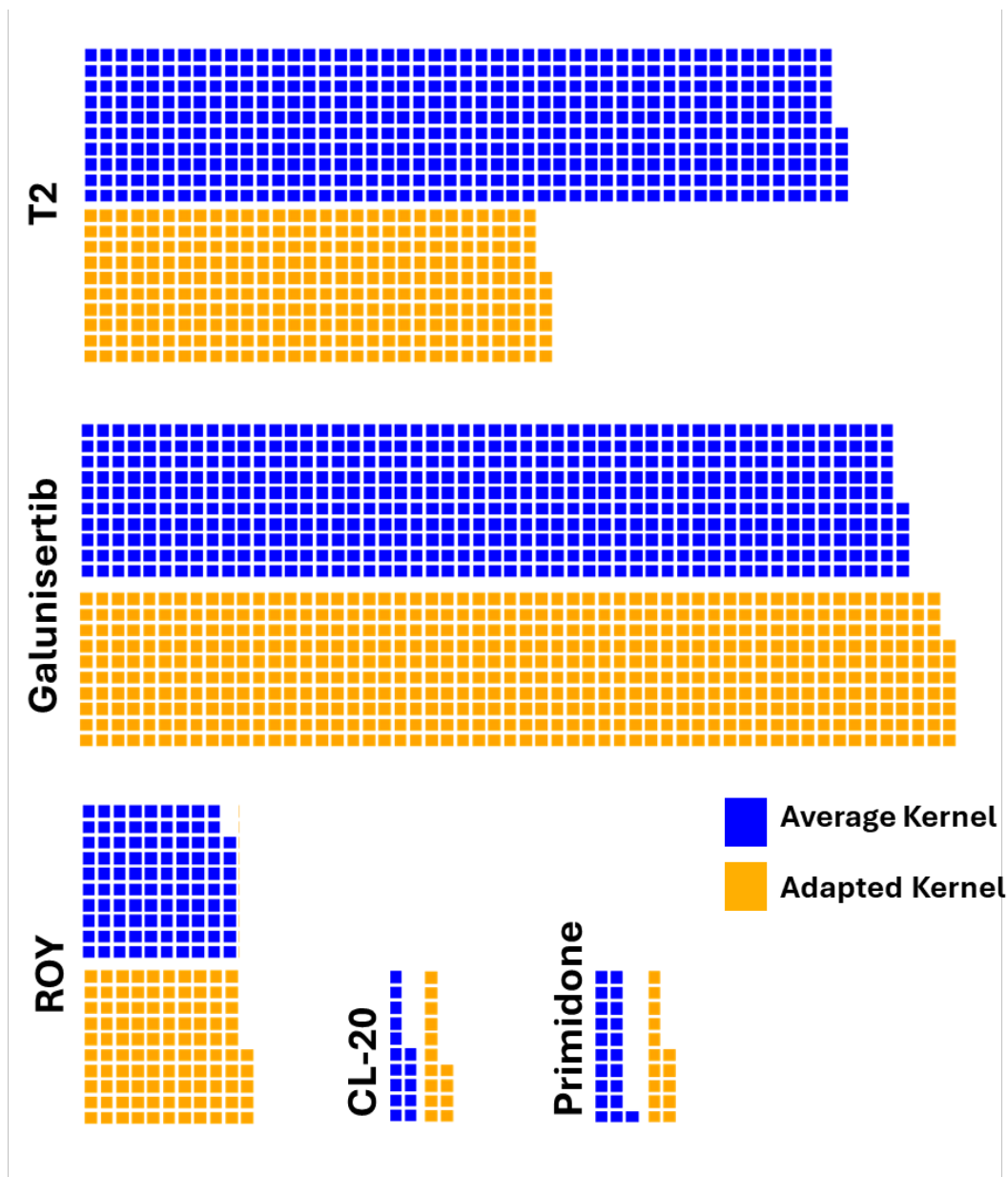


Figure 5.5: The candidate pools for each system - as assessed using GCH implementations (4 Å cut-off radii, 1D hull) employing the average and adapted SOAP kernels. Each block represents a single predicted crystal structure in the candidate pool.

These results, again, display variability depending upon the system explored. There is no clear trend highlighting one or other kernel construction as leading to a more efficient GCH implementation. This ambiguity can be seen for all explored metrics. In most cases, the candidate pools are similar - with neither method presenting a significantly smaller pool. The exception to this is the case of T2, in which the pool is drastically reduced when using the adapted kernel implementation. However, this singular case is not sufficient to prove a pattern.

Already, this simplified exploration indicated that it may be difficult to distinguish a superior option between the adapted and average GCH implementations. This is a negative result, particularly in light of the fact that the adapted kernel was designed to be more theoretically reasonable. The adapted kernel implementation could therefore be expected to improve results, or at least not worsen them. It is not known why this was not found. One possibility is simply that the GCH method itself could have poor performance and applicability to these systems - and so a more reasonable underlying kernel would not meaningfully impact the results of an otherwise inappropriate method. Another possibility is that the discrepancies in performance may simply be ‘noise’ and not represent significant findings. This is explored in Section 5.4.5.

Despite these initial results, it remained necessary to explore the true picture - across the varied GCH parameters of dimensionality and SOAP cut-off, in case trends could be found in this wider investigation.

### 5.4.3 Intrinsic Dimensionality

One sign that the additional parameters of the GCH could impact trends in the relative performance of the different kernel implementations can be seen from exploring the intrinsic dimensionality of the full GCH landscapes in each case.

A measure of the intrinsic dimensionality of the CSP landscape in the SOAP kernel space (here referred to as the **full** GCH landscapes) arises naturally from the formation of the kPCA projection. As discussed in Section 2.4.5 the kPCA algorithm produces eigenspectra for the space that it is reducing. These eigenspectra are used for deriving the projection and ‘ranking’ the resulting components. However, they can also provide a measure of the intrinsic dimensionality of the space - i.e the number of ML descriptors needed to capture meaningful information.

This can be seen by direct exploration of the eigenspectra. Identification of elbows or ‘drop-offs’

in the spectra, beyond which the variance captured by descriptors significantly decreases, can show the number of components containing important data. Alternatively, the cumulative variance captured by a given set of components can be calculated:

```
def get_var(spectrum):  
    total_var = sum(spectrum)  
    ind_var = [i/total_var for i in spectrum]  
    cumulative_var = [sum(ind_var[0:j+1]) for j in range(len(ind_var))]  
    return cumulative_var
```

where *spectrum* is the ordered array of eigenvalues from kPCA.

The cumulative variance of a subset of components shows the percentage of the full dataset variance that will be included in the corresponding reduced dataset. The intrinsic dimensionality is given by the number of components - collated in rank order- required to capture a high percentage of the variance.

Both the eigenspectra and the cumulative variance have been calculated for each kernel for each system in this chapter. These can be seen in Figures 5.6 and 5.7 respectively.

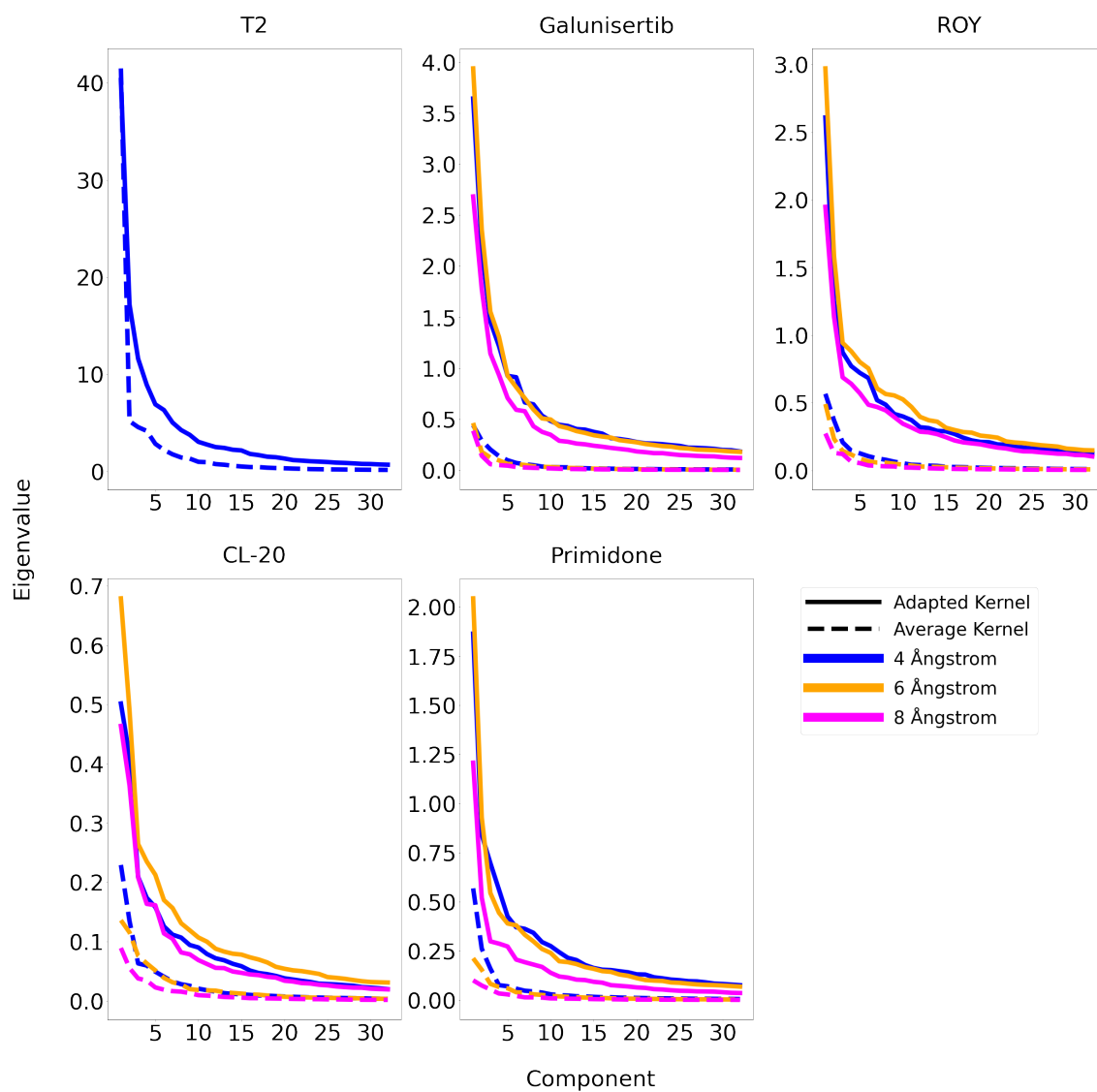


Figure 5.6: Eigenspectra for the kPCA projections for each system explored in this chapter. Spectra are shown over the limited range of the first 32 components of the projection in each case. Different line-styles and colours denote the kernel used to form the kPCA projection.

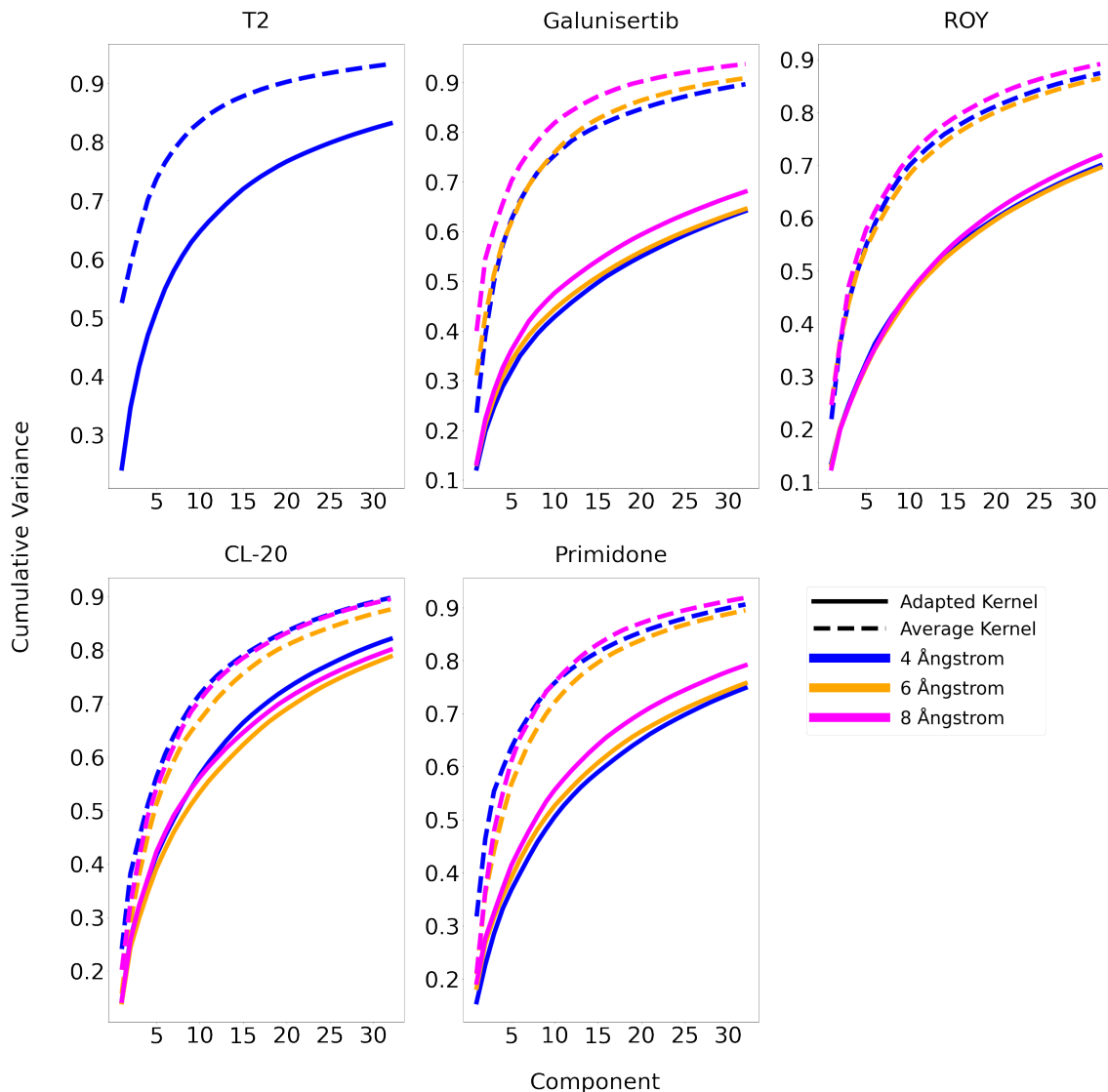


Figure 5.7: Graphs of cumulative variance captured by the first  $x$  components of the kPCA projections for each system explored in this chapter. Graphs are shown over the limited range of the first 32 components of the projection in each case. Different line-styles and colours denote the kernel used to form the kPCA projection.

The plots indicate a higher intrinsic dimensionality of the adapted SOAP kernel space over the average SOAP kernel space - a finding that appears consistent across all systems and SOAP cut-offs. The different eigenspectra/variance curves from the different kernel constructions indicate the impact of hull dimensionality on the results may vary between kernel implementations. Therefore, it is necessary to compare the results of each kernel GCH implementation across different hull dimensionalities.

Unfortunately, the higher intrinsic dimensionality of the full adapted GCH landscapes is a concern

for the approach’s utility. If the required dimensionality of the GCH implementation is higher, this can potentially lead to negative impacts, such as larger candidate pools. Though the impact of hull dimensionality upon the candidate pool is by no means linear, the number of hull vertices increases with increased dimensionality. This therefore increases this contribution to the candidate pool, and leads to a wider collection of structures for which a relative ‘stability ranking’ cannot be obtained. Further, as discussed in Section 2.4, the GCH identifies structures that are stabilisable by some constraint related to the descriptors used to construct the hull. On a high-dimensional hull, some identified structures may only be stabilisable by simultaneous control of several experimental variables - making experimental discovery a trickier process.

However whilst this may negatively impact a materials discovery workflow, it does not necessarily indicate a truly inferior approach to dimensionality reduction when viewed in light of the original dataset. The eigenspectra suggest that the space can be re-defined using fewer kPCA components without data loss from the full SOAP kernel space in the case of the average kernel. However, it could be possible that information about the structure set has already been ‘lost’ on construction of the average SOAP kernel itself.

#### **5.4.4 Polymorph Rankings and Candidate Pools**

Tables 5.4, 5.5, and 5.6 show the candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using 4 Å, 6 Å and 8 Å SOAP cut-offs respectively.

System	Kernel Type	Hull Dimensionality	Candidate Pool	Mean Polymorph Ranking	Maximum Polymorph Ranking
T2	Average	1	485	148.4	480
		2	440	146.2	410
		3	466	147.4	353
		4	826	208.4	512
		5	962	64.0	316
	Adapted	1	296	81.2	290
		2	315	134.0	294
		3	1275	484.0	1166
		4	1195	189.0	879
		5	1763	207.4	1033
Galunisertib	Average	1	525	76.7	521
		2	571	79.0	553
		3	535	77.4	481
		4	571	69.8	473
		5	605	54.8	431
	Adapted	1	557	144.6	550
		2	450	134.7	432
		3	547	84.2	490
		4	352	28.0	244
		5	432	28.6	249
Continued on next page					



Table 5.4 – continued from previous page					
ROY	Average	1	98	18.1	91
		2	76	9.8	52
		3	118	7.8	63
		4	158	6.7	70
		5	184	5.2	55
	Adapted	1	105	16.8	98
		2	147	14.6	113
		3	153	12.5	95
		4	136	3.7	24
		5	173	2.3	11
CL-20	Average	1	15	4.3	8
		2	22	1.3	2
		3	46	1.0	1
		4	77	1.0	1
		5	99	1.0	1
	Adapted	1	14	3.7	8
		2	38	6.0	16
		3	70	5.3	14
		4	106	4.3	11
		5	136	2.7	6
Primidone	Average	1	21	10.5	15
		2	27	3.5	6
		3	56	1.0	1
		4	104	1.0	1
		5	147	1.0	1
	Adapted	1	15	7.5	10
		2	38	11.0	18
		3	77	12.0	22
		4	115	6.0	11
		5	152	3.5	6
Continued on next page					

<b>Table 5.4 – continued from previous page</b>
---

Table 5.4: Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a 4 Å underlying SOAP cut-off radius.

System	Kernel Type	Hull Dimensionality	Candidate Pool	Mean Polymorph Ranking	Maximum Polymorph Ranking
Galunisertib	Average	1	292	63.2	286
		2	350	59.1	334
		3	148	24.4	103
		4	211	25.1	122
		5	274	15.9	95
	Adapted	1	549	136.6	544
		2	286	77.6	263
		3	309	66.6	243
		4	174	6.2	48
		5	213	2.1	11
ROY	Average	1	89	15.8	80
		2	140	18.1	116
		3	177	15.4	125
		4	175	9.2	84
		5	149	2.2	16
	Adapted	1	99	17.4	94
		2	127	17.2	98
		3	117	14.0	58
		4	147	10.9	48
		5	177	4.0	26
Continued on next page					

Table 5.5 – continued from previous page					
CL-20	Average	1	10	3.7	6
		2	24	2.7	5
		3	48	2.3	4
		4	92	1.0	1
		5	126	1.0	1
	Adapted	1	20	5.3	13
		2	46	7.7	21
		3	70	4.3	11
		4	96	2.7	6
		5	137	2.0	4
Primidone	Average	1	25	12.0	19
		2	33	5.5	10
		3	53	1.0	1
		4	97	1.0	1
		5	143	1.0	1
	Adapted	1	16	6.0	8
		2	29	2.0	3
		3	62	1.0	1
		4	94	1.0	1
		5	142	1.0	1

Table 5.5: Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a 6 Å underlying SOAP cut-off radius.

System	Kernel Type	Hull Dimensionality	Candidate Pool	Mean Polymorph Ranking	Maximum Polymorph Ranking
Galunisertib	Average	1	154	56.8	147
		2	208	47.1	186
		3	275	49.2	221
		4	187	18.0	81
		5	195	1.0	1
	Adapted	1	556	137.7	550
		2	284	64.6	257
		3	330	58.4	261
		4	194	13.3	70
		5	194	1.0	1
ROY	Average	1	94	15.1	88
		2	121	16.9	102
		3	178	17.3	123
		4	149	6.9	50
		5	159	2.0	14
	Adapted	1	94	22.2	89
		2	116	19.8	84
		3	138	16.5	81
		4	155	9.8	51
		5	187	7.0	40
Continued on next page					

Table 5.6 – continued from previous page					
CL-20	Average	1	11	3.7	7
		2	21	1.3	2
		3	51	1.0	1
		4	95	1.0	1
		5	143	1.0	1
	Adapted	1	11	3.0	6
		2	40	5.3	14
		3	60	4.7	12
		4	113	5.7	15
		5	143	1.0	1
Primidone	Average	1	23	9.0	17
		2	44	8.5	16
		3	73	10.5	20
		4	97	1.0	1
		5	148	1.0	1
	Adapted	1	13	3.5	5
		2	27	1.0	1
		3	68	1.0	1
		4	113	1.0	1
		5	166	1.0	1

Table 5.6: Candidate pools, mean polymorph rankings, and maximum polymorph rankings determined for each system as a function of kernel type and hull dimensionality for cases using a 8 Å underlying SOAP cut-off radius.

These results show that, even under wider investigation, the kernel adaptations cannot be shown to have a consistent positive or negative impact upon the effectiveness of the GCH in identifying stabilisable structures. It can now be seen that the ‘winning’ implementation at any given point depends not only upon the system explored but also upon the hull dimensionality selected. For example in the case of T2, the adapted kernel implementation initially performs best, but this performance gap quickly reverses at higher dimensionality. The average polymorph ranking and maximum polymorph ranking metrics appear similarly variable - and add little to the conclusions in this respect.

It is worth observing, however, that there are instances such as for the systems of CL-20 and primidone at higher hull dimensionalities, in which all polymorphs are ranked well - or indeed all representing hull vertices - but the candidate pools are large. This is again due to the presence of other structures on the hull. These expanded results highlight the risk of applying the GCH with high dimensional hulls for molecular systems. For example the candidate pool of a GCH implementation (Adapted kernel, 8 Å cut-off) rises from 27 to 166 when increasing the hull dimensionality from 2D to 5D - despite no improvement in the polymorph rankings.

The candidate pools as a function of hull dimensionality are visualised in Figure 5.8. Different curves denote different kernel constructions - with different kernel types and underlying SOAP cut-offs.

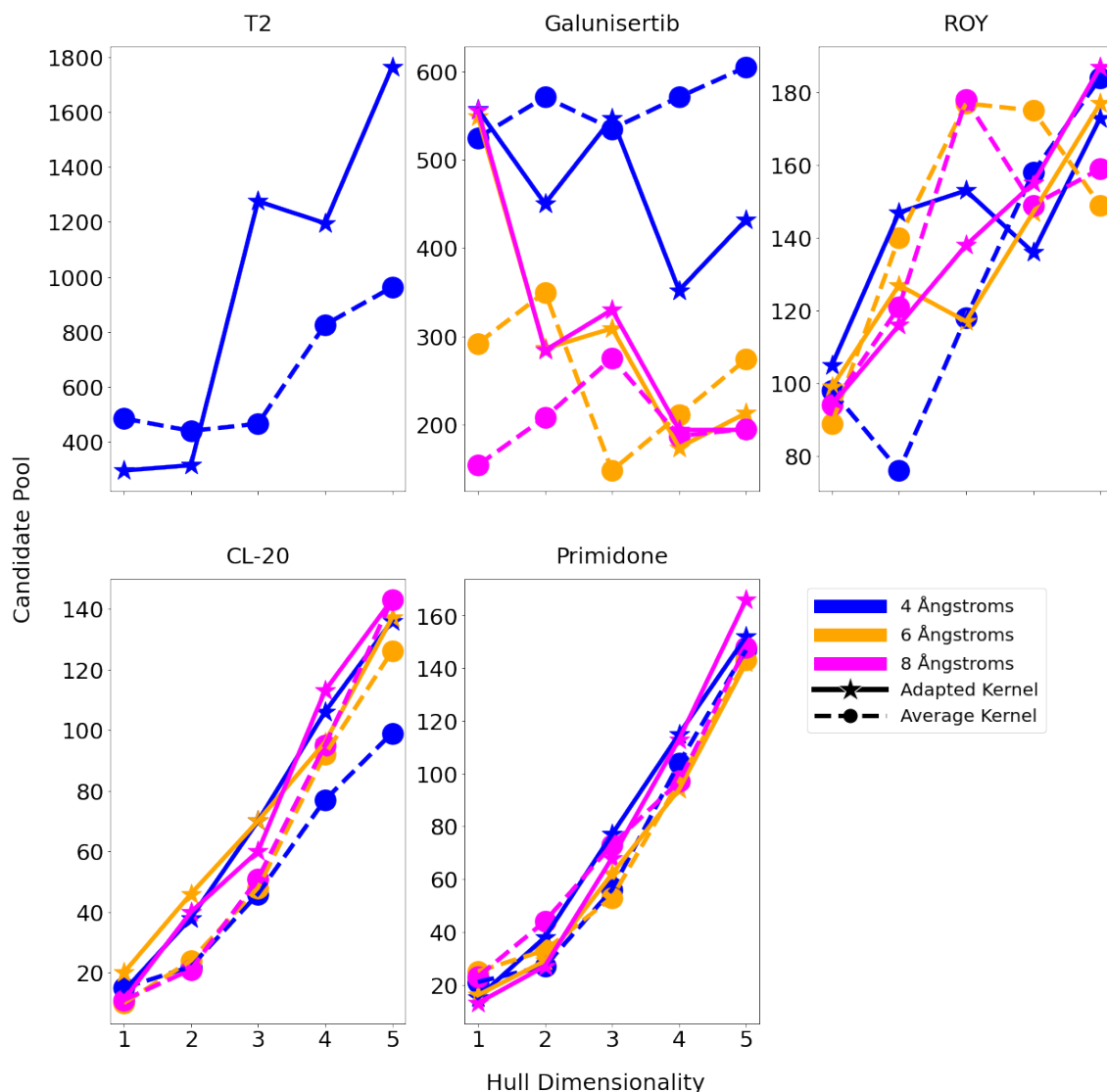


Figure 5.8: Candidate pools for each system as a function of hull dimensionality, SOAP cut-off, and kernel construction for each. Different curves denote different kernel constructions - with different kernel types and underlying SOAP cut-offs.

The figures reiterate the unpredictable nature of the results. Further, they display a finding perhaps harder to discern from the data tables alone, that the results, including the determination of the superior kernel implementation, are also dependent upon SOAP cut-off. The exceptions to this, are the systems of CL-20 and primidone, which also display a consistent increase in candidate pools with increasing hull dimensionality. This is likely due to the dominant factor in these candidate pools being the sheer accumulation of hull vertices. No consistently ‘best’ SOAP cut-off can be found from the data shown.



### 5.4.5 Accounting for Energetic Uncertainty

One question to consider, particularly given the unpredictable nature of the kernel comparison results, is whether or not the differences in each case are significant. As a means of testing this, the impact upon the candidate pools of uncertainty in the underlying calculated energies was assessed. Then, it was tested whether or not the discrepancies in performance between the average and adapted kernel GCH implementations exceeded these margins of error.

The candidate pools discussed thus far were derived via hulls constructed once upon landscapes using exact calculated energies for each crystal structure. Therefore, they were vulnerable to the error and uncertainty in those energy calculations. This was an issue that the original GCH implementation attempted to resolve via probabilistic hull sampling.[1] However, the probabilistic hull sampling protocol [164] had limitations with regard to implementation for this work. The impact of the probabilistic hull sampling upon results was unknown. Further, whilst the hull vertices were sampled probabilistically, The dressed energy calculations still relied using the exact calculated energies, retaining a sensitivity of the dressed energy rankings to energetic uncertainty. Lastly, the approach additionally incorporated structural uncertainty. This perturbed structures using the *rattle* function implemented in ASE [164]. This function applies a random displacement of up to a given size to each atom individually. Such an approach is not appropriate for molecular crystals. In light of these concerns and in order to focus effort on the aspects of the GCH process most relevant to this thesis, the probabilistic hull sampling was removed from the process.

However, the original approach to probabilistic hull sampling did inspire the approach used here to assess the impact of energetic uncertainty. This is done by recalculating the hull and dressed energies multiple times, beginning from different lattice energies within the uncertainty in calculations. This does not improve the calculated energies themselves, but is designed to examine the sensitivity of calculated candidate pools to the exact lattice energies used - which within the uncertainty of calculations are arbitrary. The workflow used here implemented a loop in which:

1. The energies assigned to each structure were randomly perturbed within reasonable windows
2. The hull was reconstructed - using the same descriptor (kPCA component) values for each structure, alongside the perturbed energies
3. The candidate pools were recalculated

Across sufficient iterations, this provides a view of the expected ‘spread’ of calculated candidate pools as the underlying energies are varied within reasonable uncertainty. This spread provides insight into the margin of error in the calculated candidate pools.

‘Reasonable windows’ for energy perturbations are dependent not upon the total average errors for a given energy calculation method, but upon the average **non-systematic** (random) errors for that method. Given an estimation of the random errors for each method, energy perturbations for each structure were then randomly selected from a uniform distribution, centered about 0, with a standard deviation equal to the estimated random error

Given the specificity of the methods applied in the CSP procedures used, clearly benchmarked non-systematic errors were not freely available for all systems. To attempt to gauge sensible estimates of these errors, different approaches were employed.

For the case of T2 [12], the energies were calculated using intermolecular forcefield FIT+DMA. A reliable benchmark [28] of this energy evaluation method was available - assessed on the X23 set of experimentally determined lattice energies [165]. From this data, the non-systematic error ( $Err_{Rand}$ ) was estimated by the difference in magnitude of the Mean Absolute Error (MAE) and the Mean Signed Error (MSE):

$$Err_{Rand} = MAE - |MSE| \quad (5.3)$$

This resulted in an approximate estimate of the non-systematic error of 1.3 kJ/mol.

For the case of ROY, the final energies were evaluated using a bespoke method[55]. The literature reference provided indication of the average error of the method on the ROY dataset - providing a figure of  $\sim 0.4$  kJ/mol Root Mean Squared Error (RMSE) on a test set of known ROY polymorphs. As precise data from which to calculate the non-systematic error was not available, this value was utilised as-is, having been taken to represent a top bound of the non-systematic error - which cannot exceed the 0.4 kJ/mol estimate.

For systems for which energies were calculated using pDFT calculations, the specific implementations (e.g dispersion corrections) varied. For the purposes of testing in this thesis, the random error was assumed to lie between that of FIT+DMA and the bespoke method applied to ROY. A rough estimate of 1.0 kJ/mol was applied.

The workflow and estimates of random error having been decided, before work could proceed, it was necessary to determine the number of iterations necessary to obtain consistent estimate of the spread of calculated candidate pools. A single example case (T2, adapted kernel, 4 Å cut-off, 1D hull) was used to test this. Figure 5.9 shows the spread of calculated candidate pools - represented as box plots, tested across different numbers of iterations of the loop. The workflow using each number of iterations was tested four separate times, to explore the consistency of results.

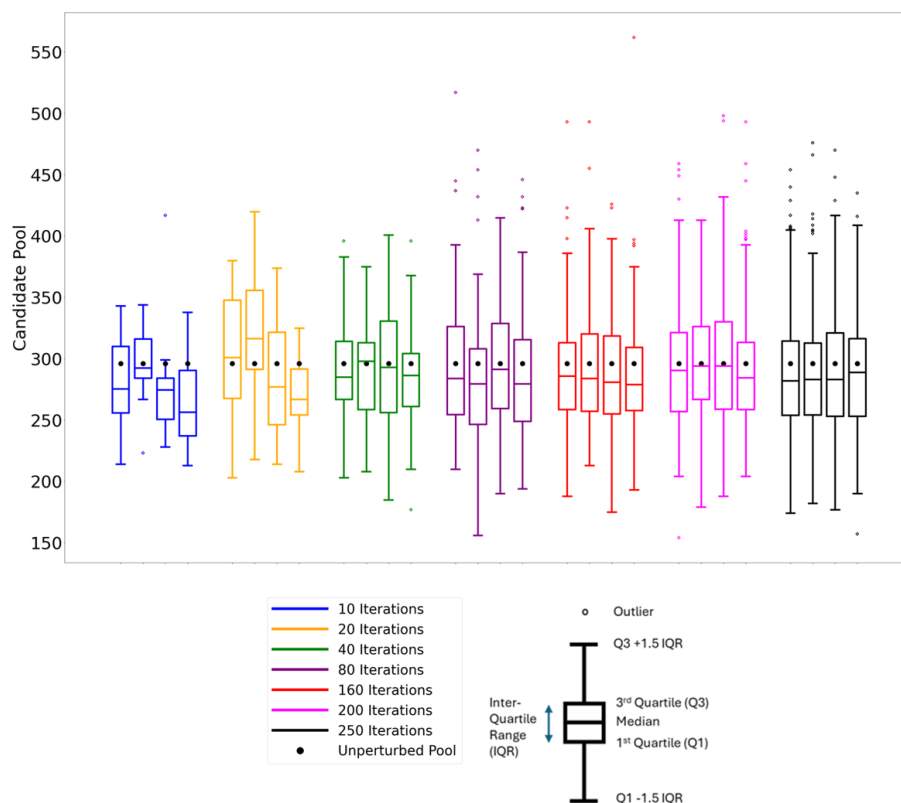


Figure 5.9: Boxplots showing the spread of calculated candidate pools for a single example case (T2, adapted kernel, 4 Å cut-off, 1D hull), when the iterative workflow to account for energetic uncertainty was applied. Each colour signifies a different number of iterations of the loop used and each boxplot of a given colour corresponds to a separate ‘run’ of the workflow with that many iterations.

Based upon the above results, it was determined that 250 iterations of the loop was sufficient to obtain measures of the spread of candidate pools that were not significantly impacted by the arbitrary nature of the specific random perturbations implemented.

Therefore, a workflow was finalised in which the energies of each structure were reasonably perturbed, and the hulls and candidate pools recalculated, 250 times. This workflow was then per-

formed for each explored system, kernel implementation, hull dimensionality, and SOAP cut-off. For simplicity, only the results pertaining to use of a 4 Å SOAP cut-off are shown here (Figure 5.10) - though results corresponding to other cut-off radii showed similar patterns.

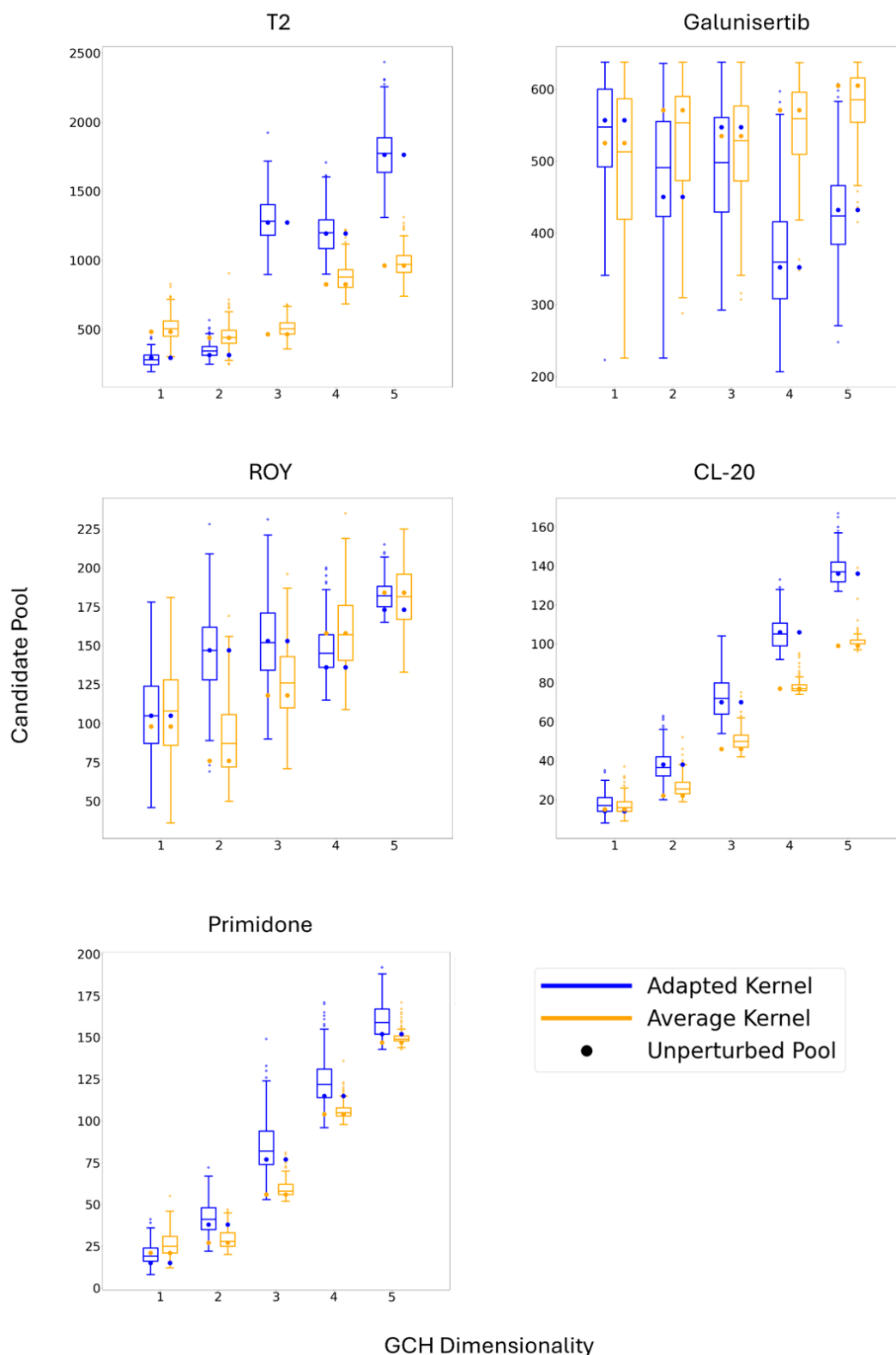


Figure 5.10: Box plots showing the spread of calculated candidate pools arising from the finalised iterative workflow for each system investigated from the average and adapted kernel GCH implementations ( $4 \text{ \AA}$  cut-off) using different hull dimensionalities.

These results show that in most instances, the discrepancies between kernel implementations are not significant. The spread of calculated candidate pools from the average and adapted kernel GCH implementations in these cases overlap significantly. Further the ‘unperturbed pools’ - i.e the values discussed earlier - for one kernel implementation mostly lie within the expected spread of pools from the competing implementation. This cements the conclusion that the impact of the kernel adaptations on the effectiveness of the GCH in identifying stabilisable structures cannot be clearly determined. It is acknowledged that there are cases (T2 and CL-20) for which at high hull dimensionality (3D/5D and 5D respectively), the average kernel GCH implementation may significantly outperform the adapted kernel GCH implementation. However, these cases alone were not considered sufficient to confirm a pattern.

One finding of interest, though, is that in many cases the average kernel implementation appears less vulnerable to energetic uncertainty with lesser spread of calculated candidate pools. The origin of this phenomenon was not known - though could be of interest for investigation in future work.

### 5.4.6 The Impact of Energetic Calculation Methods

The high sensitivity of the candidate pools to energetic uncertainty naturally raises the related question of the sensitivity of results to the energetic method applied - i.e to the level of theory at which predictions were conducted. Due to time constraints, this has not been extensively investigated here. As a preliminary test however, the candidate pools (relying upon exact calculated energies) were also calculated for the primidone and CL-20 GCH landscapes at lower levels of theory. These ‘lower-level’ landscapes are those discussed in Chapter 4. These results are shown in tables 5.7 and 5.8 respectively. For consistency, candidate pools from the CL-20 system were assessed based only on the three polymorphs that were found in predictions at all levels of theory. The  $\zeta$  polymorph was not considered.

		Forcefield		DFTB+		pDFT	
SOAP cut-off (Å)	Hull Dimensionality	Candidate Pool (Average   Adapted)					
4	1	231	115	74	354	21	15
	2	308	256	70	135	27	38
	3	467	198	109	100	56	77
	4	743	318	198	209	104	115
	5	959	408	302	326	147	152
6	1	298	125	50	619	25	16
	2	341	68	119	36	33	29
	3	419	90	104	104	53	62
	4	448	236	206	235	97	94
	5	611	447	267	370	143	142
8	1	390	128	121	622	23	13
	2	186	143	82	73	44	27
	3	200	99	84	105	73	68
	4	280	214	161	213	97	113
	5	398	445	236	379	148	166

Table 5.7: Candidate pools for the primidone system calculated using various GCH implementations acting on the original CSP landscapes at different levels of theory.

		FIT + DMA		MACE-OFF		pDFT	
SOAP Cut-off (Å)	Hull Dimensionality	Candidate Pool (Average   Adapted )					
4	1	366	127	978	661	15	14
	2	390	191	1053	746	22	38
	3	497	357	1153	1109	46	70
	4	707	507	520	1139	77	106
	5	861	693	738	1199	99	136
6	1	511	120	969	420	10	20
	2	706	137	1060	507	24	46
	3	886	307	535	966	48	70
	4	711	447	476	1058	92	96
	5	634	598	735	1057	126	137
8	1	619	144	1282	598	11	11
	2	783	186	1270	793	21	40
	3	1001	261	1314	1103	51	60
	4	1050	356	618	1140	95	113
	5	833	554	632	1073	143	143

Table 5.8: Candidate pools for the CL-20 system calculated using various GCH implementations acting on the original CSP landscapes at different levels of theory

These results indicate that the effectiveness of identification of stabilisable structures is vulnerable to the level of theory used in energetic calculations. There are some cases where this dependency upon the underlying method is particularly extreme. This is expected, as the relative lattice energy rankings of the polymorph matches also varied greatly between levels of theory (Chapter 4). However, the inability of the GCH - regardless of the kernel implemented - to overcome this sensitivity limits the utility of the approach.



A more thorough investigation of the sensitivity of the GCH approach to the energetic calculation method could be to explore the correlation of dressed-energy rankings of structures arising from GCH implementations at different underlying levels of theory. This could be done, for example, using Kendall Rank Correlation [166]. For the examples here, not only the calculated energies of the structures change between landscapes at each level of theory, but the structures themselves also change. Therefore it may require thought to determine the most sensible implementation - with respect to which structures are treated as the same between landscapes. However, this could be of interest for future work.

## 5.5 Comparing Kernel Based Methods to Traditional Approaches

### 5.5.1 Overview

Having compared different traditional approaches to identifying synthesisable structures and then having compared different GCH implementations, it remains to compare the former and latter. It can be expected that identification of synthesisable structures via the GCH will differ from more traditional approaches - even the energy-density convex hull. This is because the landscapes, upon which the hulls are constructed, differ. An example of this, for the GCH and energy-density landscapes of the ROY structure set, is shown in Figure 5.11.

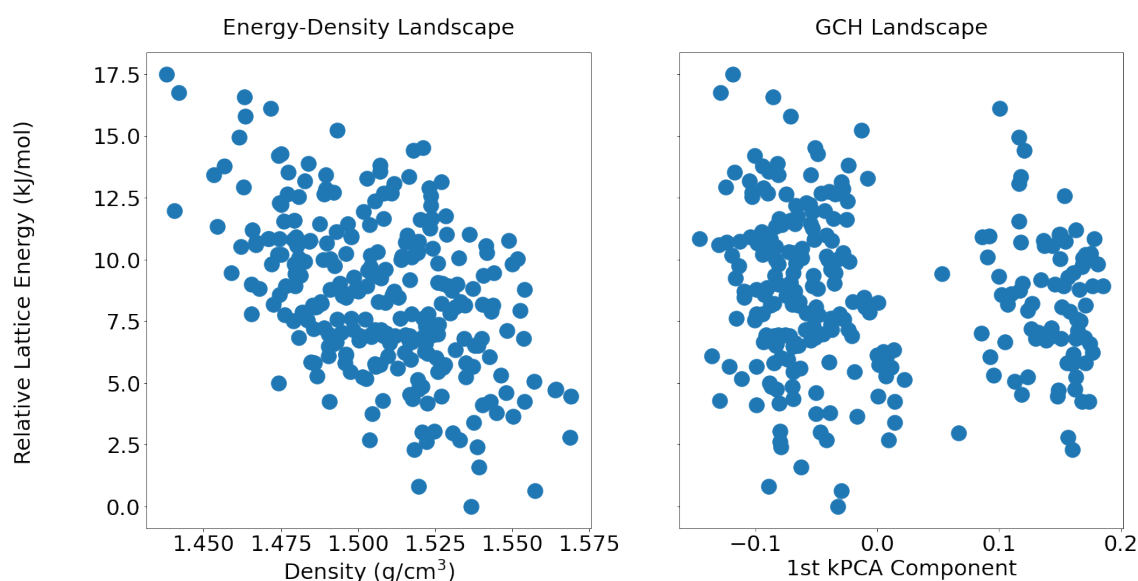


Figure 5.11: Energy density and GCH (Adapted kernel, 4Å cut-off, 1D hull landscapes of the ROY CSP set.

### 5.5.2 Polymorph Rankings and Candidate Pools

The nature of comparisons between traditional and GCH based approaches to identifying synthesisable structures is complicated by the fact that the GCH is an approach with several variable parameters defining its implementation, whereas the energy cut-off and energy-density hull based methods are uniquely defined.

For this work, the candidate pools and polymorph rankings determined by the energy cut-off and energy-density convex hull based methods were compared to those determined by each GCH implementation investigated in Section 5.4. Figure 5.12 shows the GCH candidate pools previously indicated in Figure 5.5 alongside the candidate pools resulting from the energy cut-off and energy

density convex hull methods.

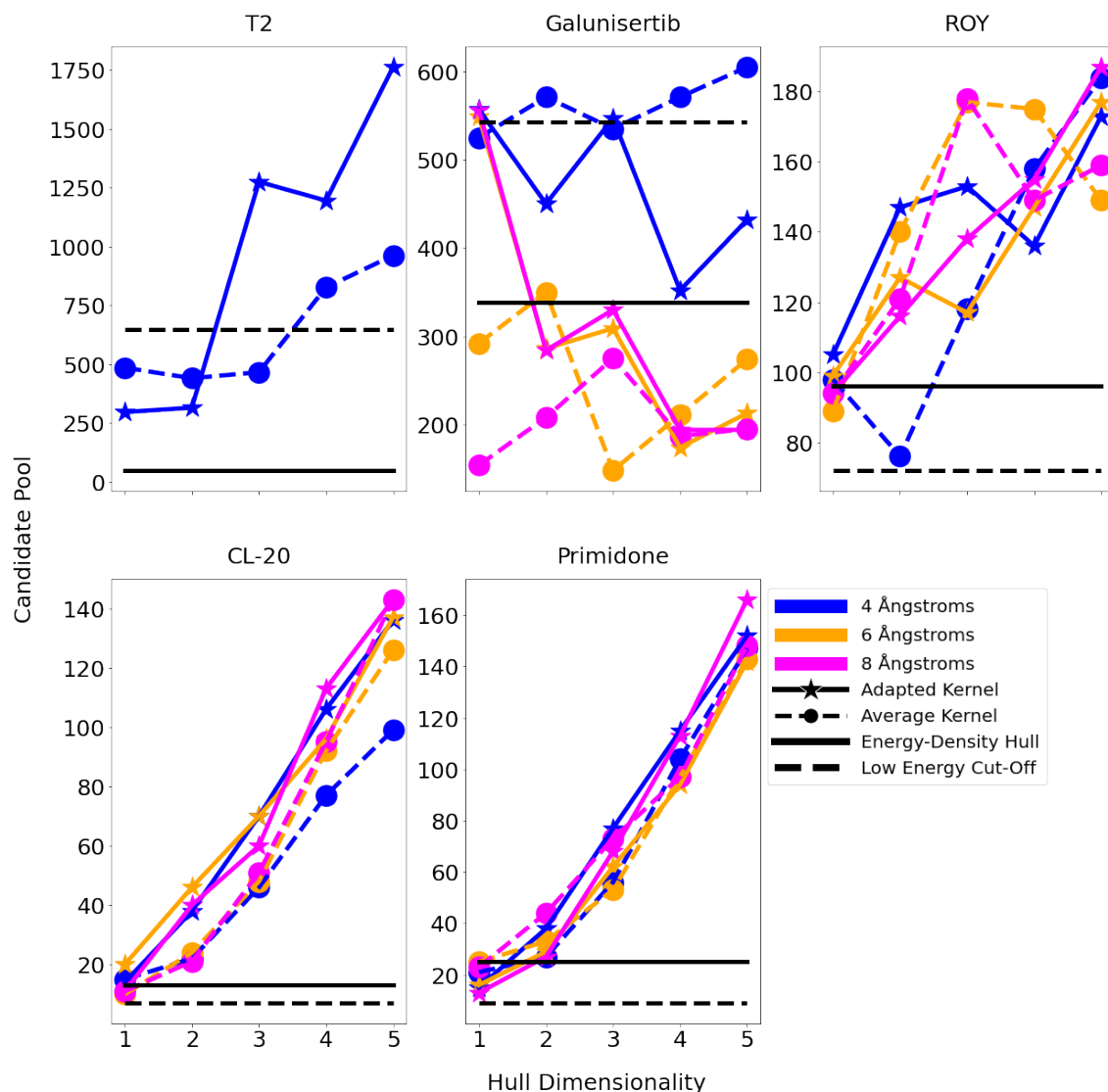


Figure 5.12: Candidate pools for each system as a function of hull dimensionality, SOAP cut-off, and kernel construction for each. Different curves denote different kernel constructions - with different kernel types and underlying SOAP cut-offs. Horizontal bars indicate the candidate pools for each system as calculated via traditional landscape analysis methods.

It can be seen that in many cases, particularly where the GCH implementation incorporates a higher-dimensional hull, the candidate pools arising from traditional methods are lower - suggesting that such approaches could be more efficient in identifying synthesisable structures. However, there are a few exceptions to this - in which the GCH proves more efficient than one or both of the traditional approaches. Particularly dramatic is the case of galunisertib, for which most GCH implementations perform more efficiently than even the energy-density convex hull. These results suggest that it may most commonly be appropriate to proceed with traditional approaches - but

that the possibility of cases for which such approaches are a poor indicator of experimental synthesizability should be noted.

Average polymorph rankings arising from GCH approaches for each system are provided in Table 5.9. The average ranking indicated in each case is that from the best kernel implementation of the GCH in that instance. Cells are coloured according to comparison to traditional methods, being green if the best GCH approach results in a lower average polymorph ranking than the best traditional approach, orange if the best GCH approach is on par, and red if it results in a higher average polymorph ranking than the best traditional approach.

		4 Angstroms	6 Angstroms	8 Angstroms
System	Hull Dimensionality	Best GCH-based Average Polymorph Ranking		
T2	1	81.2	N/A	N/A
	2	134	N/A	N/A
	3	147.4	N/A	N/A
	4	189	N/A	N/A
	5	64	N/A	N/A
Galunisertib	1	76.677	63.222	56.777
	2	79	59.111	47.111
	3	77.444	24.444	49.222
	4	28	6.222	13.333
	5	28.556	2.111	1
ROY	1	16.769	15.846	15.077
	2	9.769	17.231	16.923
	3	7.846	14	16.538
	4	3.692	10.846	6.923
	5	2.308	4	2
CL-20	1	3.667	3.667	3
	2	1.333	2.667	1.333
	3	1	2.3333	1
	4	1	1	1
	5	1	1	1
Primidone	1	7.5	6	3.5
	2	3.5	2	1
	3	1	1	1
	4	1	1	1
	5	1	1	1

Table 5.9: Lowest average polymorph rankings calculated via a GCH implementation for each system, hull dimensionality, and SOAP cut-off radius. Cells are colour-coded according to comparison to traditional methods.

Green = the best GCH approach results in a lower average polymorph ranking than the best traditional approach in that instance.

Orange = the best GCH approach is on par with the best traditional approach

Red = the best GCH approach results in a higher average polymorph ranking than the best traditional approach.

On the one hand, it is shown that the average declared ranking of polymorph matches tends to be lower using GCH approaches than traditional approaches - which appears positive. However, this may simply be due to their being multiple structures of rank 1 in these cases. It cannot be claimed from this finding that the GCH approach actually classes fewer structures as being more stabilisable/synthesisable than any given polymorph than traditional approaches would.

The most confident conclusion that can be drawn from this data is that the poor candidate pools from GCH approaches are likely due either to extreme ranking of one or two polymorphs or, again, due to the presence of large number of hull vertex structures.

## 5.6 Concerns and Considerations

### 5.6.1 Reasonable Constructions

As discussed, the dimensionality of the hull is a key parameter of the GCH implementation. The importance of this parameter, and the unpredictable nature of its impact on results raises two issues with the work.

The first issue is one of applicability to real-life implementation. In the most likely implementation of the work, for novel CSP, an optimal hull could not be constructed via knowledge of the known polymorphs and respective assessment of the ideal hull dimensionality. The included components for hull construction would have to be chosen a-priori. From the work conducted so far, it is difficult to select an appropriate ‘default’ dimensionality for a high quality of results. However, it can be said that this dimensionality should remain low - e.g 3D or below - due to the concerns of increasing candidate pools with high dimensional hulls.

The second issue is one of how to most fairly compare the average and adapted kernel GCH implementations. The ‘optimal’ parameters for each kernel implementation may differ and so it may unfairly hinder one approach or the other to compare ,for example, the average kernel 2D hull candidate pools to the adapted kernel 2D hull candidate pools. Conversely though, comparing the candidate pools from the best performing implementation of the GCH for each kernel gives a somewhat unrealistic view of their performance because, as mentioned, the parameters used in real-life implementation would have to be determined ahead of time.

One possible solution would be to compare the different kernel GCH implementations using hull dimensionalities determined by a required minimum of the cumulative variance captured. Default hull dimensionalities could be selected in the same way. However, as can be seen from the cumulative variance graphs in Figure 5.7, incorporating a significant proportion of the variance of the data would require use of a high-dimensional hull, especially so for the adapted kernel implementations. This ,in turn, would lead to issues with unreasonably large candidate pools and complicated experimental constraints for stabilisation. Due to these concerns, coupled with time constraints upon the research, the possibility was not investigated here.

### 5.6.2 Centering

Another potential issue with this work was that an error - inherited from the original GCH code-base [164] meant that the kernels may have been improperly centred prior to derivation of the kPCA projection.

Kernels in this work had been ‘centered’ by application of the following code:

```
k = kernel.copy()
cols=np.mean(k,axis=0);
rows=np.mean(k,axis=1);
mean=np.mean(cols);
for i in range(len(k)):
    k[:,i]-=cols
    k[i,:]-=rows
k += mean
```

where k at the end is the resulting kernel. This is not guaranteed to appropriately centre a kernel. Accurate kernel centering can be achieved using off-the-shelf functionality in packages such as sklearn[167]:

```
from sklearn.preprocessing import KernelCenterer
k=kernel.copy()
k=KernelCenterer.fit_transform(k)
```

To determine if this had impacted the findings in this work, each calculated kPCA projection used was appropriately recalculated using *KernelCenterer* in Sklearn. Then, the two kPCA projections were compared. It was established that, at least across the first 32 kPCA components, the projected values of each structure along each component were identical to within  $1 \times 10^{-11}$ . 32 kPCA components is the maximal set explored in investigation of intrinsic dimensionality in this work, and significantly more than the number of components used anywhere else in the research. Therefore, this issue should not have significantly impacted the results discussed in this chapter - or those in Chapter 6.



## 5.7 Concluding Remarks

This chapter discussed work comparing the effectiveness of different landscape analysis methods in identifying synthesisable crystal structures from CSP datasets. Methods investigated were the energy cut-off (identifying synthesisable structures based purely on their relative lattice energy), extracting structures via an energy-density convex hull, and identifying structures using GCH approaches - incorporating either an average or adapted SOAP kernel.

The main metric by which the methods were compared was the size of the candidate pools required to capture the known polymorphs of a given system using each landscape analysis method. The average and maximum rankings of polymorph structures for each system as determined by each landscape analysis method was also explored.

The findings of the work were largely inconclusive. The relative performance of the average and adapted kernel GCH implementations could not be consistently determined, and was heavily dependent upon the system explored and the additional factors of the hull dimensionality and underlying SOAP cut-off radius used. The noted performance gaps in many cases may not be significant. Further, there were systems for which it was more effective to identify observed structures as synthesisable via an energy-cut off than via proximity to energy-density convex hull. However, there were also systems for which the converse was true. Unfortunately, in most cases, the GCH approaches proved less effective than traditional methods in identifying observed structures as synthesisable. That said, there were yet again counter-examples to that finding. These tests were used to assess the utility of the methods, based on the assumption that the identification of observed structures as synthesisable corresponds well to the general ability of a method to identify synthesisable structures. Using that assumption, these findings can be said to show the variable relative performance of different methods in identifying synthesisable structures.

Primarily, perhaps, the greatest takeaway from the research in this chapter is a cautionary reminder that determining the most likely synthesisable structures from sets of predicted crystal structures is a difficult task, sensitive to many decisions made by the researcher during the process. Therefore, resulting sets of identified candidates should be viewed with this in mind.

## Chapter 6

# Gathering Meaning from the Kernel

### 6.1 Overview

As discussed in Section 2.4, the GCH identifies structures that are stabilisable by some constraint(s) coupled to the descriptors used to construct the hull. However, unlike with a case such as the energy-density convex hull, the machine-learned descriptors used are somewhat abstract. Loosely speaking, the descriptors correspond to the similarity of a given structure to some linear combination of all other structures [1]. However, the use of the underlying atomic-environment based SOAP descriptors allows the machine-learned components to ‘pick up’ on structural features. Therefore, a machine-learned descriptor may relate to a more intuitive feature such as density or molecular conformation.

Given that kPCA components/ML descriptors used for GCH construction are derived from SOAP similarity kernels rather than from combinations of simple variables, more intuitive interpretations of the descriptors cannot be obtained directly from the components or the underlying PCA weightings themselves. In order then to find these interpretations, components can be plotted against more conventional descriptors to identify possible relationships. This can then in theory facilitate understanding of how the ML descriptors can be used to explore CSP landscapes and identify potential constraints that could be used to stabilise near-hull structures. Further, and perhaps most pertinent to this thesis, exploration of these ML descriptor - intuitive descriptor relationships can be used as a means to evaluate the kernel constructions used to derive the ML descriptors. Under the assumption that a more reasonable and useful SOAP kernel would be better able to pick out meaningful structural features, the strength of these relationships as arising from different kernel constructions can be used as a means of comparing kernel choices. The ‘top-ranked’ ML descriptors - i.e the kPCA components with the highest eigenvalues - still encapsulate structural variance and can

prove useful in discriminating between structures and identifying stabilisable structures. However, being able to interpret these descriptors both affirms the reasonable construction of the kernel and offers additional potential in identifying stabilising constraints.

This chapter discusses work to identify intuitive interpretations of the ML descriptors for several systems, and explores how the strength of the relationships varies with the construction of the kernel from which they are derived. Discussion opens with explanation of some common and useful intuitive structural descriptors for molecular crystals, as well as tools that can be used to calculate these descriptors for predicted structure sets. Then, encountered relationships between these descriptors and derived ML descriptors are discussed, along with both qualitative and systematic comparison of the strength of these relationships between the corresponding kernel constructions. Lastly, some limitations of the research and suggestions for future work are posed.

## 6.2 Intuitive Descriptors

### 6.2.1 Overview

This section discusses some key intuitive descriptors for molecular crystal structures and the calculation of these descriptors for predicted structure sets. The intuitive descriptors discussed here are those for which attempts were made to identify relationships to ML descriptors.

### 6.2.2 Density

Density is a readily available/calculable property of a crystal structure. It is a useful feature to explore in possible interpretation of kPCA components for the GCH as it has clear links to known crystallisation constraints such as pressure and solvent inclusion and so presents a useful case for testing the ability of the GCH to identify structures stabilisable by such constraints. Further, as energy-density landscapes are a common method for visualising and exploring CSP datasets, it is useful to test whether such a landscape is always ‘optimal’ or whether there may be other structural features through which it may be beneficial to compare structures and explore CSP landscapes.

### 6.2.3 Molecular Conformation

Another key structural descriptor for molecular crystal structures is the underlying in-crystal molecular conformation. The in-crystal conformation can be an important property, for instance impacting the colour of ROY crystals [55]. Further, it is of particular interest in this work because, if a molecule is sufficiently flexible, molecular conformation could be the source of significant structural variance. It is also a property that can be impacted by experimental constraints. For example, the solvent used can affect the conformations present [168].

There are several measures that can be used to describe in-crystal conformation. Approaches used in this work are to employ measurement of intramolecular angles, and grouping/classification of conformations.

### 6.2.4 Hydrogen Bonding

Another intuitive descriptor investigated in this work was that of the hydrogen bonding motifs. Hydrogen bonding is an important structural feature of molecular crystals as it can influence properties such as stability, solubility, and mechanical properties [169]. Hydrogen bonding motifs within predicted structures can be identified using the *motif search* in Mercury.

### 6.2.5 Unsuccessful Intuitive Descriptors

Some preliminary work also attempted to identify relationship between ML descriptors of a porous system - DAP - and porosity descriptors including pore dimensionality and the largest accessible free sphere. All porosity descriptors were calculated using open -source software Zeo++ [170]. However, no relationships were identified that were thought to be of benefit to explore further. Some relation was identified between ML descriptors and the largest accessible free sphere - however it was thought that this relationship was likely a proxy of a clearer relationship to density. As a result, porosity descriptors were not investigated further in this thesis.

Attempts were also made to identify relationships between ML descriptors of semiconductor molecule NTCDA and a descriptor of the aromatic interactions in the crystal structure. This used a script designed to mimic and expand upon the functionality available in *AromaticsAnalyser* in Mercury. For each crystal structure the number of key aromatic interactions classified as  $\pi$ - $\pi$  stacking interactions was calculated as was the average inter-centroid distance and offset of those interactions. However, the search for relationships of these intuitive descriptors to ML descriptors did not prove fruitful. Due to this, and the shift of research focus away from semiconductor molecules, this was not investigated further.

## 6.3 Relationships to kPCA Components

### 6.3.1 Overview

This section outlines the qualitative relationships between ML descriptors and intuitive descriptors that were successfully identified.

The systems explored in this work were selected based upon there being potential intuitive descriptors which could be expected to comprise significant structural variance and be useful for exploring the CSP landscapes. This makes such systems useful for testing the ability of the SOAP kernel and kPCA decomposition to pick out these structural features and derive interpretable ML descriptors.

On this basis, the systems selected included ROY, galunisertib, DAP and T2. These systems, and the origins of the CSP data have been discussed previously. Another system, trimesic acid, was also explored. The CSP data for trimesic acid was previously predicted using CSPy [171].

### 6.3.2 ROY

In the case of ROY crystal structures, a known important structural descriptor is that of underlying molecular conformation. The key flexibility in the ROY molecule is the torsion angle ( $\alpha$ ) indicated in Figure 6.1 [55].

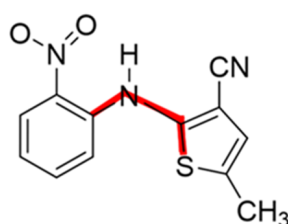


Figure 6.1: Structure of the ROY molecule - the bonds defining the key torsional angle  $\alpha$  are highlighted in red.

The value of this intramolecular torsion was therefore a sensible choice of intuitive descriptor to explore for this system. For the set of predicted ROY crystal structures [55], this angle was measured for each molecule in the asymmetric unit for each crystal structure. Then, to conceptualise the descriptor as the angle between the planes of the two rings, the absolute value of the torsion was taken in each case. Lastly, it was necessary to obtain a single descriptor value for each crystal

structure. For each of the three  $Z'=2$  structures the mean of these values across the molecules of the asymmetric unit was taken. For these cases, this intramolecular torsion angle was similar in each molecule of the asymmetric unit - meaning that the average should be largely representative of both molecules in the asymmetric units (Figure 6.2).

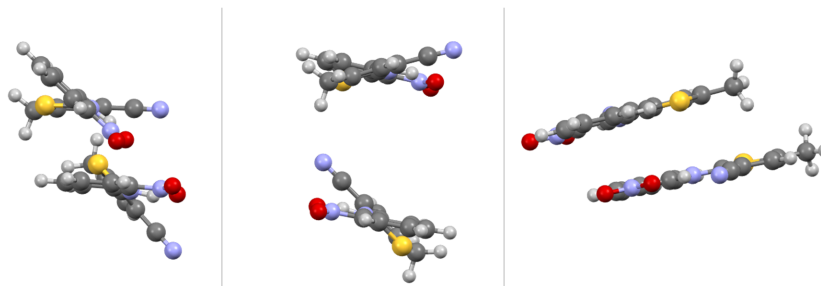


Figure 6.2: Visualisations of the asymmetric units of the three  $Z'=2$  experimental crystal structures of ROY.[55]. It can be seen that in each respective asymmetric unit, the two molecules will share similar absolute values of torsion  $\alpha$ .

To explore qualitative relationships between this descriptor of molecular conformation and the ML descriptors derived from different kernels, GCH landscapes were plotted - with data points being coloured by the value of the intuitive descriptor. As an initial test, this was performed for the case of the 1D GCH landscape from the adapted kernel, using a 4 Å SOAP cut-off. This can be seen in Figure 6.3

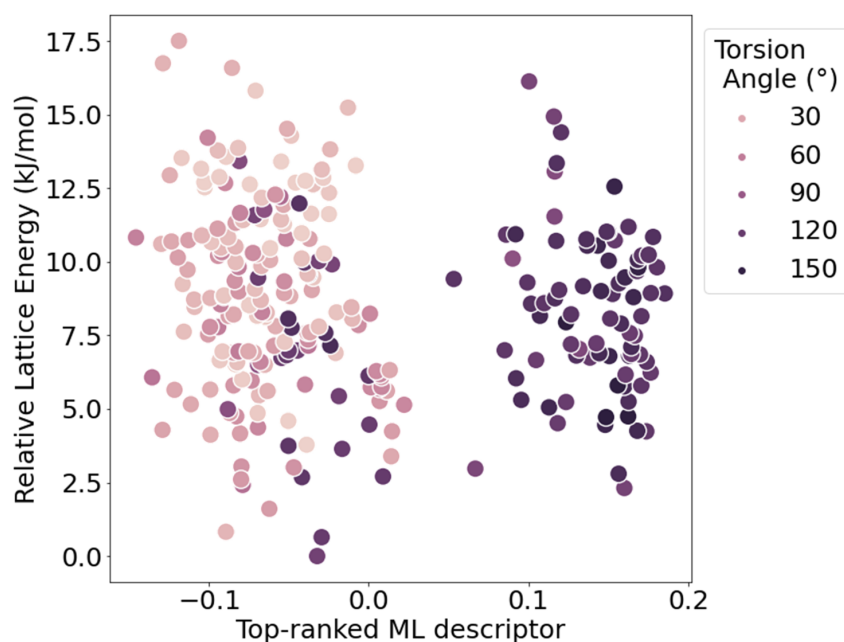


Figure 6.3: 1D GCH landscape of ROY (4Å SOAP cut-off, Adapted kernel construction) with points coloured by values for the key torsional angle  $\alpha$  in the underlying molecules.

This suggested a qualitative relationship between the top-ranked kPCA component and molecular conformation. Roughly speaking, the ML descriptor separates crystal structures into two ‘islands’ corresponding to structures of mostly large and mostly small intramolecular torsion angle. To simplify the relationship, the molecular conformation descriptor was then reformulated as a categorical descriptor - through binary classification of the structures. The new molecular conformation descriptor described ROY crystal structures as being of one of two classes:

- class 0 if  $\alpha < 90^\circ$
- class 1 if  $\alpha \geq 90^\circ$

Figure 6.4 shows the same GCH landscape as in Figure 6.3, coloured by this reformulated descriptor, alongside an equivalent plot - albeit with the GCH landscape derived via the average kernel.

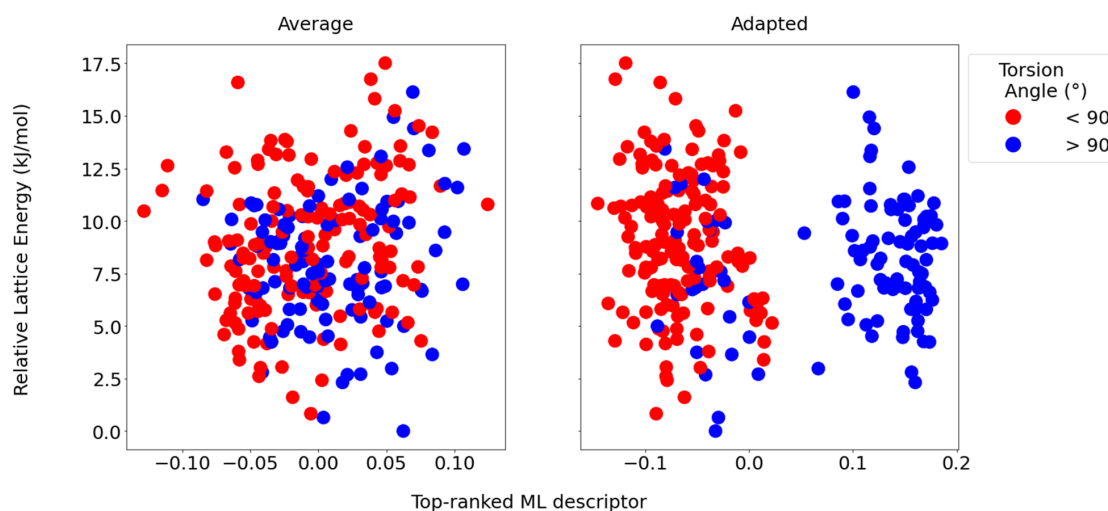


Figure 6.4: 1D GCH landscapes (4 Å SOAP cut-off) of ROY from the average and adapted kernel constructions - coloured by the value of the molecular conformation binary classification descriptor.

This, again, shows a clear relationship between the top-ranked ML descriptor derived from the adapted kernel and an intuitive descriptor - the reformulated molecular conformation descriptor. It can also be seen that the same relationship may be present for the top-ranked ML descriptor as derived via the average kernel, though the separation of structures is poorer.

The identification of this relationship, particularly given its lucidity in the case of the adapted kernel, is a reassuring sign. It confirms that the SOAP kernel constructions used in this work - including that of the adapted kernel - can pick out meaningful structural features.



Some consideration, however, must be given to other derived ML descriptors - not just those top-ranked in the kPCA decomposition in each case. The equivalent data was plotted for the top-five ranked ML descriptors from each kernel, in an attempt to search for other ML descriptors that may relate to molecular conformation. This search showed that for the adapted kernel, the qualitative relationship appeared strongest for the top-ranked ML descriptor - with other ML descriptors being notably less related to the intuitive descriptor. However, for the average kernel case, the second-ranked ML descriptor appeared to be most strongly related to molecular conformation (Figure 6.5).

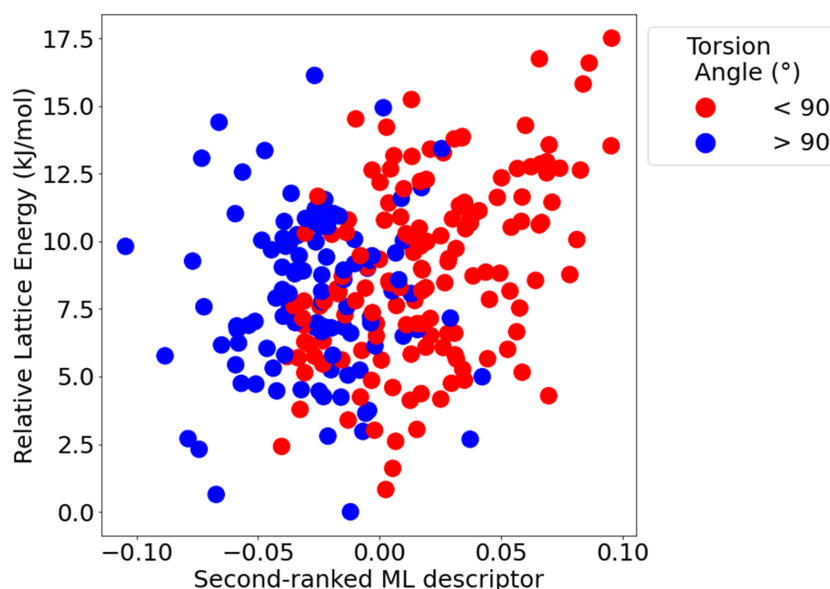


Figure 6.5: 1D GCH landscape of ROY (4Å cut-off) using the second-ranked ML descriptor from the average kernel construction. Points are coloured according to the molecular conformation classification descriptor.

Lower-ranked ML descriptors were not considered here as it was assumed that the important structural information, relating to meaningful intuitive descriptors, would be captured within the highly ranked ML descriptors. Lower-ranked ML descriptors - which capture less structural variance - are less likely to contain useful structural information. Another factor that must be considered is the impact of the underlying SOAP cut-off radius used in kernel calculations. Discussion of this impact, and further discussion of the dependence upon the ML descriptors selected, is given in Sections 6.4.2 and 6.4.3 respectively.

### 6.3.3 Galunisertib

#### Molecular Conformation

Relationships between ML descriptors and molecular conformation descriptors were also investigated for the system of galunisertib. Preliminary work attempted to identify relationships between ML descriptors and flexible torsions in the galunisertib molecule, but this did not prove informative.

A second attempt to explore molecular conformation descriptors was then made. For each crystal structure in the prediction set, the asymmetric unit molecules were extracted and geometry optimised at DFT level (PBE0+GD3BJ/6-31G\*) implemented in Gaussian09. These structures were then clustered to obtain the set of unique conformers. Each  $Z'=1$  crystal structure in the set was assigned a label denoting the unique conformer which could be most closely overlaid with its in crystal conformation.  $Z'=2$  structures were not included in this investigation due to the complexity of assignment in these cases.

However, the resulting dataset was too complex to effectively search for clear relationships to ML descriptors - with there being 53 unique conformers to which crystal structures had been assigned. To resolve this, the conformers were grouped into 'families' using agglomerative clustering and the crystal structures were reassigned labels based on the family to which their previously assigned conformer belonged.

The hierarchical clustering was performed using a pre-computed pairwise distance matrix, that provided the RMSD of overlay of each pair of conformers. Using this distance matrix, clustering was performed via the *AgglomerativeClustering* functionality in sklearn. This clustering was performed so as to form eight families. It can be seen from the hierarchical clustering dendrogram (Figure 6.6) that this is a sensible number of clusters.

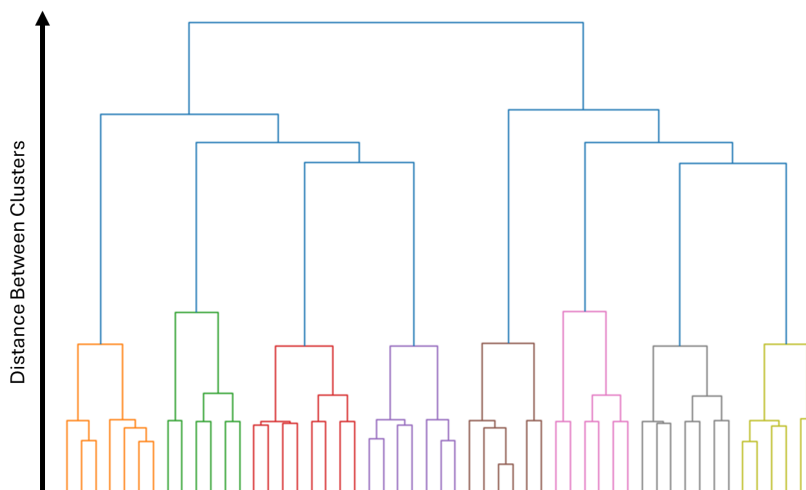


Figure 6.6: Truncated dendrogram for the agglomerative clustering upon the full galunisertib conformational assignment data. Each ‘splitting’ of the tree represents the separation of two clusters. The y axis denotes the distance between clusters and their parent clusters. Each of the eight clusters used in final clustering has been uniquely coloured within the diagram.

Whilst the 8 clusters are clearly distinct from their respective parents, splitting the dataset into further clusters would introduce less additional variance between clusters. Further, eight clusters, was a number that was feasible to work with when attempting to visually identify relationships.

Figure 6.7 shows the 1D GCH landscapes, using varying kernel constructions, of  $Z'=1$  galunisertib structures. The landscapes have been coloured by the conformer family that most closely matched the in-crystal molecular conformation of the structures. All GCH landscapes shown here used a 4 Å SOAP cut-off radius for kernel construction.

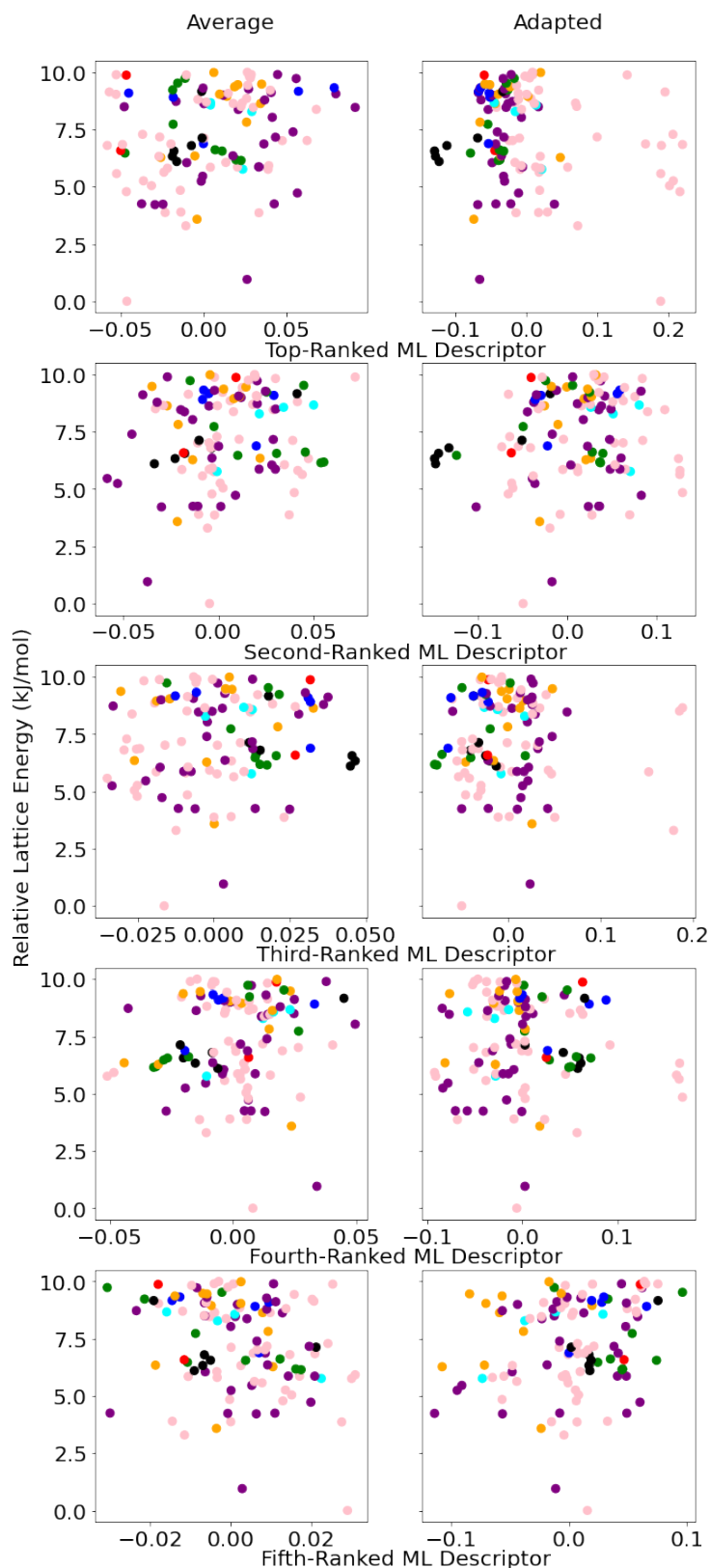


Figure 6.7: 1D GCH landscapes of  $Z'=1$  Galunisertib crystal structures using differently ranked ML descriptors derived from the average and adapted kernel constructions. All kernel constructions used a 4 Å SOAP cut-off. Points are coloured according to the cluster to which the structure was assigned based upon its in crystal molecular conformation.

These plots indicated a possible weak relationship between molecular conformation and ML descriptors for the galunisertib system. It can be seen for the adapted kernel case that all structures taking high values for the top-ranked ML descriptor have been assigned to the same cluster. However, this does not appear to be a strongly indicative relationship - and was also not seen for the average kernel case. This relationship to molecular conformation was not investigated further in this thesis, but future work could expand upon this exploration - for instance considering the possible relationships had the intuitive descriptor assignments been performed differently.

### Hydrogen Bonding

Another intuitive descriptor of interest for galunisertib crystal structures is description of the hydrogen bonding motifs that they contain. The galunisertib molecule has four hydrogen bond acceptors and two equivalent hydrogen bond donors. Therefore, a hypothetical crystal structure can contain any combination (or none) of the four corresponding hydrogen bonding motifs. The hydrogen bonding in galunisertib crystal structures has been shown to be an important source of structural variance - differing, for instance, between experimental polymorphs [15].

To investigate the hydrogen bonding motifs in the structure set, a search was performed using *motif search* in Mercury - searching for hydrogen bonding between the donor and the four hydrogen bond acceptors. The ‘molecular fragments’ used in the motif search were designed to mimic those shown in discussion of galunesertib hydrogen bonding in reference [15]. The search was performed to identify all such intermolecular hydrogen bonds with lengths  $\leq \sum(vDW radii) + 0.1 \text{ \AA}$ . An illustration of the fragments defining this search can be seen in Figure 6.8. For this work, the full galunisertib structure set - including  $Z'=2$  structures- was used.

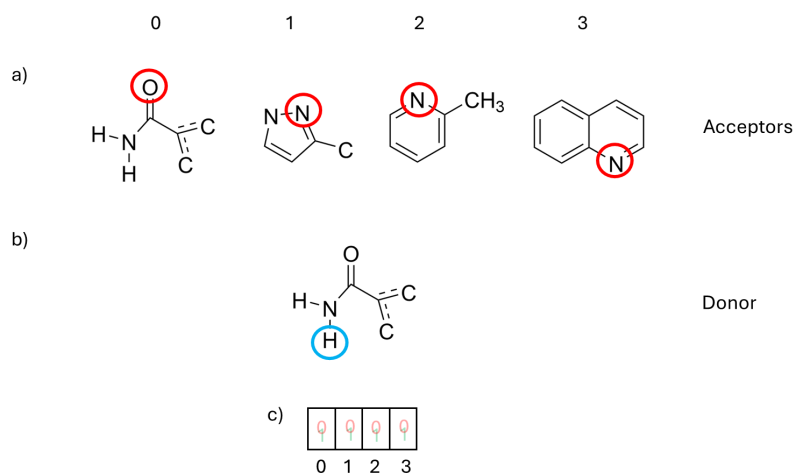


Figure 6.8: Fragments of molecular structures used within the motif search to identify utilised hydrogen bond acceptors (a) and donors (b) within the crystal structures. The vector construction shown in c) indicates how class labels were applied to structures base upon the presence (1) or absence (0) of hydrogen bonds utilising the respective acceptors.

From the resulting data, structures were classified according to which motifs could be found within the crystal structure - using labels of concatenated binary values (0/1) to indicate the presence or absence in the crystal structure of hydrogen bonds with the corresponding motif (Figure 6.8(c)). Figure 6.9 shows a single example of a 1D GCH landscape, coloured by this complete hydrogen bonding motif classification of each structure.

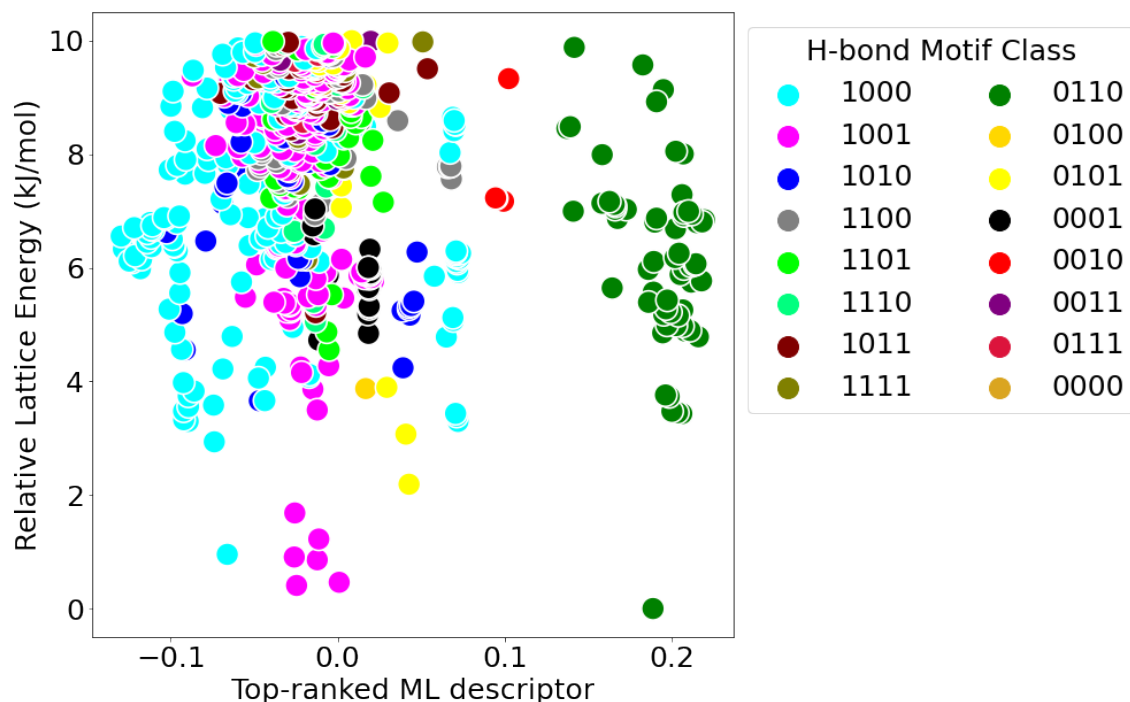


Figure 6.9: 1D GCH landscape of galunisertib (Adapted kernel construction, 4Å SOAP cut-off). Points are coloured by the class to which each structure was assigned based upon the hydrogen-bonding motifs present in the crystal.

From this initial exploration, it was determined that a possible qualitative relationship may be found between top-ranked ML descriptors and whether or not the crystal structure contained hydrogen bonding utilising the oxygen hydrogen bond acceptor (Acceptor 0 in Figure 6.8 a. All motifs whose label begins with ‘1’ utilise this acceptor). The intuitive descriptor was then reformulated to classify each crystal structure simply by whether or not that oxygen hydrogen bond acceptor was utilised. Figure 6.10 shows the 1D GCH landscapes, using varying kernel constructions, of galunisertib structures. The landscapes have been coloured by this binary classification. All GCH landscapes shown here used a 4 Å SOAP cut-off radius for kernel construction.

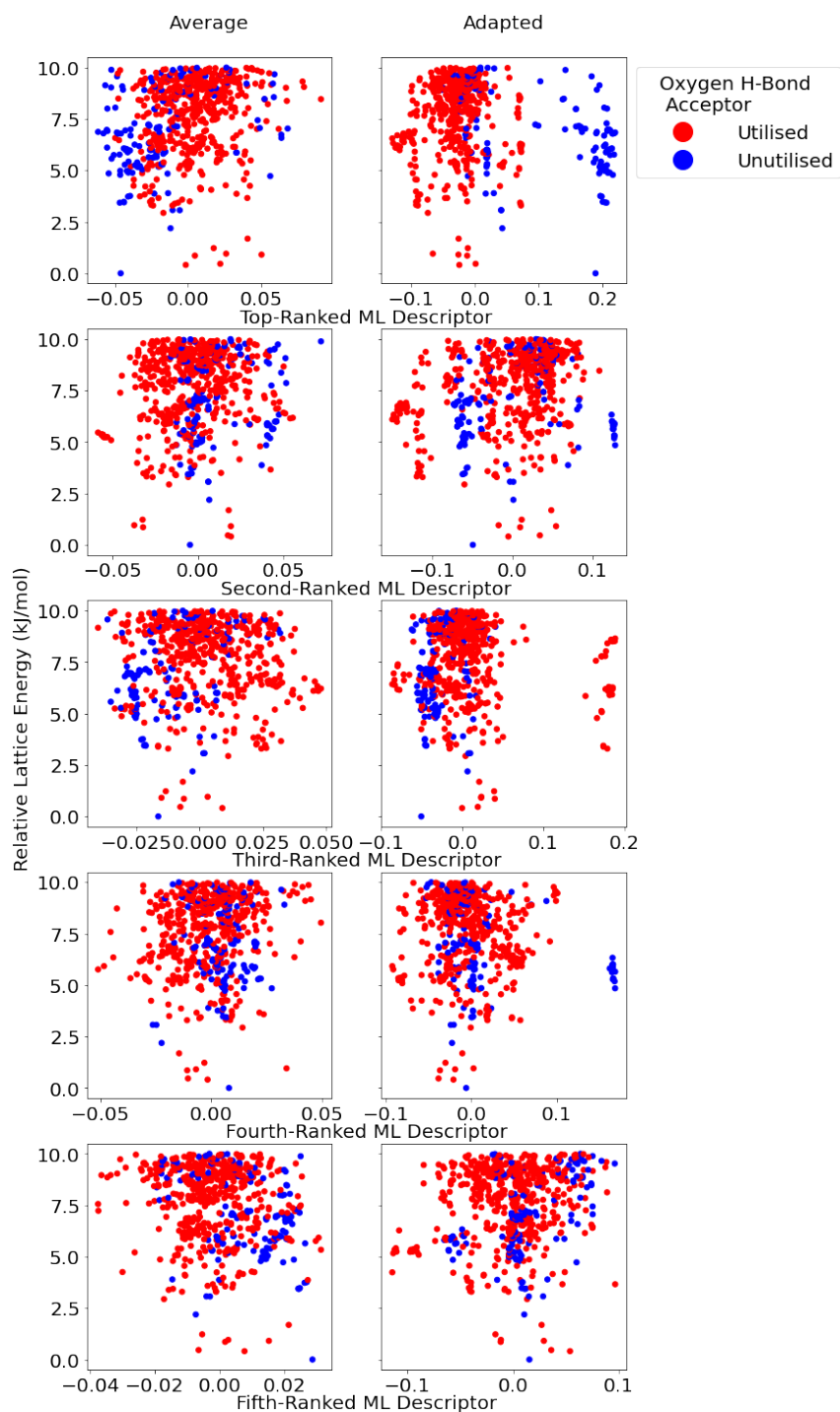


Figure 6.10: 1D GCH landscapes of galunisertib crystal structures using differently ranked ML descriptors derived from the average and adapted kernel constructions. All kernel constructions used a 4Å SOAP cut-off. Points are coloured according to the cluster to which the structure was assigned based upon its use or neglect of the oxygen h-bond acceptor.



It can be seen that there is some qualitative relationship of ML descriptors to this binary classification - particularly for the top-ranked ML descriptors. The structures corresponding to the two different intuitive classes are distributed differently across the ML descriptor - effectively forming overlapping ‘clusters’ of each intuitive class. This is particularly apparent for the top-ranked ML descriptor derived from the adapted kernel. This suggests that ML descriptors derived from SOAP kernels may also be able to pick out hydrogen bonding information.

#### **6.3.4 Porous Systems**

For systems likely to contain porous structures, a key intuitive descriptor is density. For the systems of DAP, T2, and trimesic acid - relationships were sought between density and highly-ranked ML descriptors derived from different kernels. For all systems, such relationships were identifiable. Examples are shown in Figure 6.11

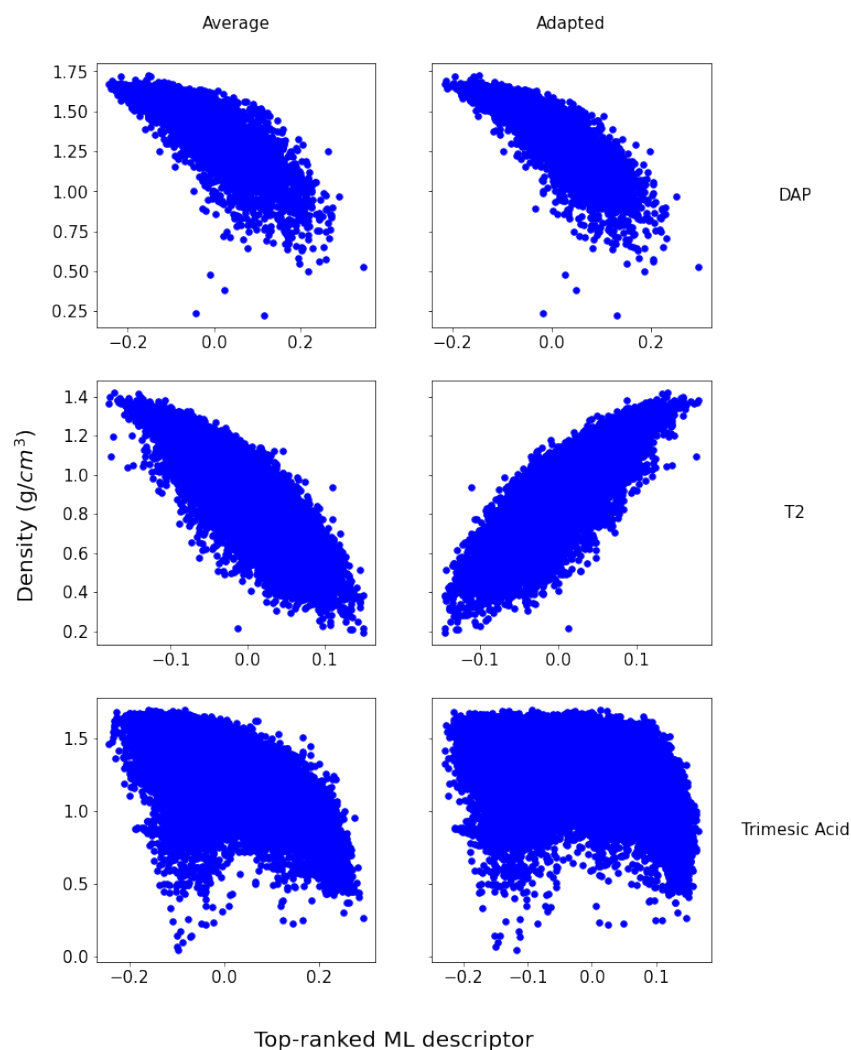


Figure 6.11: Plots of the top-ranked ML descriptors (derived using average and adapted kernel constructions with a 4 Å SOAP cut-off) for the systems of DAP, T2 and trimesic acid against the density of the crystal structures.

In all cases bar one, the relationship to density was clearest for the top-ranked ML descriptor arising from each kernel construction - these relationships being significantly stronger than for any other ML descriptors. The exception to this was for the adapted kernel construction (4 Å SOAP cut-off) for trimesic acid. In this instance, the second-ranked descriptor appeared most closely related to density. Similarly, for the average kernel case for trimesic acid, both the first and second ranked ML descriptors showed a relationship to density - the density ‘character’ seemingly being split across the first two kPCA components (Figure 6.12).

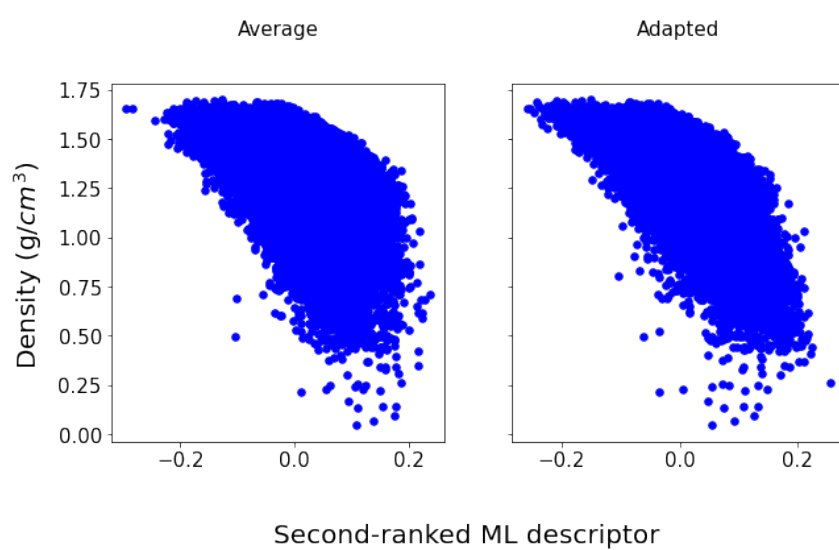


Figure 6.12: Plots of the second-ranked ML descriptors (derived using average and adapted kernel constructions with a 4 Å SOAP cut-off) for the systems of trimesic acid against the density of the crystal structures.

## 6.4 Investigating Relationship Strength

### 6.4.1 Overview

These promising trends having been identified, it is useful to extend the investigation. The following section explores quantification of the relationships and their sensitivity to the SOAP cut-off radius used.

### 6.4.2 Impact of SOAP Cut-Off

The underlying SOAP cut-off radius used was an important factor to consider when exploring kPCA component-intuitive descriptor relationships. This is because the cut-off radius defines the volume around each atom that is considered as its environment - i.e it impacts what structural information is contained within the SOAP descriptors and hence the SOAP kernel. Therefore, the SOAP cut-off radius may impact what structural features the SOAP kernel can identify.

This was explored for example cases using the adapted kernel. Work focussed on the most promising ML-intuitive descriptor relationships identified and on systems for which multiple kernels, with different SOAP cut-offs, could be quickly calculated. These cases were the relationship to molecular conformation for the ROY system, the relationship to hydrogen bonding for the galunisertib system, and the relationship to density for the DAP system. In each case, the work explored only relationships between the intuitive descriptor and the **top-ranked** ML descriptor. For all corresponding relationships, the top-ranked component had been previously suggested by the findings in Section 6.3 to be the key ML descriptor for that relationship. Equivalent investigation for ML descriptors derived from average kernels is not shown here. However, brief testing confirmed that the same qualitative trends of the impact of SOAP cut-off could also be found in those instances.

Figure 6.13 shows the 1D GCH landscapes of the ROY system, using the top-ranked ML descriptor from the adapted kernel and varying SOAP cut-off radii. Each landscape is coloured according to the binary classification of molecular conformation discussed in Section 6.3. That is, each landscape is coloured by whether or not the key torsion angle in the underlying molecule was classed as acute.

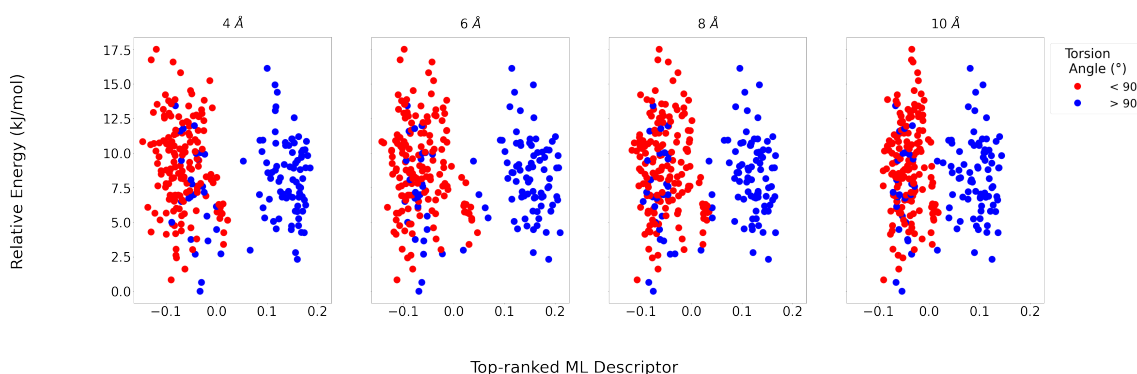


Figure 6.13: 1D GCH landscapes of ROY from the adapted kernel constructions using various SOAP cut-off radii- coloured by the value of the molecular conformation binary classification descriptor.

It can be seen that, while the qualitative relationship remains identifiable, the ‘separation of islands’ decreases. The relationship between the ML descriptor and intuitive descriptor weakens with increasing SOAP cut-off radius. The kernel, and derived top-ranked descriptor, becomes poorer at discriminating between structures on the basis of molecular conformation. This finding is in line with expectation. Molecular conformation is primarily a short-range structural feature - with its biggest influence being upon the intramolecular contributions to the environment of each atom. Therefore, it is reasonable that the identified relationship would weaken with increasing SOAP cut-off as the relevant short-range structural information would be diluted by inclusion of longer-range information within the wider cut-off radius. It is reassuring, with regard to the ability of the GCH approach to derive meaningful ML descriptors, that the impact of the SOAP cut-off on relationship strength follows expectation.

Hydrogen bonding, whilst intermolecular, is also a short range feature - with the X-A distances in typical hydrogen bonds ranging from around 2.2-4Å [172]. Therefore, the relationship between ML descriptors and hydrogen bonding for the galunisertib system would also be expected to weaken with increasing SOAP cut-off. This again appears to be the case, particularly at longer cut-offs (8 Å and 10 Å), as can be seen in Figure 6.14

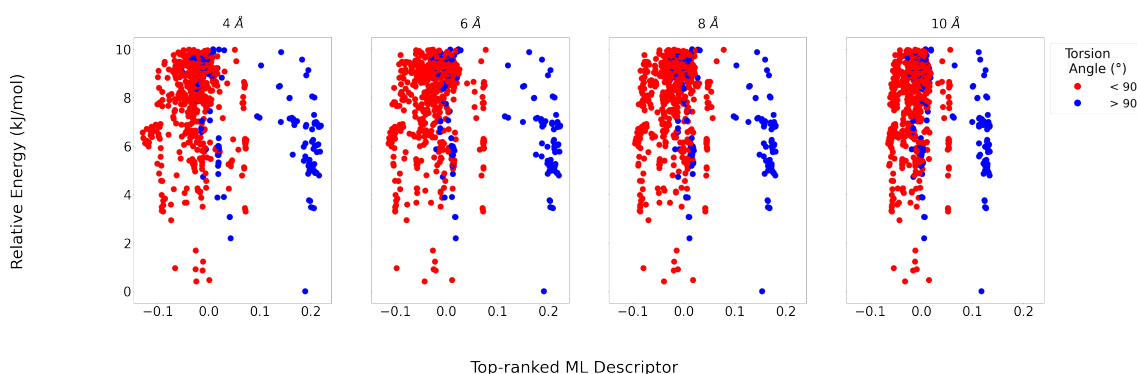


Figure 6.14: 1D GCH landscapes of galunisertib from the adapted kernel constructions using various SOAP cut-off radii- coloured by whether or not the crystal structures utilised the oxygen h-bond acceptor.

Density, by contrast, is a long-range structural feature. Therefore, it would be expected that relationships between density and the ML descriptors derived from SOAP kernels would strengthen with increasing SOAP cut-off radius. Though perhaps less clear, it appears that this is true to an extent for the DAP system (Figure 6.15). There is however, some indication that the relationship may be non linear. Further investigation of quantifying the non-linear relationship or exploring intuitive descriptors that may demonstrate a more linear relationship could be of interest for future work.

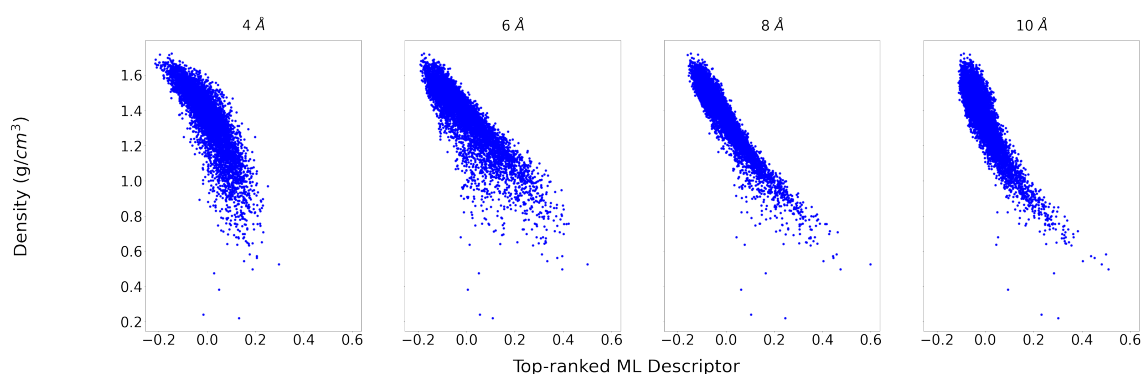


Figure 6.15: Plots of the top-ranked ML descriptors of DAP (derived using adapted kernel constructions with varying SOAP cut-offs) against the density of the crystal structures.

Particularly, the ML descriptor-density correlation appears notably stronger at an 8 Å cut-off than at 4 Å. However, the relationship then weakens again when further increasing the cut-off to 10 Å. Further quantitative investigation of this relationship strength is given in Section 6.4.3.

### 6.4.3 Systematically Comparing Kernels

Potential qualitative relationships having been identified, it is useful to quantify the strength of these relationships and investigate if that is impacted by the construction of the kernel used to derive the machine-learned descriptors.

For the case of the relationships identified between ML descriptors and density, quantifying the strength of the relationship is straightforward. The relationships can be seen to be linear correlations. Therefore, a measure of the strength of the relationship can be found by simply taking the coefficient of determination,  $R^2$ . This is equal to the square of the Pearson correlation coefficient:

$$R^2 = \left( \frac{\sum z_x z_y}{N} \right)^2 \quad (6.1)$$

$$z_x = \frac{x - \mu_X}{\sigma_X} \text{ and equivalent for } z_y$$

where  $N$  is the number of  $x, y$  pairs in the dataset,  $\mu_X$  is the mean of variable  $X$  and  $\sigma_X$  is the standard deviation of variable  $X$  [173].

This value describes the extent to which the density could be predicted by linear regression to the ML descriptor. Table 6.1 shows examples of the  $R^2$  - i.e the strength of the relationship - for the **best** correlations between a highly-ranked ML descriptor and density for the systems of DAP, T2, and trimesic acid. Results using both adapted and average kernels to derive the ML descriptors are shown and all kernels used a 4 Å cut-off radius.

System	Kernel Type	$R^2$	Rank of ML Descriptor
DAP	Adapted	0.69724	1
	Average	0.63235	1
T2	Adapted	0.69652	1
	Average	0.69785	1
Trimesic Acid	Adapted	0.56436	2
	Average	0.35882	1

Table 6.1:  $R^2$  values for the best ML descriptor-density relationships identified for the DAP, T2, and trimesic acid systems using each kernel type. All kernel constructions used a 4 Å SOAP cut-off.

These results show that the best ML descriptor-density relationships for the adapted kernel cases

are on a par with or better than that for average kernel cases. Here, the average kernel only outperforms the adapted kernel for the case of T2, and this distinction is subtle ( $R^2 = 0.697$  vs  $0.698$ ). The adapted kernel much more strongly outperforms the average kernel for the systems of DAP and trimesic acid. For trimesic acid, this may be due to the ‘splitting’ of density character across the first two components from the average kernel. These stronger relationships suggest that the adapted kernel may perform better than the average kernel in extracting useful structural information - leading to more meaningful ML descriptors.

Using the same  $R^2$  metric, the relationship strengths for the DAP system were explored additionally as a function of SOAP cut-off (Table 6.2 ).

Kernel Type	SOAP cut-off (Å)	$R^2$	Best Component
Adapted	4	0.69724	1
	6	0.75720	1
	8	0.88306	1
	10	0.79733	1
Average	4	0.63235	1
	6	0.75909	1
	8	0.88197	1
	10	0.80023	1

Table 6.2:  $R^2$  values for the best ML descriptor-density relationships identified for the DAP systems using each kernel type and various SOAP cut-off radii.

These findings confirm the trend proposed in Section 6.4.2 that the strength of ML descriptor - density relationships generally increases with increasing SOAP cut-off. However, the results also complicate kernel comparisons, as at some cut-offs the adapted kernel outperforms the average and at others the converse is true. It therefore appears that the comparison is not straightforward, and that the impact of SOAP cut-off affects the relative utility of the kernels in this respect.

The correlations between intuitive and machine-learned variables indicated in systems with relationships to density are only the simplest of cases. Such relationships are perhaps the clearest and most intuitive. Additionally, they may also be of the most direct application to materials discovery - with the ‘link’ to stabilisation by a constraint related to the intuitive descriptor being more straightforward in these cases. However, other intuitive-ML descriptor relationships remain useful



for exploring CSP landscapes and proposing experimental constraints that may separate possible crystal structures - even when the connection cannot be described merely by a correlation. The strength of these relationships can also still be used as a means by which to assess the utility of the kernels in deriving meaningful ML descriptors - as a proxy to assessing their reasonable underlying construction. It therefore remains of interest to evaluate the impact of kernel choice on these more complex relationships.

The relationships of ML descriptors to molecular conformation and hydrogen bonding for the systems of ROY and galunisertib respectively cannot be described as correlations. Therefore, their strength cannot be evaluated using  $R^2$ . However, it can be seen, for instance from Figures 6.4, 6.5, and 6.10 that the intuitive descriptors are related to the distribution of structures across the ML descriptor, with structures sharing an intuitive class being somewhat grouped along the ML descriptor.

This phenomenon, alongside the nature of the intuitive descriptors as binary classifiers, suggests that the performance of supervised machine-learning classification models could be used to assess the strength of the relationships - and that, in particular, a useful model may be Support Vector Classification (SVC) (See Section 2.5.2). This form of assessing relationship strength relies upon the assumption that - for an otherwise equivalent ML model - greater performance in target prediction corresponds to a stronger, or as a minimum clearer and easier to learn, relationship between feature and target in the training data. Therefore, if machine-learning were used to predict the value of the intuitive descriptor from the value of the ML descriptor, then higher learning performance would correspond to a stronger relationship between the ML and intuitive descriptors. Soft-margin SVC is an obvious choice of model to apply in these cases, as the ‘clustering’ of structures of each intuitive class along the ML descriptor suggests that use of a separating hyperplane may be able to discriminate well between intuitive classes.

The strength of the identified relationships for each system was quantified in this way, separately training and assessing models that learned from the ML descriptors derived from each constructed kernel. Therefore, the learning performance - and so relationship strength - was tested as a function of both the SOAP cut-off and kernel type.

Support Vector Classification was implemented in *sklearn* - using a linear kernel- to predict the intuitive classification of each crystal structure from the value of the machine-learned descriptor(s)

for that structure. Full parameterisation of the models was not performed. However, separate models were trained and assessed using varied values (Powers of 10 from 0.1 - 1000) of model parameter C. Parameter C controls the strength of the penalty on misclassification, with higher values of C resulting in a loss function that more heavily weights misclassification of structures.[174]. Testing the models across a range of C values ensured that any identified trends in the relative performance of the kernels were not merely fortuitous results impacted by an arbitrary selection of C. Training was performed using ‘balanced’ class-weights. This is a step designed to more reasonably train models on datasets with uneven class sizes by adjusting the misclassification penalty based upon the frequency of the class within the set- such that the disproportionate influence of the majority class upon training is reduced.[174] The linear kernel was chosen here because its simplicity makes it most appropriate for identifying relationships useful for choosing related synthesis constraints. Further, identifying these ‘simple’ relationships is a clearer approach to proving that the underlying kernel has picked up important and reasonable structural information and formed interpretable descriptors.

Figure 6.16 shows the **balanced** accuracy scores of a linear support vector machine in learning the intuitive class of a crystal structure from its machine-learned descriptor value(s) - derived from various kernels. The assessment was performed for each case, learning from:

1. A single machine-learned descriptor. The indicated results are that of the descriptor that resulted in the highest learning performance for each case - tested across the first 32 ML descriptors.
2. A combination of the five highest ranked ML descriptors. All combinations of up to 5 of the descriptors were tested and the indicated accuracy is that from the combination that resulted in the highest learning performance for each case.

Here, balanced accuracy is a learning performance measure designed to assess the overall model performance - accounting for uneven class sizes in the dataset. The balanced accuracy is the average of the accuracy of predictions for each class in the dataset. In the binary classification problems here, this is given simply by the mean of the true positive rate and the true negative rate:

$$\text{Balanced Accuracy} = \frac{1}{2} \left( \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (6.2)$$

where TP and TN are the numbers of true positive and true negative predictions (correctly identified samples of classes 1 and 0) respectively. FP and FN are the numbers of false positive and false negative predictions respectively [174].

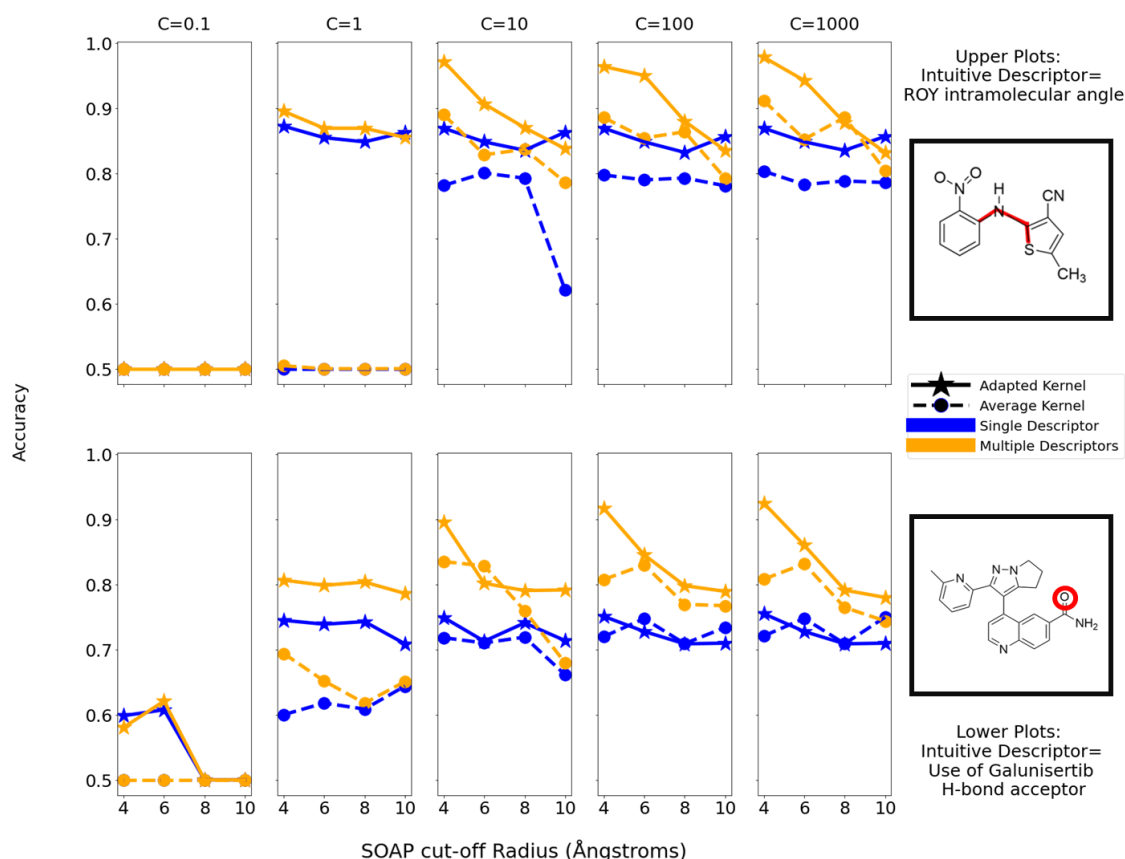


Figure 6.16: Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor from either single (blue) or multiple (yellow) ML descriptors as a function of the underlying SOAP cut-off radii. Line style denotes the type of kernel construction used to derive the ML descriptors.

These results suggest a general increase in balanced accuracy when learning from descriptors derived via the adapted kernel than from those derived via an average kernel. This is particularly clear for the ROY system - where the adapted kernel outperforms the average regardless of the SOAP cut-off or C-value used. The learning performance for this relationship is strong, with accuracies of  $\sim 0.8$  or even higher being achievable for both kernel types with higher C-values used for the model. This suggests that the ML descriptor-molecular conformation relationship for the ROY system is clear and strong.

However, for the galunisertib system the models' ability to learn the relationship between ML descriptors and hydrogen bonding varies less with kernel construction. At low C-values, where the model is trained with little penalty for misclassification, the adapted kernel significantly outperforms the average kernel. However, at higher C-values, the performance from the two kernel types when learning from a single ML-descriptor is largely-on par. The maximal achieved accu-

racy when learning from a single descriptor is  $< 0.8$ , suggesting that the ML-intuitive descriptor relationship in this case may be less clear. However, when the model is allowed to learn from multiple ML descriptors, much higher accuracy can be reached and the adapted kernel still outperforms the average in this regard.

In this work, the models' ability to learn relationships from multiple ML descriptors was investigated in order to test whether the structural information needed to predict the intuitive descriptor values may be distributed across multiple highly ranked components. However, it is important to note that this is only performed as a means to assess where the SOAP kernels and kPCA decomposition have been able to pick out meaningful features. In a true GCH workflow the ML descriptors would not be combined and used as a single descriptor for hull construction. However, the information gleaned is still useful. Here, the increased accuracy of the models when learning from multiple ML descriptors may suggest that there are several highly-ranked kPCA components with some character relating to the intuitive descriptor.

One interesting finding from the results in Figure 6.16 is that there appears to be little dependence of the learning performance upon SOAP cut-off, especially for the adapted kernel. Whilst initially this would appear to contrast with the qualitative findings in Section 6.4.2, this is not necessarily the case. Given the nature of the SVC model, it is possible for the relationship and the 'separation' of clusters to weaken, but for the optimal separating hyperplane - and corresponding model performance - to be largely unchanged.

A useful note with regard to the implementation of this work is that the data used in the SVC training was **not** scaled prior to training of the model. In most cases, it would be conventional and beneficial to scale the input data during pre-processing, such that all included variables cover the same range. This prevents the greater variance along one variable from lending said variable undue weight during training - impacting the results. However, in this application, that differing variance is instructive and the information should not be lost. Therefore, SVC training was performed using the unscaled data.

The work discussed in this section has compared the utility of the different kernel constructions in leading to meaningful ML descriptors. For the single descriptor relationships, this relied in each case upon taking the ML descriptor - of the first 32 - that most strongly related to an intuitive descriptor. The relationship strengths across these 32 explored descriptors in each key case was

then further investigated. The purposes of this investigation were two-fold:

1. To explore the distribution of interpretable ML descriptors across the kPCA decomposition
2. To gauge the level of SVC model accuracy that represents ‘stand out’ performance and denotes a strong ML-intuitive descriptor relationship

Figures 6.17 and 6.18 show the results of this work for the ROY and galunisertib systems respectively. For a subset of the previously investigated ML models (C-values 1,10 and 100 and SOAP cut-off radii 4 Å 6 Å and 8 Å) the balanced accuracy of SVC in predicting the intuitive classes from each of the first 32 ML descriptors is shown.

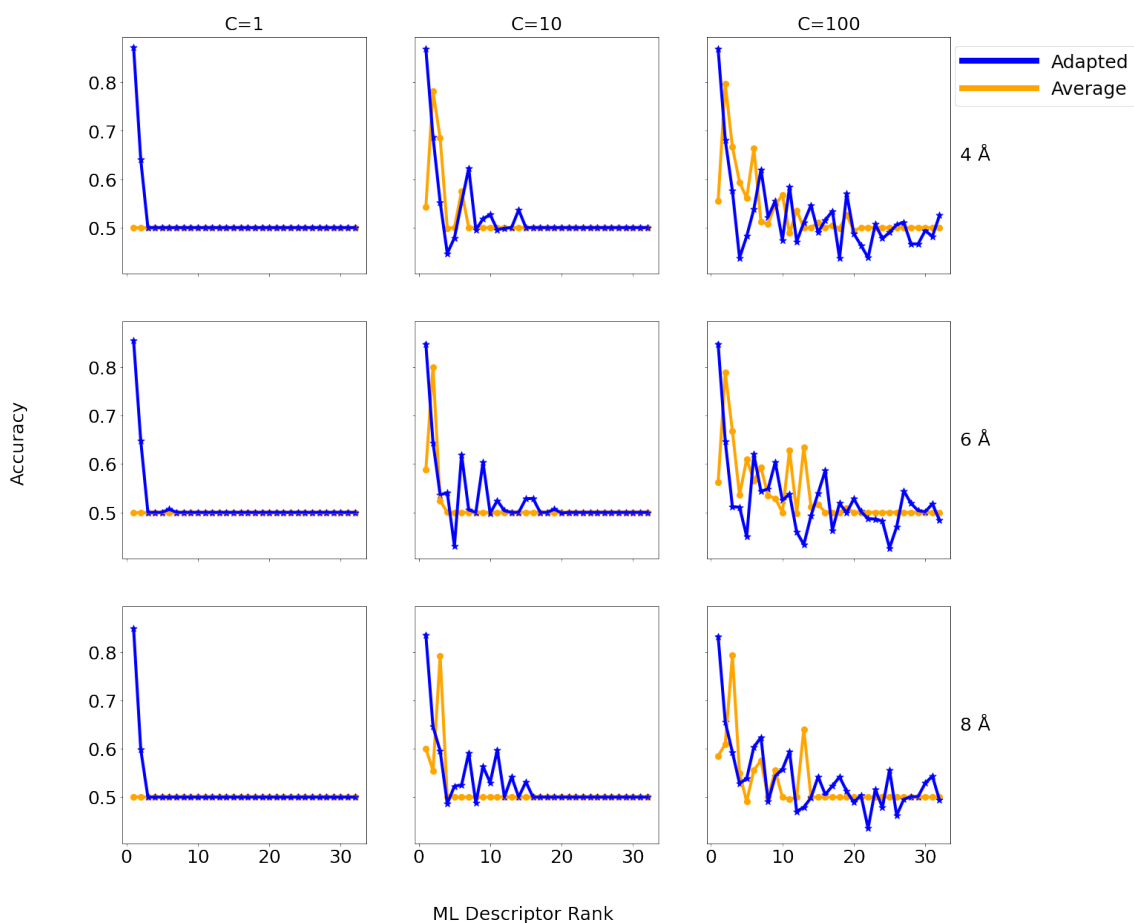


Figure 6.17: Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor of ROY structures from single ML descriptors as a function of the ranking -within the kPCA decomposition - of the ML descriptor. Line style and colour denotes the type of kernel construction used to derive the ML descriptors.

The above results for the ROY system are as expected for a case in which highly ranked derived ML descriptors have a clear and unambiguous relationship to the intuitive descriptor. Firstly, it can be seen that - regardless of model parameter C - the ‘best’ ML descriptors are those highly

ranked in the kPCA decomposition. This is a reassuring finding that affirms the use of the SOAP kernels and kPCA in identifying ML descriptors that both are interpretable and encompass a significant proportion of the structural variance of a prediction set. Further, the findings suggest that prior investigations - which have focussed upon only the highly-ranked ML descriptors - have solid grounding and are unlikely to have overlooked important descriptors. The findings for this relationship demonstrate that, for all kernel constructions, there can be found a highly-ranked ML descriptor for which the SVC model accuracy is significantly higher than for other components. These ML descriptors can be said to demonstrate stand-out performance, significantly above the ‘noise’ in the learning performance across other components. This, coupled with the high achieved accuracies, suggests that the identified relationship is indeed meaningful and significant.

However, the picture for galunisertib is more complex.

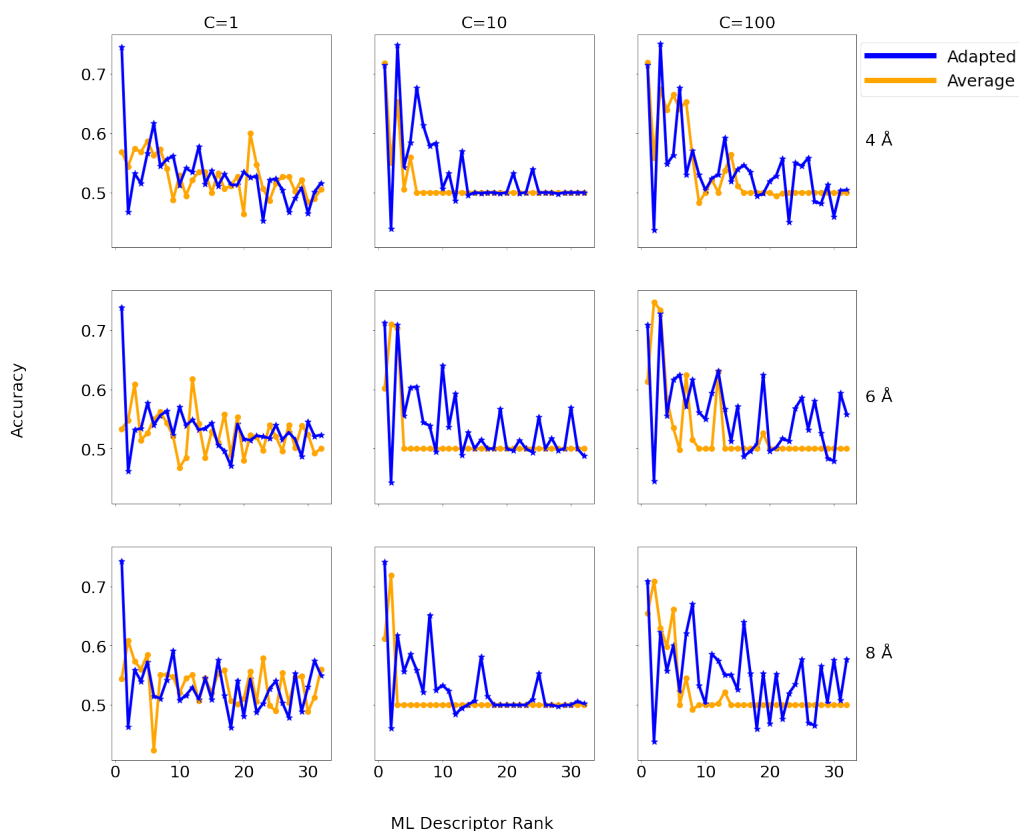


Figure 6.18: Plots showing the balanced accuracy of SVC models in learning the values of an intuitive descriptor of galunisertib structures from single ML descriptors as a function of the ranking -within the kPCA decomposition- of the ML descriptor. Line style and colour denotes the type of kernel construction used to derive the ML descriptors.

In almost all cases - particularly for the adapted kernel - it remains clear that the highly ranked ML descriptors are more meaningful than low-ranked descriptors - again supporting the focus of

investigations and of the GCH approach upon these descriptors. When  $C=1$ , the findings demonstrate a single stand-out descriptor for the adapted kernel, but largely noise-like performance from the average kernel - with no particular stand out descriptor and all accuracies remaining low. As  $C$  increases, however, it appears that for the adapted kernel many ML descriptors can then lead to accuracies approaching that of the most promising descriptor. The performance of the average kernel here improves and can rival that of the adapted kernel.

One possible explanation for these findings could be that there is indeed **significant** ‘hydrogen bonding character’ to many components derived from the adapted kernel of galunisertib. However, given the lower accuracies ( $\sim 0.6$ - $0.7$ ), it may also be that there are several components each **weakly** related to an aspect of the hydrogen-bonding or even that the model implementation is able to achieve accuracies in the region of  $0.6$ - $0.7$  in the absence of truly meaningful relationship. That is, the perceived accuracies could lie within the expected region of noise. Due to this consideration, it is possible that the model accuracies found for this relationship - particularly at high  $C$ -values - are not instructive and should be treated with appropriate caution.

#### 6.4.4 Summarising Investigation of Relationship Strengths

Investigation of the strength of proposed ML-intuitive descriptor relationships has confirmed that meaningful and instructive relationships to kPCA components have been identified for density in porous systems and molecular conformation in the ROY system. The response of these relationships to changing the underlying SOAP cut-offs affirmed the meaningful nature of the ML descriptors. The findings also demonstrated that the most ‘meaningful’ ML descriptors are likely to be those highly-ranked by the kPCA decomposition - a reassuring finding backing the GCH approach. Importantly, it was also seen that in these instances the adapted kernel generally led to ML descriptors more strongly related to the intuitive descriptors - particularly at short SOAP cut-offs.

The relationship between ML descriptors for the galunisertib system and hydrogen bonding motifs within the crystal structures was also further investigated. This relationship did demonstrate the same trends with respect to kernel comparison, but the lower model accuracies and poorly established connection between kPCA component ranking and SVC model accuracy suggest that this relationship is weaker than those seen for ROY and for the porous systems. A degree of caution may be needed concerning this relationship. Future work could investigate this further to clarify results.

## 6.5 Concluding Remarks

Work in this chapter explored the utility of different SOAP kernel constructions in the derivation of interpretable machine-learned structural descriptors. Several qualitative relationships were identified between ML descriptors and more intuitive conventional structural features, including relationships between ML descriptors of ROY crystal structures and the underlying molecular conformation and between ML descriptors of porous crystal structures and density. Identifying interpretations of the ML descriptors in this way provides insight into useful ways to explore CSP landscapes - through descriptors capturing much of the structural variance. It could also help to propose experimental constraints to be used in selective synthesising candidate structures identified by GCH methods. Further, the ability to ‘pick out’ meaningful structural information is a positive sign for the reasonable construction of the kernel.

Next, attempts were made to quantify the strength of the identified relationships - including visual analysis, calculation of  $R^2$  values, and use of supervised machine learning model accuracy as a proxy to relationship strength. This found that the underlying SOAP cut-off had an impact on relationship strength - and that this was in line with expectation based on whether the intuitive descriptor was a short or long range feature. Additionally, the work suggested that descriptors derived from the adapted kernel may be more interpretable than those derived from the average kernel.

The investigation of relationship strength also suggested that some identified relationships are clearer than others. One possibility for future work may be to further test and seek clarity on one of the weaker relationships - between ML descriptors of galunisertib crystal structures and hydrogen bonding. Other possibilities for future work could include searching for additional relationships and testing the performance of the ReMatch kernel in producing meaningful ML descriptors.



## Chapter 7

# Machine Learning of Energies

### 7.1 Overview

One useful application of a similarity kernel is in energy prediction. This can be used in a CSP workflow to re-optimize structures with accuracy approaching that of high levels of theory, but with greatly reduced cost. Gaussian Process Regression (See Section 2.5.3) has been employed in this way [31], with promising recent results, for example achieving lattice energy prediction errors  $< 1$  kJ/mol for crystal structures of pentacene [62]. It has also been used alongside a multi-fidelity learning approach to exploit correlations between lattice energies predicted at different levels of theory to further reduce the number of highest level energy calculations required, while increasing accuracy of the predicted energies [175].

This section compares the utility of the adapted and average SOAP kernels in energy prediction via GPR. This is not as a means to providing meaningful advancement in energy prediction, or to demonstrating the potential of GPR in chemical applications, which is well-established [31]. Rather, it is intended purely as a means by which to compare the impact of the kernel construction upon potential results, and to give possible insight into where each kernel could be best put to use in this context.

## 7.2 Approach

Using Gaussian Process Regression with a precomputed kernel via an adapted sklearn *GaussianProcessRegressor* class [176], relative total energies of crystal structures were predicted and the Mean Average Error (MAE) and root mean squared error (RMSE) of predictions was used as a metric to assess the utility of the average and adapted SOAP kernels in the GPR implementations for energy prediction in molecular crystals.

In the adapted GPR class, the kernel used to derive the prior distribution is the user-selected pre-computed kernel on the training set data. For the purposes of this work, the kernel of choice is the kernel under investigation at any given time. As such we perform separate prediction workflows to test the performance of different kernel types and underlying SOAP cut-off radii, implementing the respective kernels in each workflow.

In each workflow, the features that are used to describe each structure are the similarity of that structure to each representative of the training set. Thus, the training data used to derive the posterior is given by the similarity of the training set to itself - i.e the training set kernel itself. The test set feature data is given by the similarity of the test set member to each member of the training set. Both the kernel across the training set and the similarity of the test set to the training set are sub-matrices of the kernel across the full structure set. Practically, the similarity kernels for each system were computed across all the available structure data - and the sub-matrices comprising the training set-training set similarities and test set -training set similarities were extracted as required (Figure 7.1).

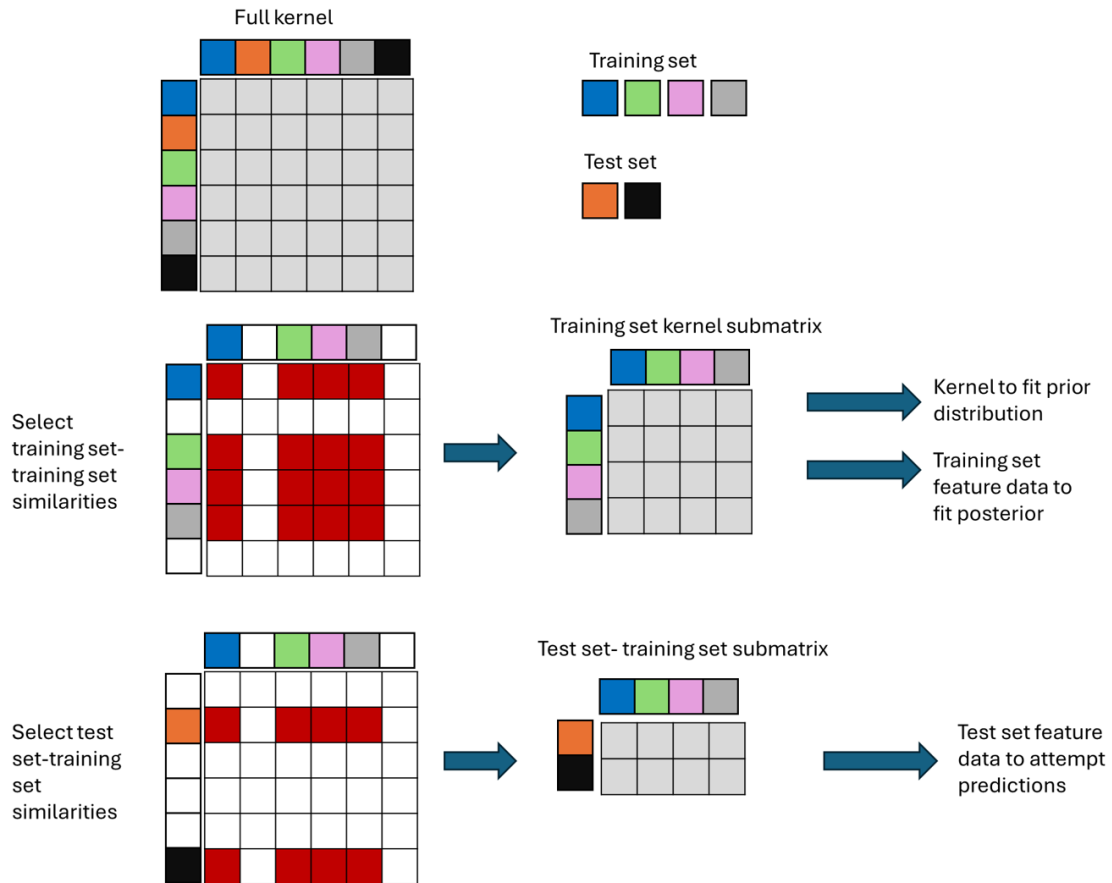


Figure 7.1: Process used to derive the kernel, training set feature data, and test set feature data for GPR workflows from the full structure-set kernel. Sub-matrices are extracted by selecting the corresponding rows and columns to obtain the training-set training set similarities (training set feature data) and test set - training set similarities (test set feature data). The training set - training set similarity submatrix is itself a complete similarity kernel across the training set and is also used in GPR as the kernel to fit the prior.

## 7.3 Initial Testing

### 7.3.1 Datasets

Initial GPR tests were run on small prediction sets for three systems: Targets XXXI and XXXII from the 7th CSP Blind-Test [17, 18], and chlorpropamide [177]. The Target XXXI and XXXII structure sets were those finalised by group 5 in the ranking stage of the blind test [18]. All sets were comprised of structures that had undergone full geometry re-optimisation at pDFT level (Table 7.1). Some structures found to be physically unreasonable were removed from the chlorpropamide set before commencing ML work.

System	Number of Structures	Final Energy Evaluation
Target XXXI	99	PBE-GD3BJ 600 eV cut-off
Target XXXII	495	PBE-GD3BJ 600 eV cut-off
Chlorpropamide	535	PBE-GD3BJ 600 eV cut-off

Table 7.1: The structure set sizes and method of final energy-evaluation for each system used in initial ML explanation. True energy-evaluation workflows are multi-step processes - the original citations should be consulted for this information

The initial average possibility kernel calculations assumed asymmetry of the molecule for each system. In the case of Target XXXII, it can be seen from the molecular diagrams (Figure 7.2) that the in-crystal molecular conformations must be asymmetrical. The possibility of adopting symmetrical conformations is precluded by the connectivity of the molecules. In the case of Target XXXI and chlorpropamide, the connectivity does not preclude symmetrical conformations. However, each in-crystal conformation present in the sets was tested to identify its symmetry - and all conformations were shown to be asymmetrical. These molecules - for example Target XXXII - may exhibit local symmetry, but recall that as yet, this is not considered in the kernel construction (see Section 3.5.3).

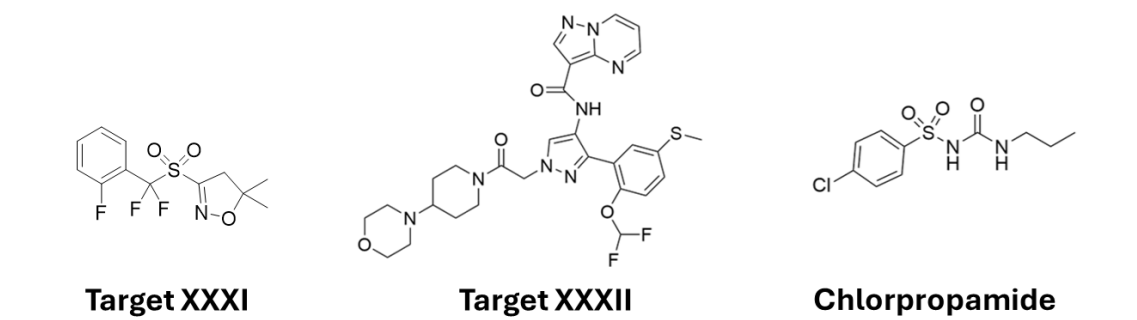


Figure 7.2: Molecular diagrams of each system used in initial ML exploration : Target XXXI, Target XXXII and Chlorpropamide

### 7.3.2 Training and Validation Process

The CSP sets were treated as complete - that is the global minimum was assumed correct. Therefore, the relative total energies of each structure were calculated as in Equation 7.1, by subtraction of the global minimum total energy from the given structure's total energy.

$$E_x^{Rel} = E_x - \min\{E_i | i \in \text{structure set}\} \quad (7.1)$$

The workflow (Figure 7.4) for training and validation used extensive cross-validation. Given the restrictions of the small datasets, testing upon many different training and test set selections was used in order to avoid results being unduly influenced by the arbitrary selection of structures for training and test sets. However, the need to maintain a reasonable size of test set despite the small structure sets prevented use of traditional cross validation with a large number of folds. Instead, the workflow called upon the use of sklearn [167] functionality *RepeatedK\_Folds*. Employing this functionality, a loop was performed in which training and test set folds were selected (with 80:20 ratio as in traditional 5-fold cross validation), training and validation is performed, the rows of the dataset are shuffled, and the process repeats. In this work, to ensure convergence of results over the number of repeats - the loop was conducted 100 times.

In order to test the impact of training set size, at each repetition of this loop smaller subsets of the declared training fold were selected - with varying sizes to test the impact of training set size. At each iteration of the loop these subsets of different sizes were sampled separately at random from the training fold. Therefore, at each iteration of the loop the differently sized subsets must draw from the same training set, and be tested upon the same test set - but can overlap (or conversely can be entirely distinct from) one another.

Then, at each iteration of the loop, and for each identified training subset, the model was fitted upon the training subset, and the predictions tested upon the test set. Learning performance was assessed on two metrics - RMSE and Mean Average Error (MAE) (Equation 7.2). Each of these metrics was averaged across all iterations of the loop - for each tested training set size, the mean and standard deviation of the corresponding set of measured errors was recorded.

$$MAE = \frac{\sum_i^N |E_x^{Pred} - E_x^{Rel}|}{N}$$

$$RMSE = \sqrt{\frac{\sum_i^N (E_x^{Pred} - E_x^{Rel})^2}{N}}$$
(7.2)

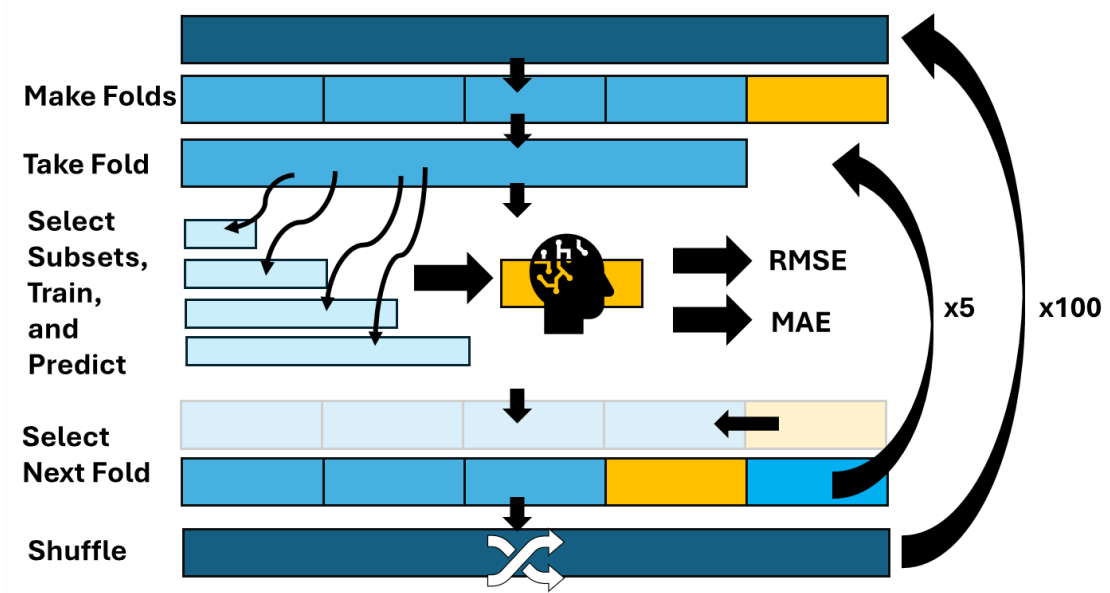


Figure 7.3: Workflow for selecting training and test sets during cross validation in initial ML exploration of each system. A repeated 5-fold cross validation is used, with shuffling of the dataset between each run. During testing on each test fold of each run, subsets of different sizes are randomly selected from the respective training fold to sample different training set sizes. The RMSE and MAE are evaluated for all trialled cases

### 7.3.3 Initial Results

The results of the initial ML tests are shown in Figure 7.4. The results shown are those of a single run of the workflow for each system - with random state variables enforced such that the results can be replicated if required.

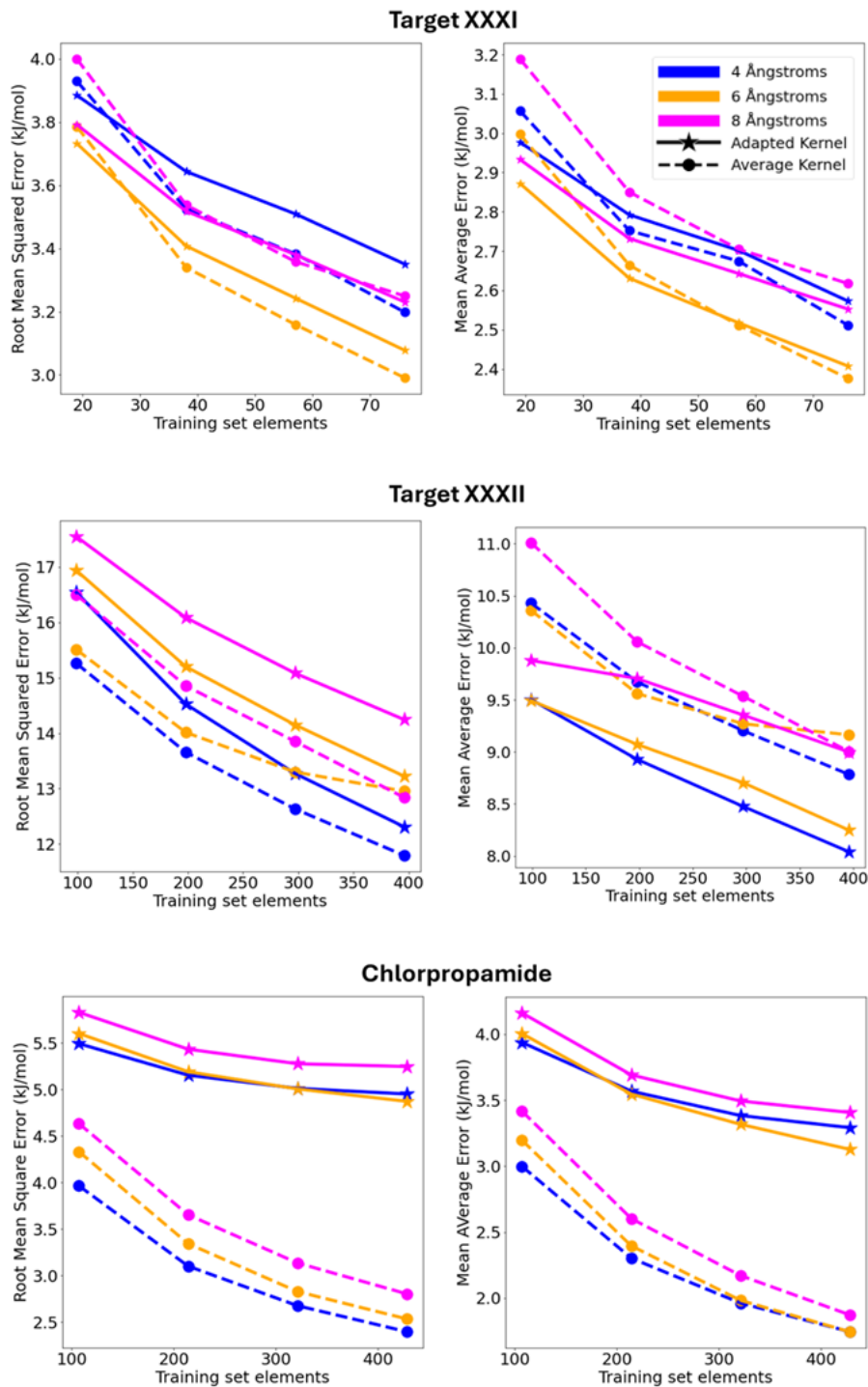


Figure 7.4: Average RMSE and MAE of energy predictions for each system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Different colours indicate different cut-off radii of the underlying SOAP descriptors used to derive the kernel.

The preliminary results for the cases of Targets XXXI and XXXII demonstrate similar learning performance of the adapted and average kernel GPR implementations. In the case of Target XXXII, the performance of all implementations is poor - with large errors far greater than the polymorph pair difference [37]. Target XXXI fares better, reaching mean average errors of 2.3 kJ/mol with a training set of just 80 structures. This is promising, however it is unclear whether this is due to successful learning of structure-energy relationships or, given the small training set size, is merely a chance finding.

The results should be read with caution due to the large standard deviations of the measured errors over the cross-validation. The severity of these large standard deviations can be seen in Figure 7.5, which shows the MAE of predictions for each case, using initial SOAP descriptors with 6 Å cut-off radii - with shading indicating the standard deviation of the errors.



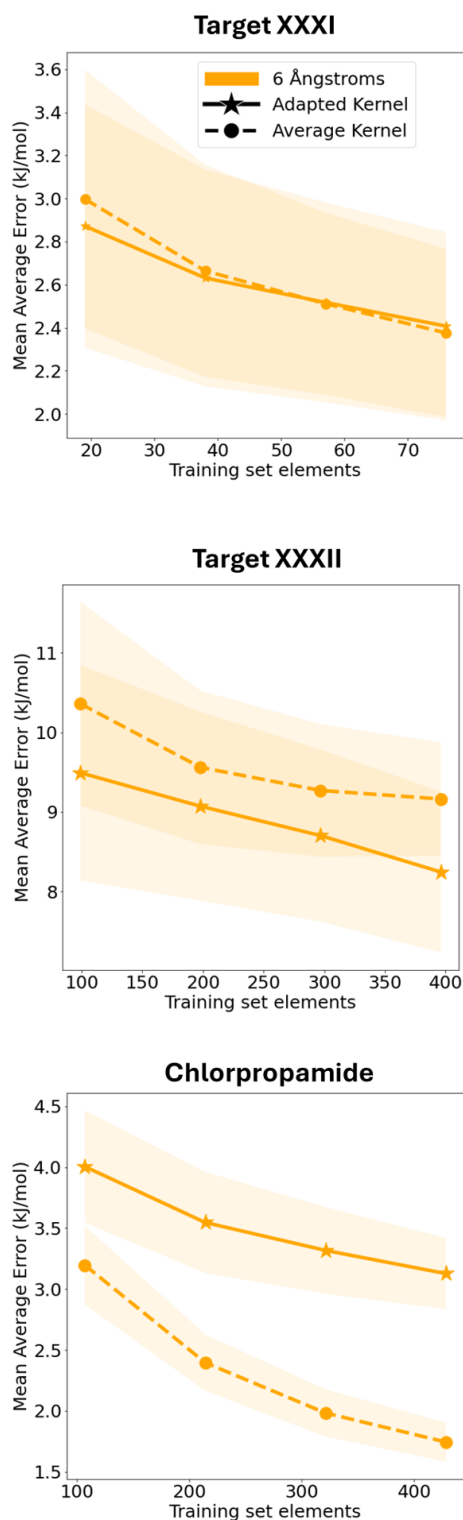


Figure 7.5: Average MAE of energy predictions for each system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Only cases with 6 Å cut-off radii of the underlying descriptor are shown, for simplicity. Shaded areas centred about each curve indicate a margin of uncertainty of the declared errors - with the width of the shading being equal to a single standard deviation either side of the curve.

A large standard deviation indicates a heavy dependence of the results upon the particular training set selection - suggesting that the quality of predictions may vary significantly. Further, it can be seen for Targets XXXI and XXXII that the performance gap between kernel implementations lies within the standard deviations of the errors - meaning that the relative performance of the implementations is inconclusive. The results arising when using different cut-off radii are not displayed here in order to ensure clarity of the figures. However, those results show similar behaviour, and there is further overlap between their respective errors. This suggests that larger training sets may be required in order to derive trustworthy comparisons between the kernel implementations.

The case of chlorpropamide, however, shows a clearer performance gap. The average kernel implementation outperforms the adapted kernel, regardless of underlying cut-off radii. Further, the average kernel implementation achieves promising mean average errors of  $< 2$  kJ/mol with just 400 structures in the training set (Figure 7.4). The standard deviations shown in Figure 7.5 indicate that this distinction may be reliable. It was unclear why this case displays a greater performance gap, but brief investigation is given in Section 7.3.4.

#### **7.3.4 Investigating Chlorpropamide**

The discussed initial results showed a significant performance gap between the compared GPR implementations in the case of the chlorpropamide system, with the adapted kernel implementation falling short.

Somewhat serendipitous investigation suggested that the poor performance of the adapted kernel implementation may be related to the inclusion of a small number of  $Z'=2$  crystal structures in the structure set used in Section 7.3.3. A test found that the learning performance, particularly for the adapted kernel, greatly improved upon removal of these structures, and indeed the performance gap between implementations appeared to close.

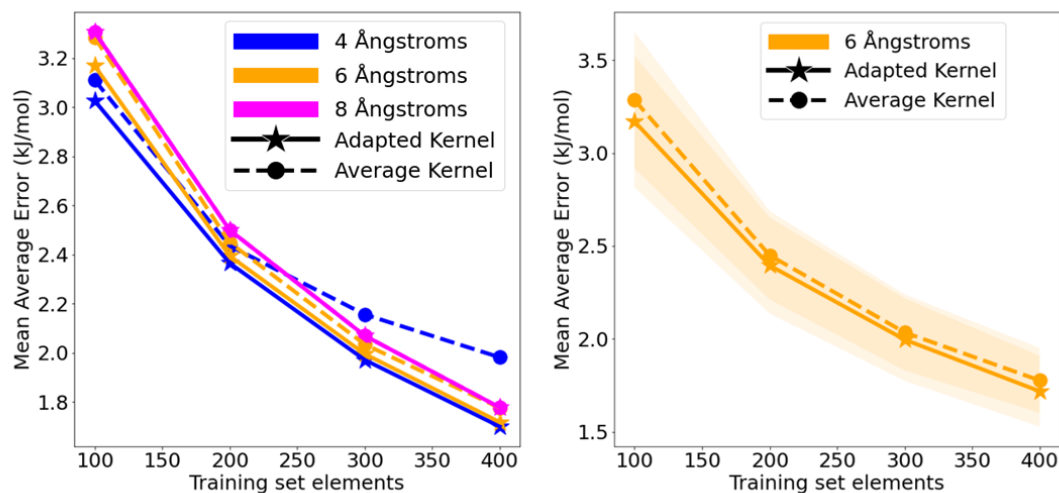


Figure 7.6: Training curves (left) and display of uncertainty of measured errors (right) - analogous to the results in Figures 7.4 and 7.5 respectively - for the  $Z'=1$  subset of the chlorpropamide landscape.

Due to time constraints, the cause of this poor performance on inclusion of  $Z'=2$  crystal structures was not investigated. However, if the results can be corroborated and replicated across other systems, it may be worthwhile to investigate this phenomenon. A possible aspect to explore would be the impact of different approaches to defining analogous atoms in the case of pairs of crystal structures with  $Z' > 1$  (See Section 3.3.2).

## 7.4 Extended Testing

### 7.4.1 Overview

The discussed ML results indicated a need for investigation of learning performance when implementing larger training sets. This was needed due to the unreliability of results from the small training sets in many cases and to explore whether greater testing could uncover a distinction in prediction errors between adapted and average kernel GPR.

The following section explores the results of training GPR models for energy prediction upon a larger training set, and the corresponding comparison of implementations using the adapted and average SOAP kernels. Unfortunately, due to time and resource constraints, research into Targets XXXI and XXXII was not extended. However, energy prediction in the case of chlorpropamide is extended and discussed in detail.

### 7.4.2 Dataset

A large dataset of predicted chlorpropamide crystal structures was obtained from the literature. The available dataset contained 16650  $Z'=1$  crystal structures predicted using quasi-random sampling alongside a bespoke approach that incorporated machine-learned strain energies into both sampling, and energy minimisation. Structures were then optimised using empirical force fields accompanied by distributed atomic multipoles [177].

In order to obtain total energies at the pDFT level of theory (PBE-GD3BJ/500 eV cut-off), single point energy calculations were performed for each structure - implemented in VASP. Some calculations failed due to reported PRICEL errors:

```
Number of cells and number of vectors did not agree
```

or other symmetry precision errors. PRICEL errors appear to be the predominant reason for failed calculations, although a small proportion of calculations failed to converge in reasonable time with unknown cause. It is possible, though not verified, that failures were related to an issue of physically unreasonable structures discussed in Section 7.4.3. On the basis of time constraints, and the minimal fraction of structures exhibiting calculation issues, the problems were not resolved and instead structures for which single-point energy calculations failed or did not finish in reasonable time were rejected and replaced in their respective training and test sets by randomly selected structures.

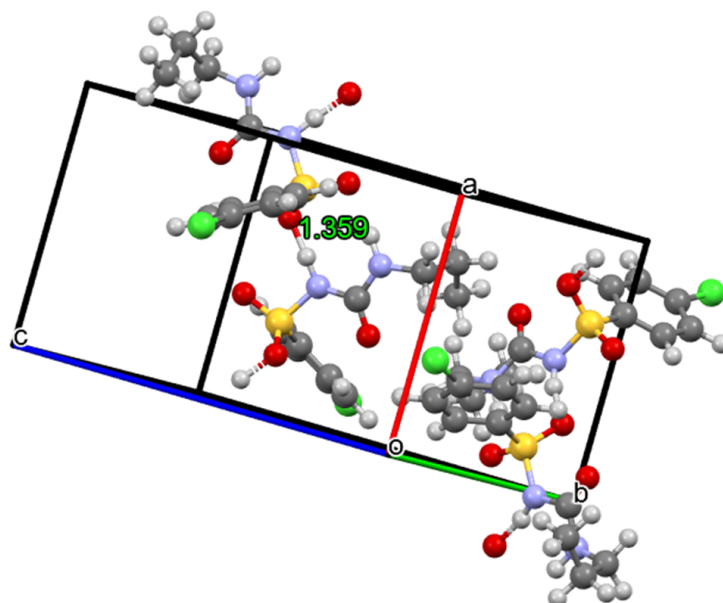


Figure 7.7: Example of physically unrealistic predicted chlorpropamide structure found in the set. The measured intermolecular oxygen-hydrogen atomic separation ( $1.359 \text{ \AA}$ ) is unreasonably short - leading to formation of a bond between molecules in the visualisation

Similarly to the smaller set used in Section 7.3.3, the set of structures contains only asymmetrical molecular conformations. There are four in-crystal molecular conformations present in the set, all of which are asymmetrical [177]. As such, the average possibilities kernel again assumed asymmetry of the molecule and compared atoms between crystal structures only where the atoms shared the same indexing within the molecule.

### 7.4.3 Pre-Processing

Predicted structure sets, especially those obtained from the literature where the user may not be able to test and verify the methods, risk containing flawed or unphysical structures. In the case of chlorpropamide, some structures in the published set were identified as having unphysically close intermolecular oxygen-hydrogen contacts - that lead to ‘conjoined molecules’ in the visualisation in some instances (Figure 7.7). As the set could not be manually/reliably searched to exclude all such cases, an energy cut-off was applied, intended to exclude all physically unrealistic structures whilst retaining valid structures across a large total energy range. Physically unrealistic structures should result in calculated DFT total energies that represent outliers from the normal energy range. From a set of 5000 single-point energy calculations, outliers were identified via the Inter Quartile

Range (IQR) criterion:

$$x \in \text{outliers} \iff E_x > E_{cut} \quad (7.3)$$

$$E_{cut} = Q3 + 1.5 \times IQR$$

This resulted in an  $E_{cut}$  value of -17651.277 kJ/mol, which was assumed to remain applicable to the extended chlorpropamide data set. As such all structures with total energy  $> -17651.277$  kJ/mol were excluded and replaced in their respective training/test set by a randomly selected structure meeting the criterion for retention.

Based upon the original 5000 structures used to derive the retention criterion, the excluded structures correspond to 2.3% of the successful single-point energy calculations (Figure 7.8).

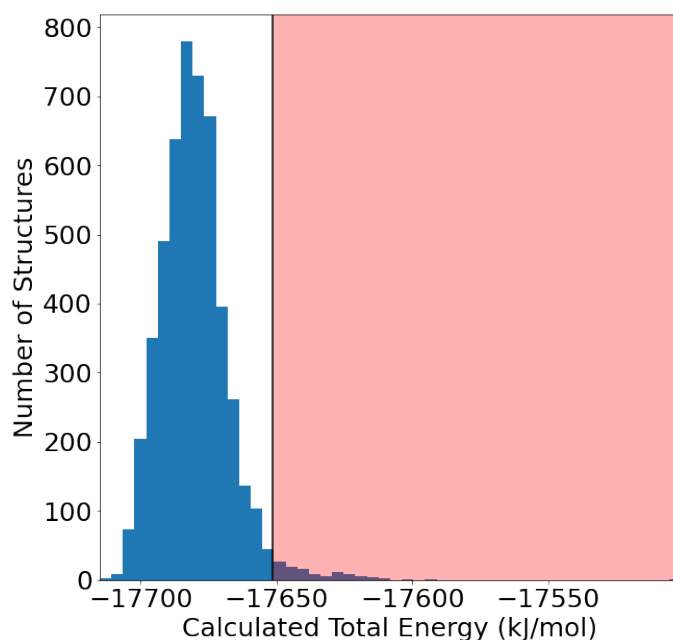


Figure 7.8: Histogram displaying the distribution of calculated total energies for the first 5000 chlorpropamide crystal structures tested. The red shading indicates the region declared unrealistic - determined by use of an interquartile range criterion on the set of energies - and any structure whose energies is calculated to lie within this region is rejected.

It is sufficient here to retain the original kernels calculated across the whole structure set - and simply extract the required matrix entries for training and testing as described in Section 7.2 without correction. This is because each unnormalised structure similarity  $K(A,B)$  concerns only the structures  $A$  and  $B$ , without influence from other structures. Further, recall that the normalisation of the kernel is given by:

$$K(A,B)_{norm} = \frac{K(A,B)}{\sqrt{K(A,A)K(B,B)}}$$

As such the kernel value assigned to an  $A, B$  structure pair can only be impacted by the presence of unphysical structures in the set if at least one of either  $A$  or  $B$  is itself unphysical - in which instance all impacted kernel values ( $K(A, x)$  or  $K(B, x) | x \in \text{structure set}$ ) will be excluded from the training and test sets, as well as the *implemented* kernel - due to the nature of the extraction of data. The remaining similarity data must therefore be unaffected by unphysical structures that were present in the set at time of kernel calculation.

#### 7.4.4 Training and Validation Process

The initial process aimed to determine the required training set size for reasonable predictions. To identify the ideal training set sizes, the model was trained on blocks of 500 randomly selected structures at a time with blocks being successively added to the prior training set and the model being retrained. In order to obtain results against a suitably large test set at each stage - including early stages of training when few final energy calculations were available, a consistent test set of 2000 structures was produced. The training set at each stage was evaluated once against this test set and the RMSE and MAE of energy predictions on the test set were recorded.

A cross-validation workflow was not applied at this stage due to the complexities of implementing such an approach consistently when performing the required single-point energy calculations progressively. Whilst this may introduce an uncertainty to the RMSE and MAE measured at each stage, it was assumed that this would not reverse the qualitative trends - particularly when using larger training sets and a large and varied test set. As such it was deemed that cross-validation was not necessary for determining the required training set size.

The test set and training set selections represented a random sample, well-distributed across the forcefield energy landscape (Figure 7.9) and so were assumed to provide a suitable exploration of the learning performance for this work.

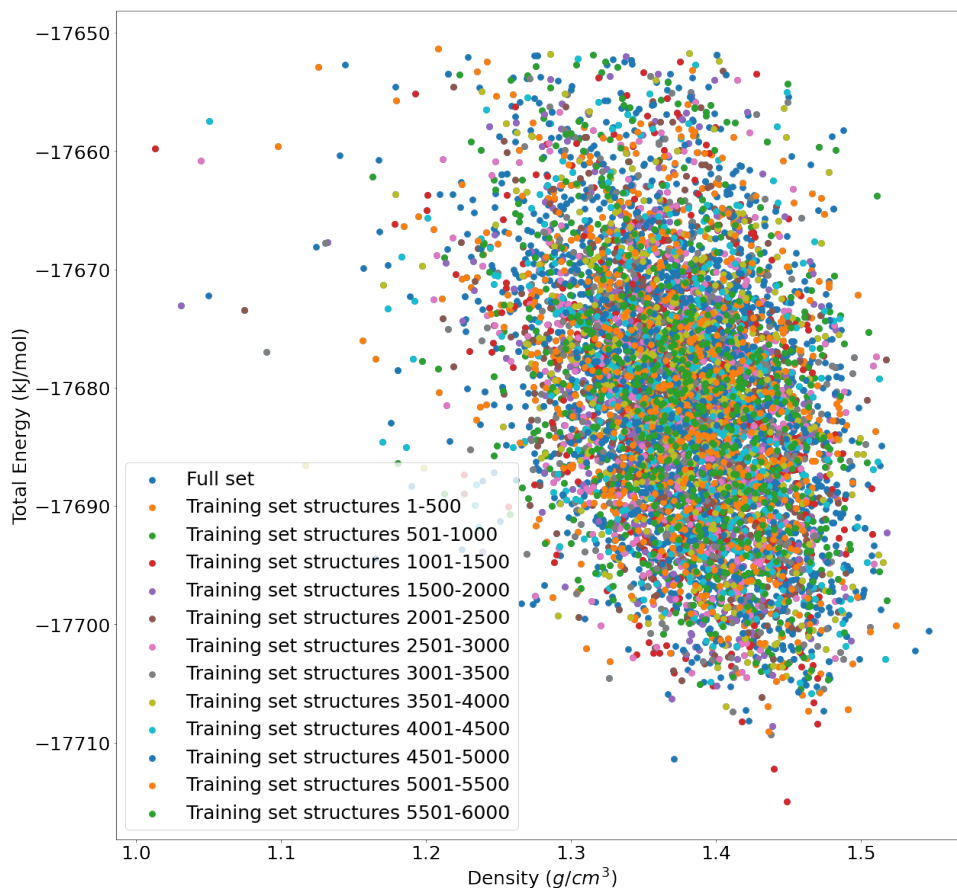


Figure 7.9: Scatterplot demonstrating the random spread of selected training and test set structures across the landscape. Marker colour denotes the set for which the structure was selected. The ‘Full set’ displayed does not include structures whose energy was considered unreasonable or that failed energy-calculations.

As the energy calculations and training were performed iteratively, there could be no definitively reliable global minimum energy by which to determine the relative energies of each structure. As such, a workflow was adopted in which the total energies were set relative to the minimum energy within the training set:

$$E_x^{Rel} = E_x - \min\{E_i | i \in \text{training set}\} \quad (7.4)$$

Such an approach represents a workflow more realistic to the real-world implementation of energy prediction - in which the final global minimum energy would similarly be unknown - but predictions could be determined based upon the available data and updated accordingly as a new ‘current minimum’ became available. In order to remain faithful to the approach as it could be implemented, and to prevent train-test set leakage, the same parameters are used to define the target data of the test set - i.e. the test set energies were also calculated relative to the minimum energy of the training set.



### 7.4.5 Results

Following the procedure outlined above, training was conducted and the MAE and RMSE of energy-prediction on the test set recorded, until reaching a large training set size of 6000 structures.

The training curves quickly demonstrated improved performance when using GPR implementing the adapted kernel than when implementing the average kernel (See Figure 7.10 and Table 7.2).

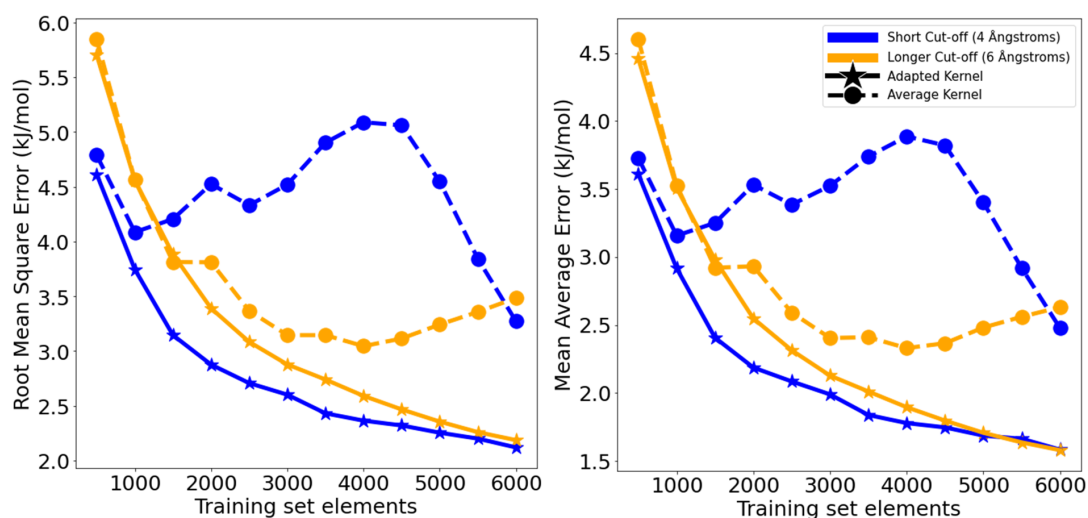


Figure 7.10: Average RMSE and MAE of energy predictions for the extended case of the chlorpropamide system as a function of training set size. Line and marker style denote the kernel construction used in the GPR model. Different colours indicate different cut-off radii of the underlying SOAP descriptors used to derive the kernel.

Training Set Size	Kernel Type	4 Å cut-off		6 Å cut-off	
		RMSE (kJ/mol)	MAE (kJ/mol)	RMSE (kJ/mol)	MAE (kJ/mol)
500	Adapted	4.614	3.609	5.702	4.459
	Average	4.789	3.727	5.849	4.602
1000	Adapted	3.746	2.917	4.560	3.506
	Average	4.087	3.158	4.564	3.527
1500	Adapted	3.146	2.407	3.885	2.981
	Average	4.206	3.251	3.813	2.921
2000	Adapted	2.877	2.188	3.391	2.549
	Average	4.525	3.531	3.811	2.932
2500	Adapted	2.708	2.086	3.084	2.315
	Average	4.332	3.385	3.368	2.592
3000	Adapted	2.602	1.990	2.878	2.128
	Average	4.521	3.524	3.146	2.405
3500	Adapted	2.432	1.840	2.740	2.011
	Average	4.903	3.739	3.146	2.410
4000	Adapted	2.365	1.779	2.591	1.896
	Average	5.090	3.887	3.045	2.330
4500	Adapted	2.321	1.748	2.467	1.798
	Average	5.061	3.820	3.115	2.367
5000	Adapted	2.255	1.687	2.356	1.708
	Average	4.547	3.401	3.242	2.482
5500	Adapted	2.201	1.664	2.260	1.637
	Average	3.843	2.919	3.358	2.561
6000	Adapted	2.120	1.584	2.187	1.580
	Average	3.272	2.477	3.485	2.632

Table 7.2: Calculated RMSE and MAE of energy predictions for the extended case of chlorpropamide - for each tested training set size, underlying descriptor cut-off radius, and kernel construction used in the GPR model.

The ‘erratic’ learning performance when implementing the average kernel suggested that the corresponding training curves may not converge. As such, it was deemed that further investigation via additional single-point calculations and model training would not justify the computational cost. Meanwhile, the smooth training curves corresponding to the adapted kernel had already reached reasonable mean average errors with training set sizes of 5000. Therefore, it was concluded that sufficient training had been performed and that, as a minimum, the qualitative trends would not change beyond this point. Whilst it could be deemed that the learning performance of the 4 Å cut-off average kernel GPR would be improved with further training, it appears likely that extended training would have worsened performance for the 6 Å cut-off average kernel GPR. As such, there could not be one fair cut-off for the comparisons. Therefore, in light of both the unpredictable performance and the constraints of research time and computational cost, the training was considered sufficient for comparison and discussion of results in this thesis.

Following from this, for true comparison of results - a cross-validation approach was applied to ensure that the results found were not dependent upon the arbitrary selection of test and training sets. A simple 5-fold cross validation was applied - using all of the available reasonable data. The available data comprised 8000 structures - corresponding to training sets of 6400 structures and test sets of 1600 structures. The list of structural IDs - from which folds were selected to facilitate training and test set formation - were shuffled prior to the selection of folds.

As before, the model was retrained on each training set-fold, with the energies to be learned being set relative to the training set minimum energy, and the RMSE and MAE of predictions on the test set fold was recorded. The mean and standard deviation of the recorded RMSE and MAE values over the cross-validation was recorded. This yielded the following results seen in Table 7.3.

Kernel Type	Cut-off (Å)	RMSE (kJ/mol)	MAE (kJ/mol)
Average	4	$3.056 \pm 0.073$	$2.309 \pm 0.050$
	6	$3.659 \pm 0.138$	$2.764 \pm 0.133$
Adapted	4	$2.168 \pm 0.036$	$1.638 \pm 0.032$
	6	$2.123 \pm 0.034$	$1.541 \pm 0.054$

Table 7.3: Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon the entire set of 8000 structures for which pDFT single point energies were calculated. These results are shown for each tested underlying descriptor cut-off radius and kernel construction used in the GPR model

This shows a clear improvement in utility in energy prediction of the adapted kernel over the average kernel - at both 4 and 6 Å underlying cut-off radii. By consideration of the standard deviation of the cross validation as a measure of the uncertainty of the results, the gap in learning performance between the adapted and average kernel implementations appears significant - with the discrepancy being much greater than the single standard deviations on the calculated errors across folds (Figure 7.11).

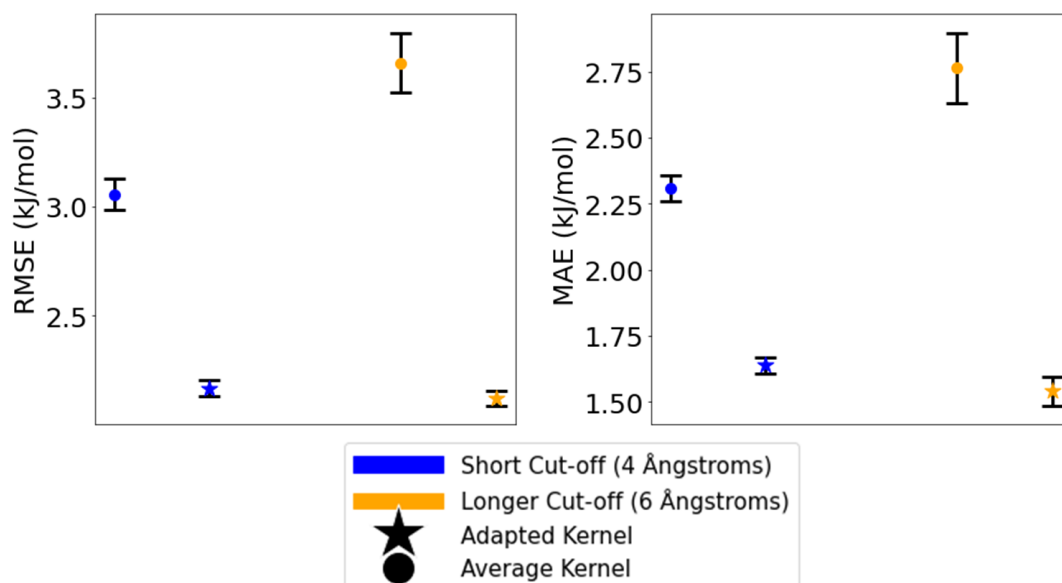


Figure 7.11: Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon the entire set of 8000 structures for which pDFT single point energies were calculated. Colour denotes the underlying descriptor cut-off radius and marker style indicates the kernel construction used in the GPR model. The error bars about each point display the uncertainty in the declared errors, with bars being of height (either side of the centre point) equal to one standard deviation of errors measured in cross validation. This displays the significance of the performance gap, with error bars arising from respective adapted and average kernel GPR implementations not overlapping.

Further, the results indicate that the adapted kernel GPR implementation can reach MAE values approaching 1.5 kJ/mol - and can achieve errors of less than 2.0 kJ/mol with 3500 structures in the training set. This prediction error is competitive with the polymorph pair difference [37]. The required training set size however is still too large as to be efficient for real world applications. The most important and promising regions of the CSP landscape - for which high-level energy calculations are required, may often contain fewer than 3500 structures, and so use of ML would

not offer practical benefit over explicit calculation. Worse so than the adapted kernel, the average kernel GPR implementation has not reached satisfactory learning performance even when using a training set of 6400 structures. This suggests that while neither implementation would be ideal for energy-prediction in this instance, the adapted kernel has shown initial promise over the average kernel in energy prediction.

However, another key difference between this work and real-world implementations - in addition to the unfeasibly large training sets used - is that this work aimed to sample reasonable structures across the **whole** available structure-energy landscape. By contrast, real-world implementations of energy prediction often focus upon a low-energy region of the landscape. Some preliminary investigation into learning performance under such conditions was performed. To do this, subsets of the full 8000 structures were selected, by extracting structures within varying energy windows. Five-fold cross validation was performed in each case, and the average RMSE, and MAE - as well as the corresponding standard deviations, were recorded. Table 7.4 shows the energy windows investigated, the corresponding training/test set sizes, and the learning performance for each kernel implementation.

Energy Window (kJ/mol)	Train Set Size	Kernel Type	4 Å cut-off		6 Å cut-off	
			RMSE (kJ/mol)	MAE (kJ/mol)	RMSE (kJ/mol)	MAE (kJ/mol)
50	5950	Average	$3.213 \pm 0.062$	$2.529 \pm 0.045$	$3.336 \pm 0.102$	$2.512 \pm 0.070$
		Adapted	$2.075 \pm 0.014$	$1.567 \pm 0.008$	$2.013 \pm 0.023$	$1.484 \pm 0.018$
40	4648	Average	$4.169 \pm 0.173$	$3.149 \pm 0.146$	$2.784 \pm 0.108$	$2.072 \pm 0.101$
		Adapted	$2.00 \pm 0.039$	$1.505 \pm 0.030$	$2.023 \pm 0.041$	$1.490 \pm 0.031$
30	2312	Average	$3.056 \pm 0.121$	$2.274 \pm 0.073$	$2.393 \pm 0.054$	$1.780 \pm 0.046$
		Adapted	$1.940 \pm 0.034$	$1.459 \pm 0.0184$	$2.069 \pm 0.089$	$1.509 \pm 0.047$
20	659	Average	$2.396 \pm 0.059$	$1.759 \pm 0.047$	$2.246 \pm 0.128$	$1.694 \pm 0.098$
		Adapted	$1.980 \pm 0.246$	$1.476 \pm 0.184$	$2.015 \pm 0.186$	$1.507 \pm 0.119$

Table 7.4: Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon given low-energy subsets of the entire set of 8000 structures for which pDFT single point energies were calculated. These results are shown for each tested underlying descriptor cut-off radius and kernel construction used in the GPR model.

These findings suggest that energy prediction with reasonable errors may be possible for both

implementations, with feasibly small training sets, if within a limited domain. This finding is in line with the smaller errors noted in the initial testing over small training sets in Sections 7.3.3 and 7.3.4. Under these circumstances, the adapted kernel continues to display an advantage over the average kernel - though the performance gap is significantly narrower. Figure 7.12 visualises the mean average errors in each of these windows, alongside the single standard deviations of the cross validation - demonstrating that while the performance gap narrows as the domain is limited to structures of lower relative energy, it remains significant in most cases. The distinction appears reliable in all cases except for implementations using a 6 Å SOAP cut-off radii and exploring crystal structures within an energy range of just 20 kJ/mol.

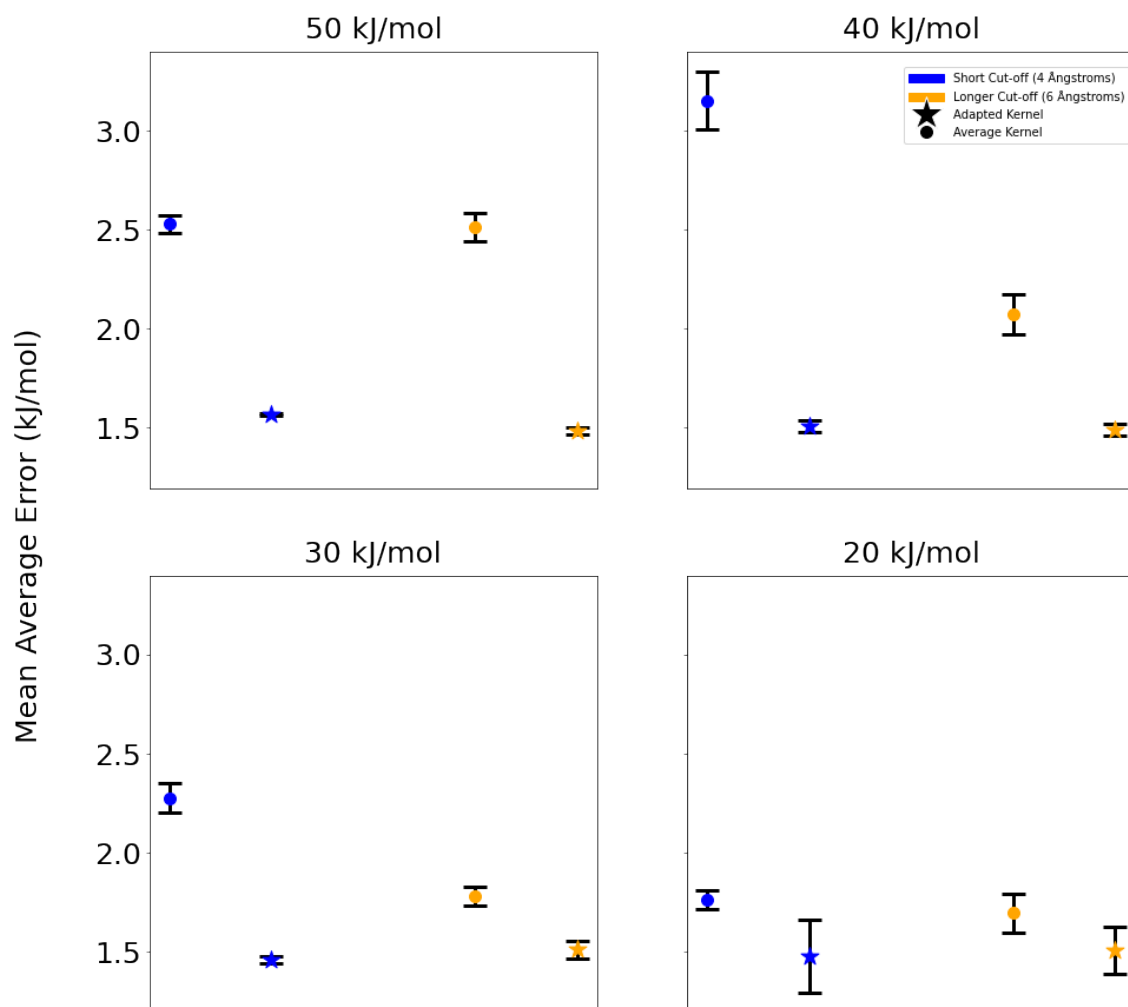


Figure 7.12: Average RMSE and MAE values of energy prediction on the extended chlorpropamide set, measured via cross-validation upon given low-energy subsets of the entire set of 8000 structures for which pDFT single point energies were calculated. Colour denotes the underlying descriptor cut-off radius and marker style indicates the kernel construction used in the GPR model. The error bars about each point display the uncertainty in the declared errors, with bars being of height (either side of the centre point) equal to one standard deviation of errors measured in cross validation.

These results are in line with the qualitative findings from the initial testing of machine learning on the  $Z'=1$  subset of chlorpropamide in Section 7.3.4, and advance upon those results - demonstrating additional benefits to the adapted kernel implementation when using larger structure sets or those with a wider energy range. However, as the larger dataset consisted entirely of  $Z'=1$  crystal structures, the extended testing can neither confirm or counter the unexpectedly poor performance of the adapted kernel implementation upon the mixed  $Z'$  structure set in Section 7.3.3 - which may still warrant further investigation.

There are other factors that should be considered in further development of this work. With sufficient resources, future work could aim to extend the training and establish whether the learning performance in the average kernel case can converge and/or reach sufficiently low error when given larger training sets. Additionally, work could explore the calculation failures and unfeasible structures to verify whether or not there are given structure types that dominate in these cases. If so, it could have lead to the model being tested and validated on a structure set not fully representative of the structural diversity of the wider set. Further work could also explore extended testing on other systems or learning of other energy metrics such as lattice energy. A more realistic real-world implementation of ML for energy prediction could be to employ  $\Delta$ -learning strategies, which learn a correction to a calculated energy rather than directly learning the energy itself. This can achieve greater accuracy [31] and often require smaller training sets. The use of GPR using the adapted kernel for  $\Delta$ -learning could be explored.



## 7.5 Concluding Remarks

Work explored the utility of the adapted and average SOAP kernel constructions in total energy prediction via Gaussian Process Regression. Initial testing was performed on predicted crystal structure sets of three systems, Targets XXXI and XXXII from the most recent CSP blind test [17, 18] and chlorpropamide [177]. Large standard deviations on measured errors and poor learning performance in some cases suggested that the initial exploration was insufficient and that training and validation on larger sets was required. Extended testing was then performed, attempting energy prediction on a larger dataset of chlorpropamide structures - for which pDFT single-point energy calculations had been performed. The extended results showed a notable advantage of the adapted kernel over the average kernel, with training curves converging faster and achieving strong learning performance. However, this still required training sets of 3500-6400 structures. Some testing indicated that satisfactory learning performance may be reached with smaller training sets when predicting the energies of only structures in the low energy region of the landscape - and the adapted kernel implementation still outperformed the average kernel implementation in these cases. Combining insights from ML investigations on the low energy landscape, and across the entire CSP landscape, it is possible that the adapted kernel implementation may better handle prediction over a wide energy range.

Future work could expand upon these results, extending the training to more varied systems or measures of energy or exploring  $\Delta$ -learning approaches. Work could also investigate issues with the underlying structure set that led to required pre-processing work and rejection of structures, and investigate the source of the poor learning performance of the adapted kernel when encountering  $Z'=2$  crystal structures.

## Chapter 8

# Templating CSP for Similar Molecules

### 8.1 Overview

As discussed in Section 1.2.1, the default workflow for most CSP methods revolves around generation and optimisation of trial crystal structures. The process can be costly as structure generation often relies upon broad sampling. This results in large numbers of required crystal geometry optimisations in order to predict a comparatively small number of low-energy local minima on the CSP landscape - often alongside many unfeasible high-energy local minima.

Attempts to sample structure space more efficiently are therefore desirable. One possible approach may be to use guided sampling. The inorganic CSP community has often used structural analogues to predict crystal structures [178]. However, this possibility appears to be underexplored for molecular CSP. The use of one predicted landscape to seed another has not been extensively investigated. That said, work has explored molecular crystal structure prediction via forming analogues of known experimental structures. One study predicted crystal structures via analogues of experimental structures of similarly-shaped molecules extracted from the CSD, successfully predicting the known crystal structures for 89% of molecules for which prediction was attempted [53]. One proposed model of crystal packing also provides hope for such approaches. The model conceptualises crystal structures as packing ‘boxes’ containing the underlying molecules, thereby restricting the number of possible conceived packing patterns. Exploration of known crystal structures found that most could be well described by one of the simplified packing patterns and that the unit cell dimensions were linked to the molecular dimension [179], thereby suggesting that similarly shaped molecules may adopt similar crystal structures. Further, analogous crystal structures of similar molecules have been found via analysis of the respective CSP landscapes. For example, discovering promising new porous structures via searching for analogues of a known porous ma-

terial [12].

Motivated by the promise of previous theory and work on analogous crystal structures, work in this chapter aimed to investigate the possibility of performing analogy-based molecular CSP, but beginning from previously predicted structures rather than from known experimental data. An initial proof-of-concept of analogy-based CSP, here called templating CSP, was developed - in which the sampling step of a traditional molecular CSP workflow was replaced with the generation of trial structures via formation of structural analogues of previously predicted structures (Figure 8.1). This was developed with a view to creation of a ‘fast-CSP’ method requiring less extensive sampling, and the potential of the approach was evaluated with respect to both its success and its efficiency.

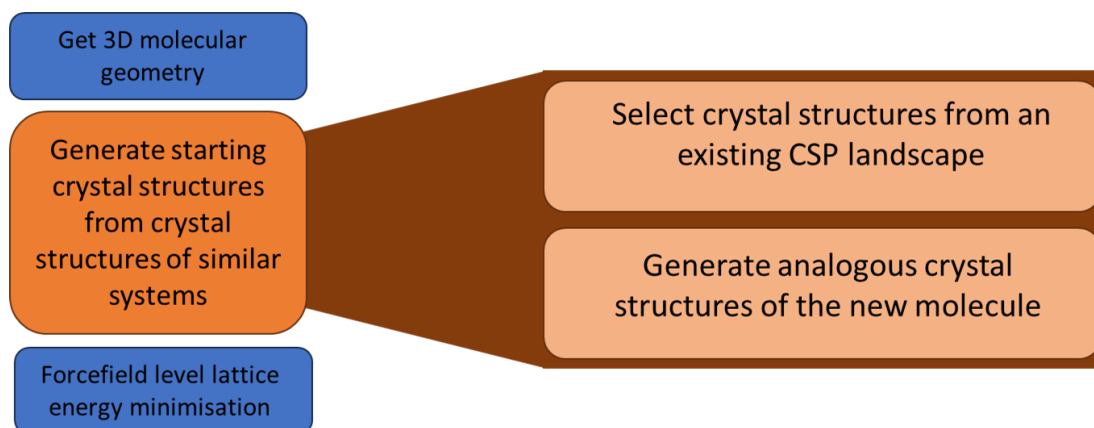


Figure 8.1: A simple CSP workflow, in which the trial structure generation stage (orange) is performed via templating - which requires gathering of previously predicted crystal structures, followed by creation of structural analogues containing a new molecule

Further, the potential of the GCH in helping to guide the approach is investigated. Work on the GCH has suggested that it may prove effective in identifying structures for which there are stable structural analogues of similar molecules [1]. The GCH could therefore be used to identify initial structures from which to begin templating CSP. The performance of templating CSP when guided by use of the GCH was explored, and compared to a templating CSP approach guided by lattice energy alone.

## 8.2 Key Definitions

It is important here to define some terms that will be used in this chapter to describe the work. These terms are defined in Table 8.1

Term	Definition
<b>Templating</b>	The insertion of new/different molecules into a previously defined crystal structure - in place of the old molecules
<b>Templating CSP</b>	The process of performing CSP using templating to generate trial structures
<b>Template/ Original Structure</b>	The previously defined crystal structure into which the new molecules are placed
<b>Original Landscape</b>	The CSP landscape from which templates are extracted
<b>Starting Set</b>	The set of extracted templates to be used in templating CSP
<b>Analogous Molecules</b>	Pairs (or families) of molecules that are meaningfully similar to one another. Pairs of analogous molecules comprise the ‘before’ and ‘after’ molecules used in templating CSP
<b>Analogue</b>	The crystal structure formed by placing new molecules into a template structure. This new structure is an analogue of the original structure
<b>Traditional CSP</b>	CSP conducted by conventional/non-templating methods. In this work traditional CSP is quasi-random CSP
<b>Target</b>	The system for which prediction is being attempted via templating CSP - The new molecule that is inserted into templates
<b>Target Landscape/ Traditional Landscape</b>	The CSP landscape of the target that would have been/has been obtained via traditional CSP methods

Table 8.1: Definitions of terms that will be used to describe work on templating CSP

## 8.3 Choosing Appropriate Systems

### 8.3.1 Overview

The first step in attempting templating CSP is to choose appropriate systems - i.e to pick pairs of analogous molecules - such that one landscape can be used to find template structures for prediction of the other. The proof-of-concept work began by testing templating CSP on sets of molecules that are clearly analogous- which, intuitively, should give templating the best chance of success.

Work here considered two categories of analogous molecules:

1. Substituted molecules - where interconversion between the molecules would be purely a case of chemical substitution
2. Differently-sized analogues - where one molecule could be considered a 'larger version' of the other

Cases were chosen to test the performance of templating CSP using each category of analogous molecules. In-order to provide varied testing , sets of analogous molecules were chosen to cover a range of molecular shapes and functional groups. In this work, all template crystal structures- and therefore all formed analogues - were of  $Z'=1$ .

### 8.3.2 Substituted Molecules

Possible sets of structures for templating investigations were identified systematically from an existing database of over 1000 small rigid molecule CSP sets [54]. This involved a search of the set of SMILES strings of the underlying molecules to identify all pairs of structures that could be inter-converted by three or fewer direct substitutions - in which individual non-H atoms were replaced by individual atoms of another element. The addition of hydrogens attached to the new atom was also possible. A non-exhaustive algorithmic approach based upon the reverse Cuthill-Mckee algorithm implemented via *reverse\_cuthill\_mckee* functionality in Scipy [107] was then used on the pairwise data to identify possible families in which all members could be inter-converted by three or fewer direct substitutions.

This process resulted in identification of 95 potential families. This set was then manually searched to find a subset of families that cover a range of chemical changes, molecular shapes, and chemistry of the unchanged substructure. Two families were selected for exploration in this thesis. For simplicity, the targets in these families will be referred to by the CSD refcodes of their respective experimental crystal structures. The two families investigated were:

1. BZDIOX, CONYAH, MEMTED, and WARPOW
2. VENYUI and VENZAP

The structures of these molecules can be seen in Figure 8.2.

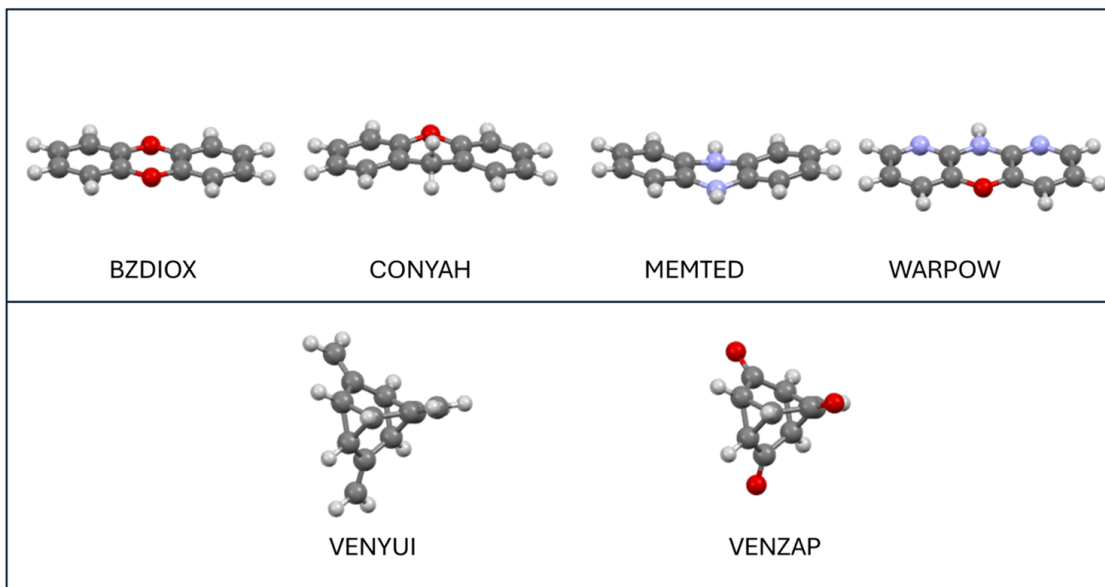


Figure 8.2: The structures of two families of chemically-substituted similar molecules used in investigation of templating CSP

### 8.3.3 Differently-Sized Analogues

Early work in this thesis performed CSP on a set of four semiconductors (NTCDA,PTCDA, MeNTCDI, MePTCDI). This set presents a useful case for exploration in templating. The set contains two pairs of differently-sized analogous molecules:

1. NTCDA and PTCDA
2. MeNTCDI and MePTCDI

Work in this thesis has explored the family of NTCDA and PTCDA (Figure 8.3) in addition to the chemically-substituted families discussed above. Exploration of the case of MeNTCDI and MePTCDI could extend and corroborate findings on templating with differently-sized analogous molecules and may be of interest for future work. The set also presents ‘cross pairs’ e.g NTCDA and MePTCDI that differ both by size and chemical substituents and could act as cases to test the limits of templating, again being of interest for additional investigation.

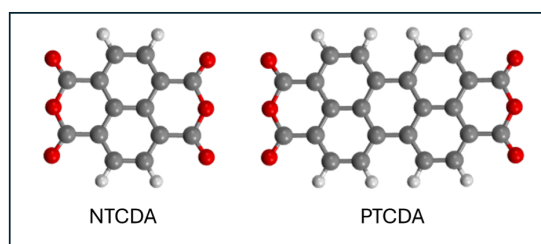


Figure 8.3: Molecular Structures of NTCDA and PTCDA - a pair of molecules investigated for templating CSP

## 8.4 Generating Analogous Crystal Structures

### 8.4.1 Approach

The following section outlines the workflow developed to construct structural analogues. As work progressed, limitations with regard to the generalisability of the method were uncovered. These are discussed in detail in Section 8.7, but not elaborated upon here.

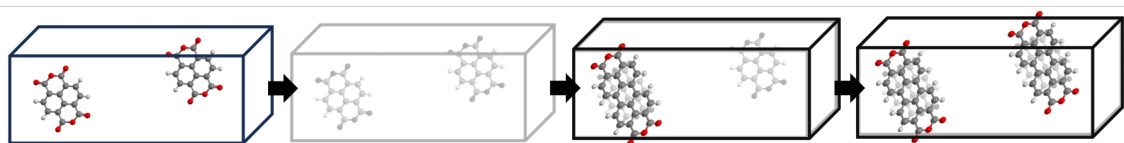


Figure 8.4: Conceptual process for generating analogous structures

To generate analogous crystal structures, the molecule(s) of the template asymmetric unit are replaced by the new molecule by sensibly overlaying the old and new molecules. The full analogue crystal is then generated by applying the unit cell parameters and spacegroup operations of the template (Figure 8.4).

‘Sensibly overlaying’ the molecules is a two step-process in which the new molecules must be correctly oriented and then correctly positioned. In the proof of concept work so far, only  $Z'=1$  systems have been investigated. So, in all cases discussed henceforth, there is a single asymmetric unit molecule.

### Orienting Molecules

Molecules are suitably oriented by use of substructure overlay. The maximum common substructure of the old and new molecules is identified, and attempts are made to overlay the pair so as to best overlay that shared substructure. The orientation of the new molecule corresponding to the lowest RMSD of the substructure overlay is taken to be the required molecular orientation.

The maximum common substructure between each pair of molecules is identified using the *MaximumCommonSubstructure* feature in the CSD API. The search is performed with restrictions such that the identified substructure must be connected - as it may not be possible to overlay two or more disconnected parts of the shared substructure of the molecules simultaneously. It is further ensured that the search is blind to bond-order. This is necessary due to the narrow margins for identifying bond-order - i.e if the search were not blind to bond order, a pair of molecules could share substructure that is not correctly identified, due to a given bond being inconsistently defined,



for example, as double in one case and single in the other.

The maximum common substructure having been defined, all instances of that substructure in each molecule are then identified. This is performed using the *SubstructureSearch* feature in the CSD API. The workflow establishes molecule objects for each molecule to be searched, alongside a molecule object containing the identified common substructure - and edits these objects to set all bond-types to 'unknown'. This allows the substructure search functionality to also behave as if blind to bond order. This search returns 'hits' of the substructure in each molecule from which the atom objects defining the substructure instance can be returned. The workflow then identifies the index from the original molecule file that corresponds to each of these atoms - obtaining ordered lists of indices to attempt to overlay between molecules. Figure 8.5 shows an example of identified maximum common substructure in pairs of molecules investigated.

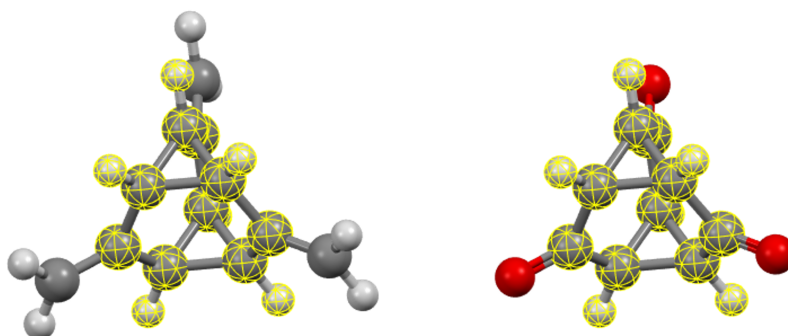


Figure 8.5: An example of a template-target pair investigated, and the maximum common substructure (highlighted in yellow) identified between them.

It is necessary for the workflow to identify **all** instances of the substructure in each molecule, and to attempt to overlay all combinations of substructure instances between old and new molecules, in order to capture all valid analogues. This is because overlaying different instances of the substructure may result in different analogue crystals - and the prediction should not be arbitrarily restricted in this way (See Figure 8.6).

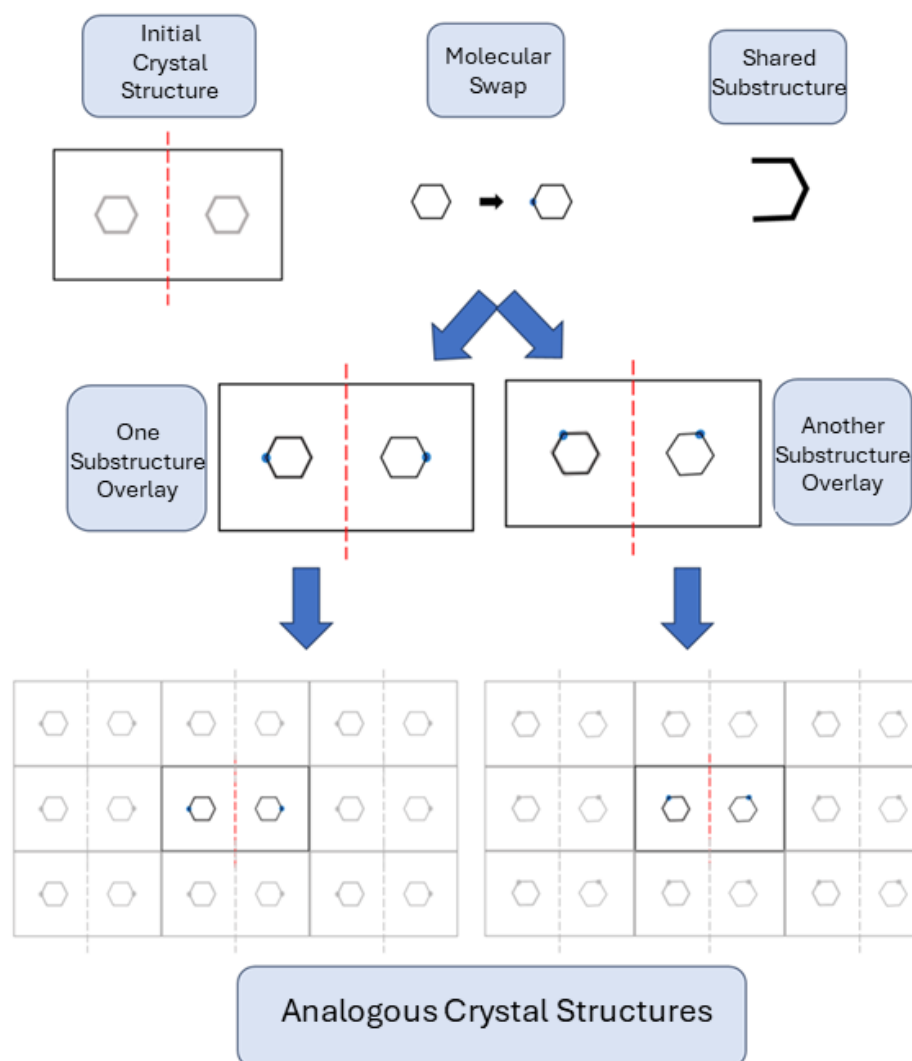


Figure 8.6: A conceptual example demonstrating the need to consider overlay of all pairs of instances of the shared substructure in order to capture all valid analogous crystals. Each example overlay shows the overlaying of the substructure in the new molecule with different instances of the substructure within the old molecule - leading to different crystal structures.

However, whilst this is necessary, it is not always sufficient. In some cases, there can also be multiple valid approaches to overlaying a pair of substructure instances - i.e which atom of the substructure instance in the new molecule should be overlaid with which atom of the substructure instance in the old molecule. This is a necessary concern in cases where the substructure has symmetry. This phenomenon is demonstrated most clearly by imagining a hyperbolic case (See Figure 8.7 ). Importantly for this work, and its likely future applications, most cases will have few instances of the maximum common substructure, and low symmetry of the substructure itself

- meaning that the problem of forming valid analogues usually remains feasible and affordable, despite the theoretical exponential growth in the number of analogues with these variables.

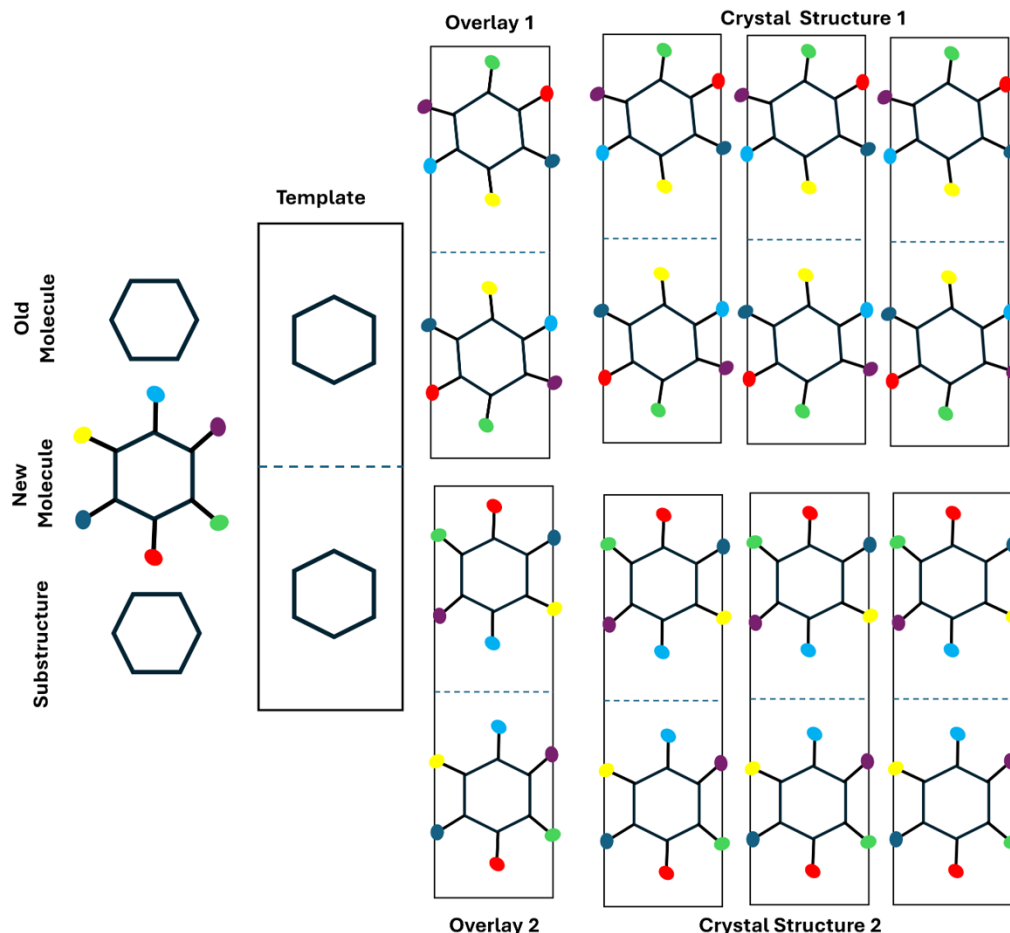


Figure 8.7: A conceptual example demonstrating the need to consider all isomorphic overlays of a single pair of substructure instances in order to capture all valid analogous crystals. Each example overlay shows the overlaying of the single present substructure instance in each molecule, albeit overlaying different pairs of atoms- each according to isomorphic mappings of the substructure instances - leading to different crystal structures.

The valid approaches to overlaying a given pair of substructure instances are given by isomorphic mappings of the corresponding molecular graphs. All such possible mappings are identified using the external package *networkx*[123]. Truly though, it is only rotational symmetry of the substructure that should be considered - the issue is discussed in Section 8.7.

The result of the substructure identification stage of the workflow is a series of pairs of ordered list of atom indices - one series per molecule. In each pair of lists, one list gives indices of some of the atoms in the old molecule, and the other gives indices of some of the atoms in the new

molecule. These are the sets of atoms to overlay according to their order - i.e the first atom of the list for the old molecule is overlaid with the first atom of the list for the new molecule. This series of list pairs should cover all valid overlays of the maximum common substructure between the molecules. This requirement to include all possible valid overlays served as one barrier to applying off-the-shelf alignment methods that focus on individually ranked optimal overlays, such as PubChem alignment methods [180].

An analogue must then be created for each valid case of the substructure overlay. An in-house code attempts to replace the molecule of the asymmetric unit of the template with the new molecule by overlay of the given sets of substructure atom indices. This step identifies the best alignment of the two molecules, such as to minimise the RMSD of the substructure overlay. As discussed, this decides the orientation of the new molecule in each case.

Additionally, this alignment partially accounts for position. If the desired positioning of the new molecules were also solely determined by that that facilitates best overlaying of the substructure, then the orientation and positioning of the new molecules into the template unit cell could be handled together by functionalities in CSPy. However, the true process is more complex.

### Positioning Molecules and Creating Crystal Structures

Simple positioning of the new molecules into the template unit cell by overlaying them with the old molecules would result in positioning that optimises substructure overlay. However, this is not what was desired for an analogue in this work. For this work, it was assumed that the fairest definition of an analogue would be that in which the centroid positions of the molecules in the analogue match the centroid positions of the template. This may not arise from substructure overlay (See Figure 8.8)

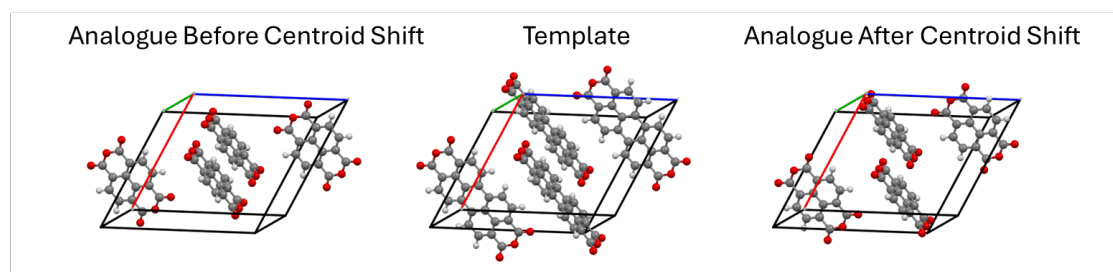


Figure 8.8: An example demonstrating the impact of shifting the centroid positions of molecules after forming an analogue via substructure overlay - altering the final analogue formed

The process therefore orients the new asymmetric unit molecule by substructure overlay - but then

translates the molecule so as to corrects the defined positions such that the geometric centroid position matches the geometric centroid position of the old molecule. Then, a full analogue is constructed by forming a crystal object using the newly defined asymmetric unit, along with the unit cell and space group of the template.

Generating analogues in this way can lead to clashes between molecules in the unit cell - particularly where the new molecule is larger along any given dimension than the original molecule in the template crystal structure. In the instance of molecular clashes, atoms of different molecules come close to one another such that many programs will predict formation of a covalent bond between them, thereby forming a different and often non-physical molecular unit within the crystal.

This issue can be relieved after the analogue is formed by increasing the size of the unit cell and shifting the molecules such that their centroids lie at the equivalent fractional positions along the lattice vectors of the larger unit cell. This largely maintains the crystal structure - albeit increasing the spacing between different molecules to avoid clashes. This is a process similar to that used to avoid clashes in the initial quasi-random generation of crystal structures in CSPy.

To remove clashes, an iterative process was used whereby:

1. The analogue was checked for clashes based upon whether the molecular graphs of the detected unit cell molecules matched the expected molecular unit
2. If a clash was detected, the length of the shortest lattice parameter of the original crystal was increased by 1 Å
3. All molecular positions were adjusted such that molecule centroid positions maintained the same fractional co-ordinates along each unit cell axis

For the purposes of developing a proof-of concept, attempts were not made to restrict this process based on reasonable cell volume, or other physical constraints. Instead, the process was simply halted if the number of size increases exceeded 60 - and the corresponding analogue structures were rejected.

Lastly, any duplicate analogues are removed by pXRD clustering, such that work progresses with a complete set of unique analogues.

### **Full Analogue Generation**

A flowchart of the full analogue generation process is shown in Figure 8.9.

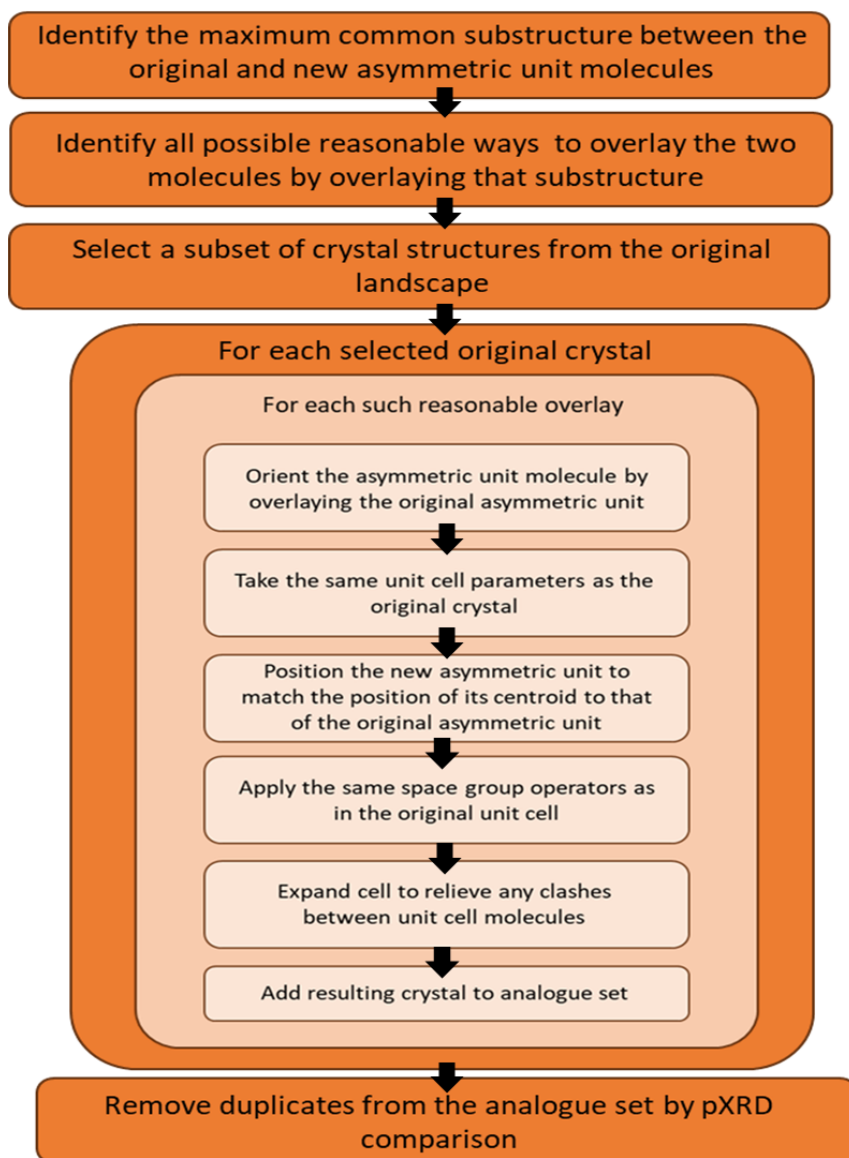


Figure 8.9: Flowchart of full process for generating analogues

Figure 8.10 shows examples of template structures, alongside some of the analogues constructed from them - demonstrating that the derived structures can indeed be considered analogous.

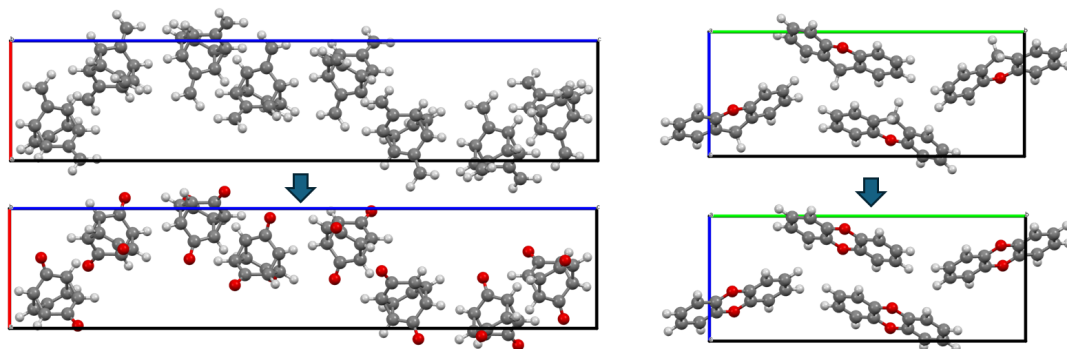


Figure 8.10: Examples of template structures, and one of the analogous crystal structures that was formed from each of them

#### 8.4.2 Optimisation

To obtain the final templating CSP landscape, the unique analogues are then lattice-energy minimised. In this way, templating replaces the sampling step of a CSP workflow, with the remaining workflow progressing as normal. If templating CSP were to be implemented in real applications, various optimisation approaches could be applied. However, for testing purposes, the optimisation workflow applied in each case was selected to match that used to derive the traditionally generated target landscape. This ensured that the success or failure in recovering structures can be attributed solely to the sampling approach - and not to any differences in the energetic method used in optimisation.

In the examples in this thesis, the optimisation therefore relies upon pairwise interatomic force-fields coupled with distributed atomic multipoles. The approach treated molecules as rigid - that is that during optimisation the unit cell parameters and asymmetric unit positions could change - but the in-crystal conformations remained unaltered. The extent of structural change under optimisation was not investigated in this work, though it could be seen that this varied between constructed analogues. An example of an analogue structure before and after lattice energy minimisation can be seen in Figure 8.11.

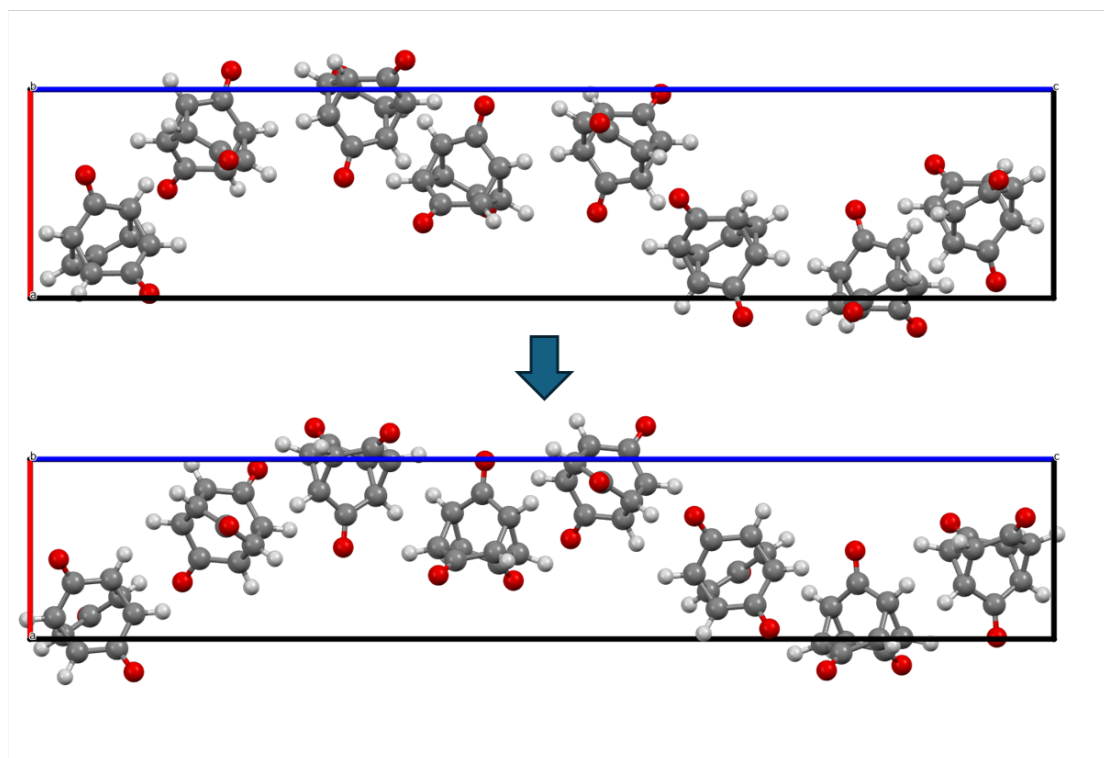


Figure 8.11: An example of an initial structural analogue formed in templating CSP, and the corresponding final structure after lattice energy minimisation.



## 8.5 Evaluation of Templating CSP

### 8.5.1 Tested Approaches

#### Overview

Work aimed to investigate the performance of templating CSP methods and the starting sets required for success, including exploration of the potential of the GCH in identifying structures for analogue formation. Therefore, work measured the performance of templating CSP relative to traditional quasi-random CSP, for several cases - with starting sets for analogue formation selected based upon energy rankings and dressed energy rankings within the original landscape.

#### Traditional Landscapes

For testing purposes, traditionally-generated CSP landscapes of all considered molecules were required. This was in order both to form analogues - i.e to have structures to use as templates - and to evaluate the approaches - i.e to see what structures would have been predicted via traditional methods, for comparison to the predictions made by templating CSP.

The traditional landscapes of NTCDA and PTCDA were generated during earlier work in this thesis, and the remaining traditional landscapes were obtained from the literature [54]. In all cases, the landscapes were obtained via quasi-random generation of trial structures, followed by optimisation with FIT + DMA. Sampling in the case of NTCDA and PTCDA generated 10 000 structures in each of the 10 space groups most common to crystals of small organic molecules [124] ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ). The remaining cases more extensively sampled 10 000 structures in each of the 26 most common spacegroups ( $P12_1/c1$ ,  $P2_12_12_1$ ,  $P\bar{1}$ ,  $P12_11$ ,  $Pbca$ ,  $C12/c1$ ,  $Pna2_1$ ,  $C121$ ,  $P1$ ,  $Pbcn$ ,  $P1c1$ ,  $P2_12_12$ ,  $Fdd2$ ,  $Pccn$ ,  $P12/c1$ ,  $I4_1/a$ ,  $R\bar{3}$ ,  $P4_1$ ,  $P4_32_12$ ,  $P4_12_12$ ,  $P4_3$ ,  $P3_2$ ,  $P3_1$ ,  $P6_1$ ). All searches were restricted to generation of  $Z'=1$  crystals.

The traditional landscapes were then clustered using both pXRD clustering and COMPACK clustering. However, due to the aforementioned potential for COMPACK clustering via the CSD API to miss some duplicates [17, 18], it is possible that a few duplicates may remain in the traditional landscapes. This potential under-clustering is a point to bear in mind during interpretation of results.

### Energy Windows

To investigate the impact of the selection of structures from which analogues are formed and to gauge the minimal starting sets needed for successful CSP, multiple runs of templating were performed in which the set of initial structures was varied. Starting sets used for each system were based upon energy windows on the initial landscape

1. The entire available landscape (set 1 : 'Full Landscape')
2. The global minimum structure on the landscape (set 2: 'Global Minimum')
3. The lowest 15 kJ/mol on the landscape (set 3: '15 kJ/mol')
4. The lowest 25 kJ/mol on the landscape (set 4: '25 kJ/mol')

Further, to extend this exploration and to evaluate the potential of the Generalised Convex Hull in aiding templating, the equivalent starting sets were extracted via the GCH construction - that is based upon dressed energy rather than lattice energy:

1. The structures corresponding to hull vertices (set 5: 'Vertices')
2. The lowest  $x$  structures ranked by dressed energy where  $x = |\text{set 3}|$  (set 6: 'Count from 15 kJ/mol')
3. The lowest  $x$  structures ranked by dressed energy where  $x = |\text{set 4}|$  (set 7: 'Count from 25 kJ/mol')

In principle, to perform templating using structures extracted via the Generalised Convex Hull, there are two key parameters that must be defined:

1. The underlying kernel construction
2. The dimensionality of the constructed hull

These parameters may be tested in future work. However, for the purposes of developing the proof of concept, investigations only explored extraction of starting structures from 1D hulls derived using the adapted kernel with a SOAP cut-off radius of 4 Å.

### 8.5.2 Metrics

The performance of templating CSP was evaluated according to three main metrics:

1. Known Crystal Structure Recovery: Whether or not matches to the known crystal structure(s) were found using the templating CSP

2. Recovery Percentage: The percentage of low-energy structures that would be found using traditional CSP that were also found using templating CSP

$$\frac{\text{No. unique low energy traditional minima recovered}}{\text{No. unique low energy minima in traditional CSP}} \times 100$$

3. Efficiency Ratio: The ratio between the number of **unique** recovered low-energy structures and the number of unique starting analogues minimised in templating CSP to retrieve them

$$\frac{\text{No. unique low energy traditional minima recovered}}{\text{No. unique starting analogues}}$$

These metrics are designed to assess the important characteristics of any successful CSP methods - namely being able to predict experimental forms, finding the wider landscape of local minima structures to assess risk of additional structures, and to work with as low a cost as feasible.

### 8.5.3 Recovery of Known Crystal Structures

The geometry-optimised structures of the analogues formed from each original landscape starting set were analysed to identify potential matches to the known crystal structure(s) of the target. Matches were identified as in the usual testing of molecular CSP workflows, using the *CrystalPackingSimilarity* feature in the CSD API. A match to the experimental structure was recorded if a 30/30 COMPACT match of predicted and experimental structures could be found within tolerances of 0.2 Å and 20 °.

Tables 8.2, 8.3 and 8.4 show the number of the known crystal structures successfully predicted using templating CSP for each target, using each starting set to form analogues.

System	Starting Set						
	Full Landscape	25 kJ/mol	Count from 25 kJ/mol	15 kJ/mol	Count from 15 kJ/mol	Global Minimum	Vertices
NTCDA in PTCDA	1/1	1/1	1/1	1/1	1/1	0/1	0/1
PTCDA in NTCDA	2/2	1/2	1/2	0/2	0/2	0/2	0/2

Table 8.2: Prediction via templating CSP of structures matching to known experimental polymorphs for the NTCDA and PTCDA template-target pairs, for each trialled starting set

System	Starting Set						
	Full Landscape	25 kJ/mol	Count from 25 kJ/mol	15 kJ/mol	Count from 15 kJ/mol	Global Minimum	Vertices
BZDIOX in CONYAH	1/1	1/1	1/1	1/1	1/1	0/1	0/1
CONYAH in BZDIOX	1/1	1/1	1/1	1/1	1/1	0/1	0/1
WARPOW in CONYAH	1/1	1/1	1/1	1/1	1/1	0/1	0/1
CONYAH in WARPOW	1/1	1/1	1/1	0/1	1/1	0/1	0/1
MEMTED in CONYAH	1/1	1/1	1/1	1/1	1/1	0/1	0/1
CONYAH in MEMTED	1/1	1/1	1/1	1/1	1/1	0/1	1/1
BZDIOX in WARPOW	1/1	1/1	1/1	1/1	1/1	0/1	0/1
WARPOW in BZDIOX	0/1	0/1	0/1	0/1	0/1	0/1	0/1
MEMTED in WARPOW	1/1	1/1	1/1	0/1	1/1	0/1	0/1
WARPOW in MEMTED	1/1	1/1	1/1	0/1	0/1	0/1	0/1
MEMTED in BZDIOX	1/1	1/1	1/1	1/1	1/1	0/1	0/1
BZDIOX in MEMTED	1/1	1/1	1/1	0/1	0/1	0/1	0/1

Table 8.3: Prediction via templating CSP of structures matching to known experimental polymorphs for the first family of chemically-substituted template-target pairs, for each trialled starting set

System	Starting Window						Vertices
	Full Landscape	25 kJ/mol	Count from 25 kJ/mol	15 kJ/mol	Count from 15 kJ/mol	Global Minimum	
VENYUI in VENZAP	1/1	1/1	1/1	1/1	1/1	1/1	1/1
VENZAP in VENYUI	1/1	1/1	1/1	1/1	1/1	1/1	1/1

Table 8.4: Prediction via templating CSP of structures matching to known experimental polymorphs for the second family of chemically-substituted template-target pairs, for each trialled starting set

The tables are colour-coded to represent success/failure - with green cells representing cases where all known crystal structures were recovered, orange cells representing cases where some of the known crystal structures were recovered, and red cells representing cases where known crystal structures were not found. It should be noted that PTCDA is the only polymorphic target, and so in all other cases success/failure is binary.

Figure 8.12 shows an example of structures predicted via templating CSP that match to the known experimental polymorphs ( $\alpha$  &  $\beta$ ) of PTCDA. The matches are considered to be of good-quality, with RMSD<sub>30</sub> values of 0.375 Å and 0.362 Å respectively.

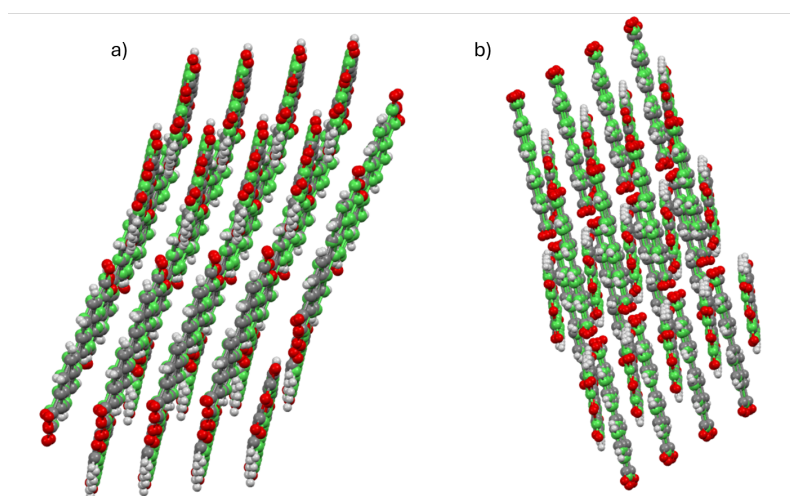


Figure 8.12: Overlays of known a)  $\alpha$  and b)  $\beta$  crystal structures of PTCDA (Element Colour) and crystal structures predicted via templating CSP (green).

Known crystal structure recovery was generally successful when forming analogues from the full original landscape, recovering all known structures except in the case of predicting WARPOW structures from BZDIOX analogues. It is unclear why this case presented a particular challenge. This success is promising in some cases - placing PTCDA into the templates from the full NTCDA landscape resulted in just 664 unique analogue structures, and still proved sufficient for the prediction task. However, other cases involve larger starting sets. For example, forming MEMTED analogues of the full CONYAH landscape formed > 24000 unique analogues prior to optimisation, making successful prediction far more likely - and the promise of templating as a fast CSP approach less clear.

Prediction remained successful in most cases when using templates of just the 25 kJ/mol window of the original landscape, or equivalent number of original templates selected based upon their proximity to the hull. At this level, the known crystal structure recovery metric cannot distinguish between extracting starting structures based upon relative energy or dressed energy - with the corresponding results being identical. When templating into the 15 kJ/mol and count from 15 kJ/mol starting sets, results were more variable, and the first distinction between selecting starting structures based on relative energy or dressed energy can be seen. However, this does not lead to any clear conclusion as to the ‘stronger’ approach, with successes and failures in both methods. When using templates of merely the global minimum or the vertex structures - success was expectedly rare. Success in such cases would require recovering the known crystal structure from starting sets of just < 20 structures - a difficult task.

One noteworthy result shown is that when replacing VENYUI into VENZAP templates and vice versa there was success in recovering the known crystal structures - even when forming analogues of just the global minimum structure. This arises because in both original landscapes, the predicted global minima structures match to the corresponding known experimental structure [177] and the two experimental structures are isostructural - with representative clusters being overlaid with  $\text{RMSD}_{30}=0.835 \text{ \AA}$  and  $\text{RMSD}_{15}=0.685 \text{ \AA}$  (Figure 8.13).

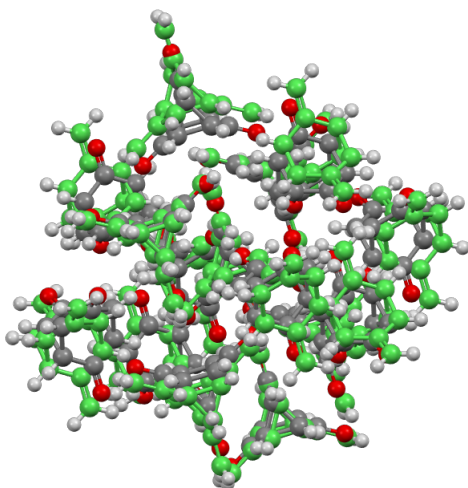


Figure 8.13: Overlay of the known experimental crystal structures of VENYUI (green) and VEN-ZAP (Element-Colour), demonstrating their isostructural nature.

This case, however, also presented an example of the need for caution when exploring results based upon *CrystalPackingSimilarity* searches. Whilst investigations showed that a constructed analogue matched to the experimental structure of VENYUI, it was also found that the constructed analogue did **not** match to the predicted global minimum of the original VENYUI landscape - despite that predicted minimum itself having been a match to the experimental structure. Whilst this ‘breaking the cycle of matches’ is not strictly impossible, given the quality of the measured matches in each case it is unlikely - and so presented a cause for concern. Attempting search for an overlay between the promising constructed analogue and the traditionally predicted global minimum again, albeit via the *Mercury GUI*, recovered a match ( $\text{RMSD}_{30} = 0.001 \text{ \AA}$ ,  $\text{RMSD}_{15} = 0.001 \text{ \AA}$ ) (Figure 8.14). This inconsistency has been seen previously, and may be linked to issues with *CrystalPackingSimilarity* noted in the 7th CSP Blind Test [17, 18]. It should be noted that the potential of missed matches may have impacted the results presented here, and so they should be read with that caution in mind. However, the concern is not unique to this thesis, and given the prior ubiquity of *CrystalPackingSimilarity* methods in assessing molecular CSP, is a commonplace concern for much past work in the area. The results shown later in this thesis were corrected to account for the now proven match.

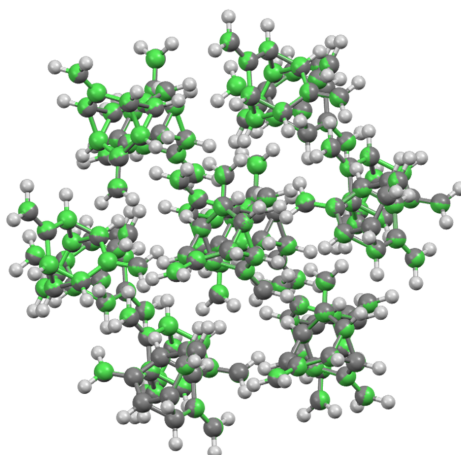


Figure 8.14: Overlay of the traditionally-predicted global minimum crystal structure of VENYUI (green) and a structure generated via templating CSP (Element-Colour), demonstrating a close match between the structures.

#### 8.5.4 Recovery Percentages

In showing a proof of concept for a fast CSP method, it is necessary not only to predict the already known crystal structures, but to predict the important structures that would have been predicted via traditional methods - to replicate the traditionally generated low-energy landscape. After all, it is this ability to predict beyond what is already known - or to propose possible structures beyond a single global minimum prediction - that is a major selling point of CSP methods. This motivated testing of the recovery percentages metric.

The recovery percentages were tested in two respects:

1. The percentage of the lowest 7.5 kJ/mol of the traditional CSP landscape recovered
2. The percentage of the lowest 25 kJ/mol of the traditional CSP landscape recovered

The most crucial of these is the recovery of the lowest 7.5 kJ/mol - due to this being the region in which experimental forms are likely to lie [37]. The recovery of the lowest 25 kJ/mol remains desirable however - especially given the previously discussed possibility of higher energy polymorphs - and serves as a greater challenge to assess templating CSP.

In both cases, ‘recovered’ structures are those structures from the relevant region of the target landscape for which a 30/30 COMPACT match could be found among the structures predicted by templating CSP. Each traditional CSP structure could only contribute to the count of recovered structures once, i.e - a traditional landscape minima may be found several times by templating CSP,



but only one ‘recovered structure’ of these instances was counted. It could additionally be possible that a given templating CSP minimum could ‘find’ more than one traditional landscape minimum. However, this should not be considered as true recovery. To demonstrate via an extreme example, a landscape containing a single structure could not be said to have replicated another low energy landscape that contains two structures - even if the single structure matched to both. In an attempt to simultaneously prevent such cases and ensure that recovered structures are defined conservatively and with confidence, a strict RMSD criterion was applied. Only matches with  $\text{RMSD}_{30} \leq 0.05 \text{ \AA}$  were able to contribute to the recovered structure count.

Whilst the relationship between measured RMSD and match ‘quality’ is not always a simple one, such a strict criterion should ensure that all claimed recovered structures match closely with a structure predicted via templating CSP. Most successful overlays between traditional and templating CSP structures had  $\text{RMSD}_{30} \ll 0.05 \text{ \AA}$  - testament to having truly reached the same minima on the energy landscape despite different approaches to the sampling. Figure 8.15 shows an example of such an overlay, with an  $\text{RMSD}_{30}$  of  $0.0006 \text{ \AA}$ .

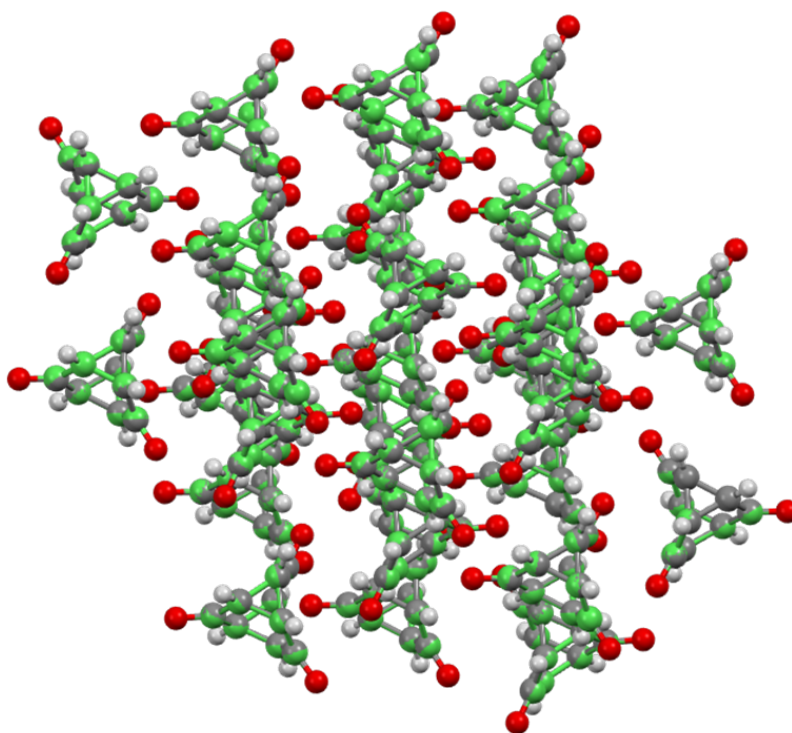


Figure 8.15: Overlay between a structure predicted by quasi-random CSP (element colour) and a structure predicted by templating CSP (green)- showcasing an almost perfect overlay

**Data Overview**

Table 8.5 shows the number of unique structures in the 25 kJ/mol and 7.5 kJ/mol windows on each traditionally-generated CSP landscape. Table 8.6 shows the number of unique starting analogues formed (prior to minimisation), and the number of traditionally generated structures recovered, in each attempted templating case. These numbers provide an overview of the task at hand for templating CSP - as well as insight into its performance. Notably, the population density of traditional CSP landscapes varies significantly, leading to large ranges in both the number of starting analogues formed and the number of crystal structures that templating CSP must attempt to predict.

One conclusion that is clear from this variation of the landscape densities seen in Table 8.5 is that the success of templating CSP can never be entirely successful in all cases, and that success is likely to be ‘asymmetrical’. That is, that while it is theoretically possible to have 100% success in predicting a one landscape using templates from another - this is dependent upon the number of analogues formed per template. For instance, when predicting a more densely populated target landscape by using templates from a less densely populated landscape, the approach may struggle - especially if only a few analogues of each template are formed. One solution to this problem may be to combine templating approaches with basin-hopping CSP, using a small number of structural analogues to each seed multiple basin-hopping trajectories, thereby allowing discovery of more minima. This would potentially facilitate success of templating CSP from smaller initial starting sets. Such investigation could be of interest for future work.

Target System	No. Unique Structures in 7.5 kJ/mol Window of Traditional CSP	No. Unique Structures in 25 kJ/mol Window of Traditional CSP
NTCDA	2	126
PTCDA	10	131
BZDIOX	307	1947
CONYAH	153	7291
WARPOW	137	1465
MEMTED	40	917
VENYUI	16	642
VENZAP	27	405

Table 8.5: The number of structures in important regions of the traditional CSP landscape for each investigated system

System	Starting Set	No. Unique Analogues Formed	No. Structures in 7.5 kJ/mol Window Recovered	No. Structures in 25 kJ/mol Window Recovered
NTCDA in PTCDA	Full Landscape	1155	2	101
	25 kJ/mol	126	2	51
	Count from 25 kJ/mol	128	2	42
	15 kJ/mol	55	2	25
	Count from 15 kJ/mol	57	2	26
	Global Minimum	1	0	1
	Vertices	11	0	8
PTCDA in NTCDA	Full Landscape	664	7	75
	25 kJ/mol	125	4	46
	Count from 25 kJ/mol	126	3	44
	15 kJ/mol	18	2	12
	Count from 15 kJ/mol	18	0	8
	Global Minimum	1	0	1
	Vertices	5	0	2
BZDIOX in CONYAH	Full Landscape	9529	276	1576
	25 kJ/mol	7215	273	1436
	Count from 25 kJ/mol	7209	273	1459
	15 kJ/mol	1873	218	639
	Count from 15 kJ/mol	1862	219	684
	Global Minimum	1	1	1
	Vertices	13	3	11
CONYAH in BZDIOX	Full Landscape	5515	104	2816
	25 kJ/mol	4571	103	2584
	Count from 25 kJ/mol	4576	103	2578
	15 kJ/mol	2284	93	1549
	Count from 15 kJ/mol	2291	95	1557
	Global Minimum	1	1	1
	Vertices	10	2	10
Continued on next page				

Table 8.6 – continued from previous page

WARPOW in CONYAH	Full Landscape	9452	120	1105
	25 kJ/mol	7215	119	733
	Count from 25 kJ/mol	7206	119	731
	15 kJ/mol	1871	98	464
	Count from 15 kJ/mol	1863	100	456
	Global Minimum	1	0	0
	Vertices	13	1	9
CONYAH in WARPOW	Full Landscape	2858	101	1674
	25 kJ/mol	2174	91	1302
	Count from 25 kJ/mol	2150	89	1288
	15 kJ/mol	897	31	588
	Count from 15 kJ/mol	850	48	582
	Global Minimum	2	0	0
	Vertices	15	2	9
MEMTED in CONYAH	Full Landscape	24996	39	724
	25 kJ/mol	18607	39	719
	Count from 25 kJ/mol	18585	39	719
	15 kJ/mol	4742	39	588
	Count from 15 kJ/mol	4620	38	557
	Global Minimum	3	0	2
	Vertices	35	1	14
CONYAH in MEMTED	Full Landscape	9116	139	2900
	25 kJ/mol	5000	132	1807
	Count from 25 kJ/mol	5018	137	1813
	15 kJ/mol	994	83	360
	Count from 15 kJ/mol	1027	87	450
	Global Minimum	6	2	3
	Vertices	33	8	23
Continued on next page				

Table 8.6 – continued from previous page				
BZDIOX in WARPOW	Full Landscape	1813	212	911
	25 kJ/mol	1391	191	751
	Count from 25 kJ/mol	1378	189	753
	15 kJ/mol	536	132	318
	Count from 15 kJ/mol	528	139	321
	Global Minimum	1	0	0
	Vertices	11	6	9
WARPOW in BZDIOX	Full Landscape	2584	81	775
	25 kJ/mol	2128	81	733
	Count from 25 kJ/mol	2127	81	731
	15 kJ/mol	1017	73	464
	Count from 15 kJ/mol	1014	71	356
	Global Minimum	1	0	0
	Vertices	9	0	4
MEMTED in WARPOW	Full Landscape	2901	35	545
	25 kJ/mol	2235	32	462
	Count from 25 kJ/mol	2198	31	469
	15 kJ/mol	923	14	219
	Count from 15 kJ/mol	875	21	246
	Global Minimum	2	0	0
	Vertices	17	0	8
WARPOW in MEMTED	Full Landscape	1801	83	525
	25 kJ/mol	981	60	336
	Count from 25 kJ/mol	992	70	335
	15 kJ/mol	201	5	74
	Count from 15 kJ/mol	214	19	95
	Global Minimum	1	0	1
	Vertices	7	0	4
Continued on next page				

Table 8.6 – continued from previous page				
MEMTED in BZDIOX	Full Landscape	7041	34	446
	25 kJ/mol	5897	34	438
	Count from 25 kJ/mol	5904	34	438
	15 kJ/mol	3015	32	351
	Count from 15 kJ/mol	3011	32	356
	Global Minimum	2	0	0
	Vertices	21	0	4
BZDIOX in MEMTED	Full Landscape	3202	146	468
	25 kJ/mol	1769	110	292
	Count from 25 kJ/mol	1783	112	276
	15 kJ/mol	378	19	66
	Count from 15 kJ/mol	368	40	87
	Global Minimum	2	0	1
	Vertices	12	2	6
VENYUI in VENZAP	Full Landscape	580	7	179
	25 kJ/mol	380	5	134
	Count from 25 kJ/mol	384	7	139
	15 kJ/mol	99	3	47
	Count from 15 kJ/mol	104	4	43
	Global Minimum	1	1	1
	Vertices	9	2	4
VENZAP in VENYUI	Full Landscape	1054	20	217
	25 kJ/mol	608	19	176
	Count from 25 kJ/mol	609	19	172
	15 kJ/mol	96	12	52
	Count from 15 kJ/mol	96	8	50
	Global Minimum	1	1	1
	Vertices	6	3	5

Table 8.6: Number of unique analogues formed and number of target structures recovered in each region for each template-target pair and starting set.

**Recovering the  $\leq 7.5$  kJ/mol region**

Figure 8.16 shows the percentage recovery of the lowest 7.5 kJ/mol of the target landscapes when using templating CSP to investigate the NTCDA/PTCDA family. Predicting crystal structures of NTCDA via analogues of PTCDA proved very successful, with recovery percentages of 100% arising from all substantial starting sets. However, no structures could be recovered via analogues of the PTCDA global minimum or 1D hull vertices. The reverse case had poorer performance, with even the 25 kJ/mol / count from 25 kJ/mol windows proving to be insufficient starting sets. One reason for this could be that in placing the larger PTCDA molecules into the NTCDA templates, many inter-molecular clashes are encountered, meaning that the approach must rely heavily on the sub-optimal approach to clash relief. This may lead to poor starting structures that differ greatly from the original template.

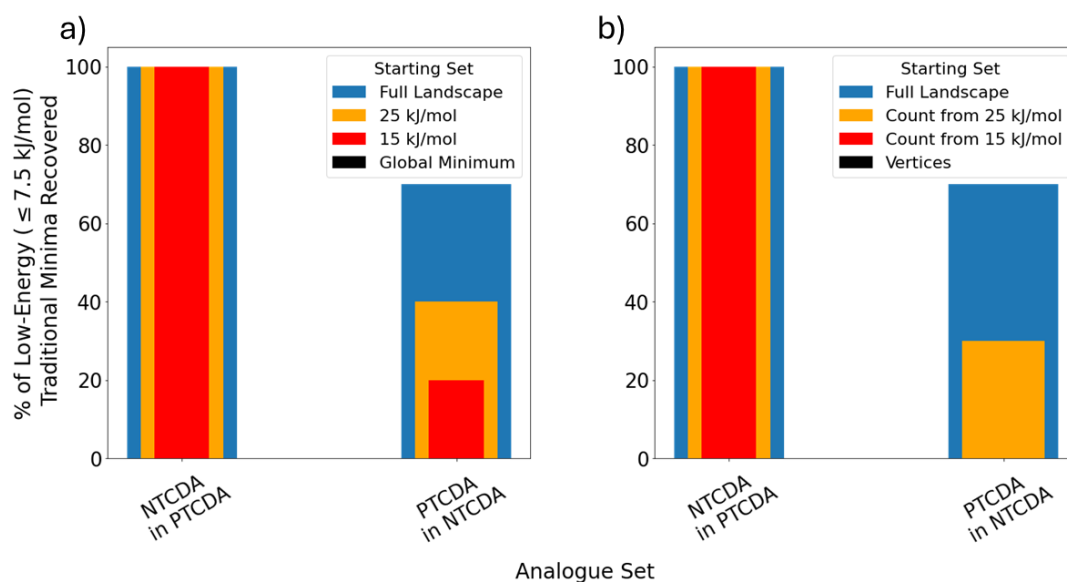


Figure 8.16: Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

Figure 8.17 and 8.18 show the same statistics for the chemically-substituted families.



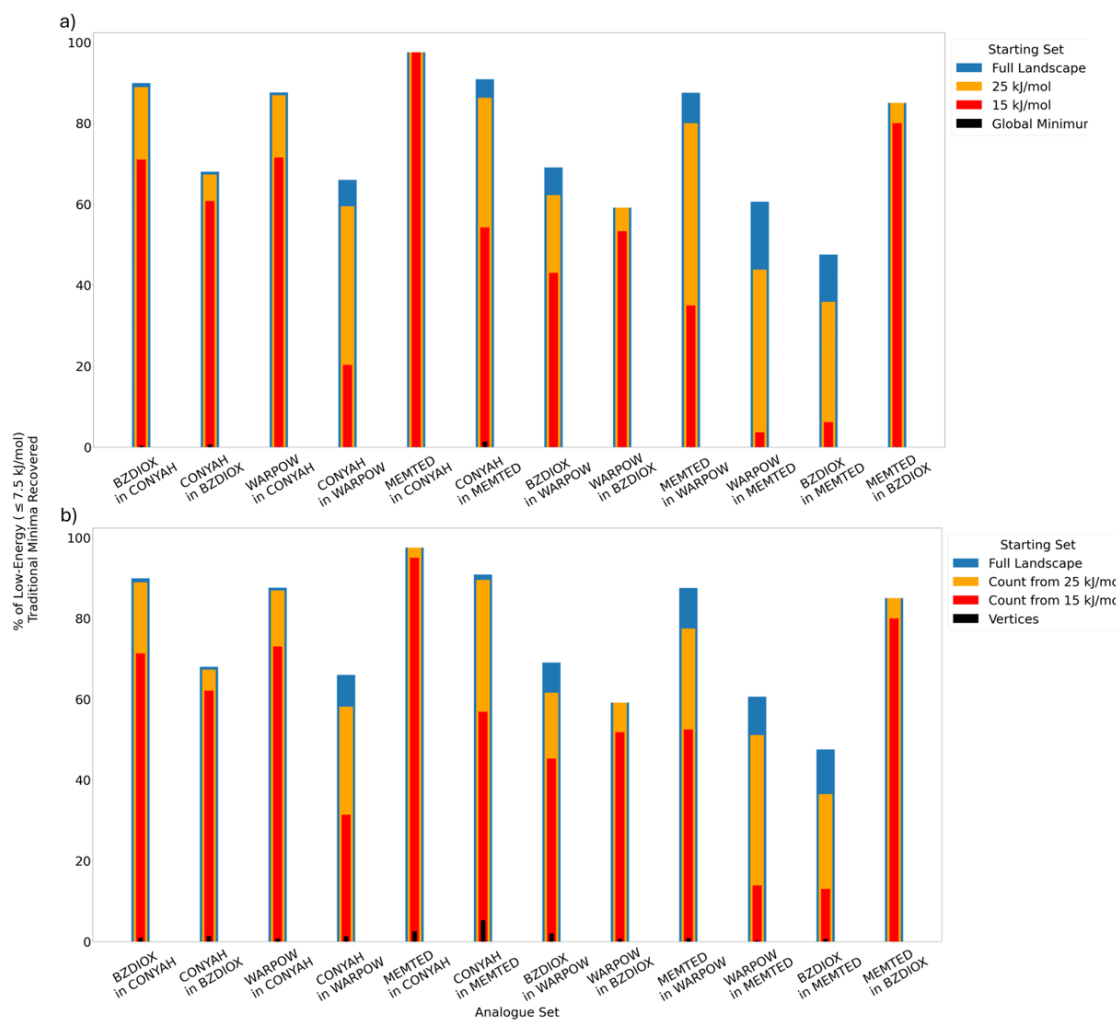


Figure 8.17: Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

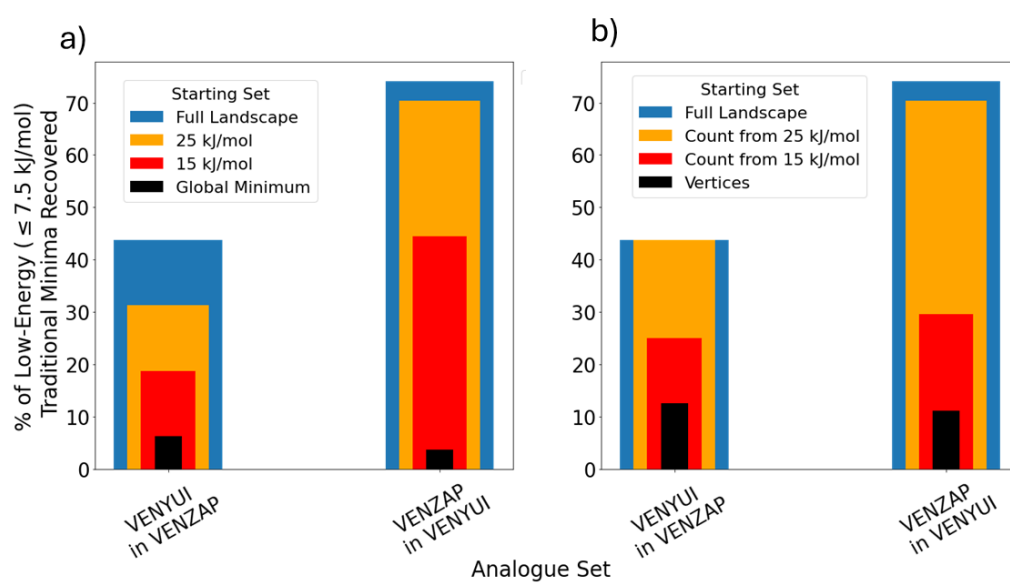


Figure 8.18: Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

This provides a greater set of examples, from which it can be seen that templating approaches are generally successful in reproducing the lowest 7.5 kJ/mol of a target landscape when beginning from a full landscape, although there are a few poorer cases. From the data, it would appear that in most instances forming analogues of structures from a 25 kJ/mol window/count from 25 kJ/mol window is almost as successful as when using a whole landscape of templates. However, in many cases the full landscape is not considerably larger than the 25 kJ/mol window - which may explain this finding. Performance when extracting structures based on a the tighter 15 kJ/mol window or associated count is more varied. It may be that pure templating approaches will not be able to replicate traditional CSP low-energy landscapes with such constrained starting sets. Again, little distinction can be drawn between the relative energy and dressed energy based approaches to starting structure selection. There are cases that perform better via each method.

One concerning result for the method is that the success of recovering a given target landscape appears to depend heavily upon the choice of template system. For example, the recovery of low-energy structures of the BZDIOX target landscape ranges from 47.6 to 89.9% depending on the template system. This could be due to limitations in the size of the respective starting sets, or due to varying strengths of relationship between the CSP landscapes of the template-target pairs. One adaptation that could help to address both issues would be the pooling of templates from multiple CSP landscapes.

### **Recovering the $\leq 25$ kJ/mol region**

In general, templating CSP recovered the most crucial 7.5 kJ/mol window of target landscapes with sufficient success as to offer promise for development of the proof of concept. A more challenging task, however, would be to reproduce the lowest 25 kJ/mol of a traditional CSP landscape. This is less vital, but if templating CSP were up to the mark, this would provide greater confidence in the methods and allow for the faster approach to CSP to be used in a greater range of scenarios, offering more promise for materials discovery. Figures 8.19, 8.20, and 8.21 show the recovery of the lowest 25 kJ/mol of the target landscapes.

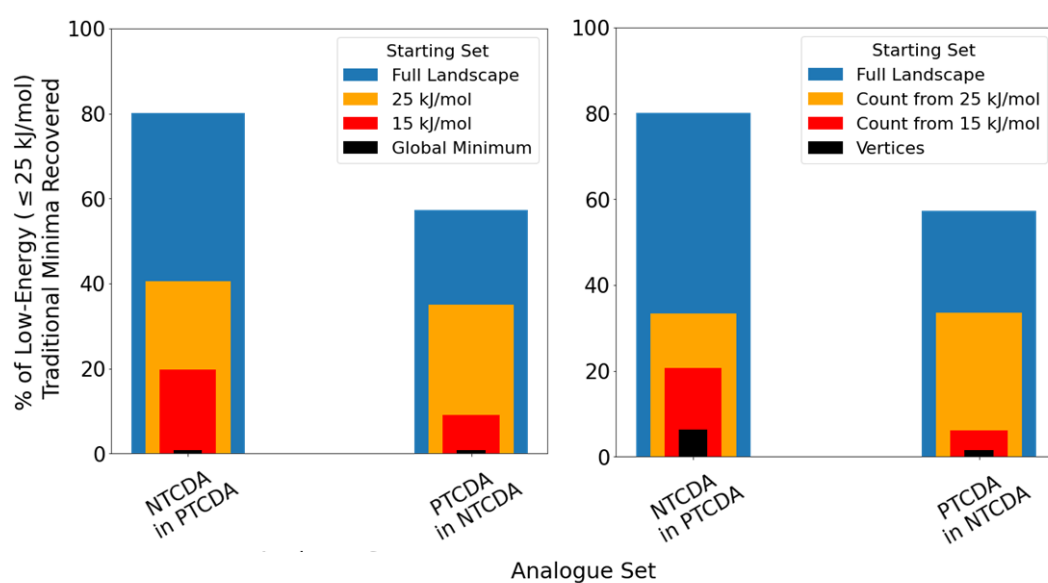


Figure 8.19: Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

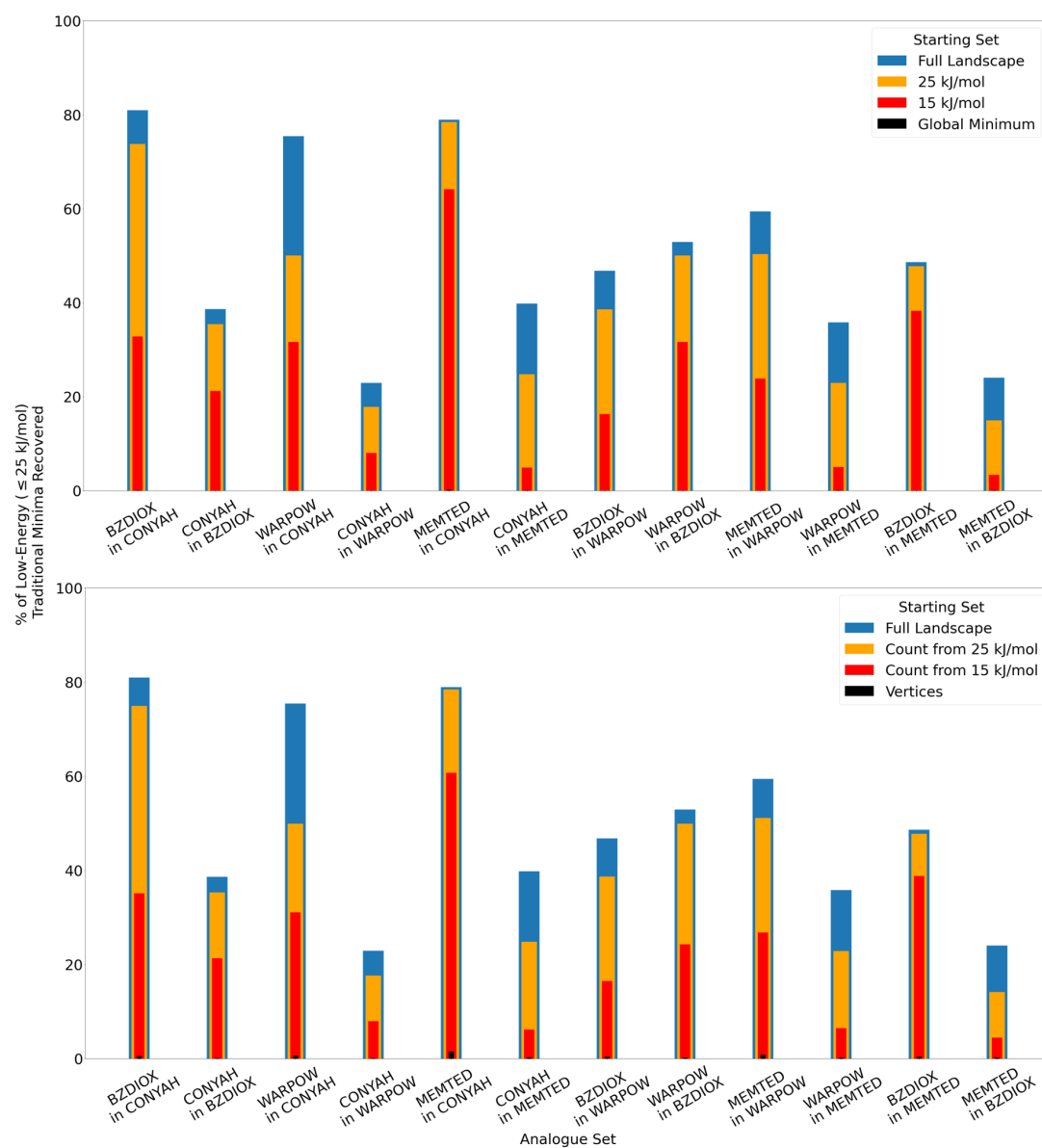


Figure 8.20: Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

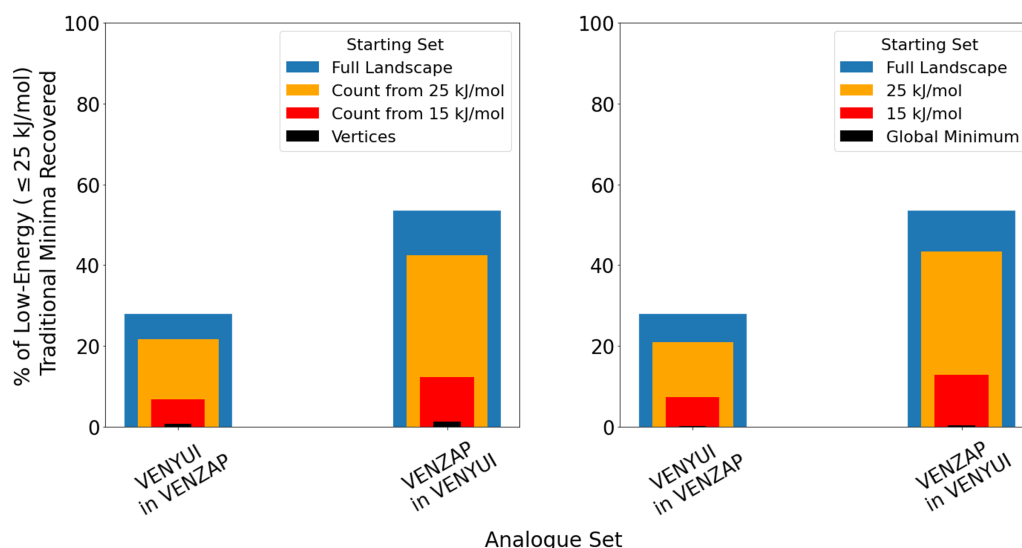


Figure 8.21: Percentage of the lowest 25 kJ/mol region of the target landscape recovered from templating CSP of the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

The pattern of results is similar to that of the 7.5 kJ/mol window recovery, and the results add little to the findings of the investigation in this respect. However, the findings do confirm that the recovery of this larger region of the target landscapes is poorer- particularly for the smaller starting sets.

One potential exploration not investigated in this thesis is to quantify the performance of structures in a higher-energy **band**, rather than exploring the nested 7.5 and 25 kJ/mol regions of the target landscapes. This may help to identify whether poorer percentage recovery seen here is simply due to the greater number of traditional CSP structures in the region, or due to formed analogues being more likely to minimise to the structures in the lowest energy region of the target landscape.

Potentially of interest here are cases in which analogues formed from templates of the original global minimum do not recover any structures in the 25 kJ/mol region of the target landscape. Whilst recovery could never be high given such small sets of analogues, these cases remain surprising. Such cases represent template-target pairs in which structures analogous to the global thermodynamic minimum on the original landscape cannot be found in even a large energy range on the target landscape - suggesting a disconnect between the most promising region of the landscapes. This emphasises the need to consider a range of template structures for analogue formation and may highlight risk in analogue-based approaches guided solely by known experimental struc-

tures.

It appears that, in its current state, templating CSP has potential in recovering the most important low-energy region of target landscapes, and so the proof-of-concept for a fast-CSP approach warrants further investigation. However, results are variable - especially given the variation in population density of energy landscapes. Additionally, recovery of the 25 kJ/mol region of target landscapes proves a tricky task. Therefore, templating CSP may need further development, such as incorporation with basin-hopping approaches, or to be considered for use as an approximate CSP method for fast applications, rather than as a fully reliable prediction approach.

### 8.5.5 Efficiency Ratios

#### Efficiency in Recovering the $\leq 7.5$ kJ/mol Window

For a CSP method to be successful, it must not only be comprehensive but also efficient - with as low a computational cost as achievable. For some ‘rough and ready’ applications, such as incorporation of CSP with genetic algorithms [181], it may even be appropriate to sacrifice some predictive power of the method in favour of greater efficiency.

Figures 8.22, 8.23, and 8.24 show the efficiency of templating CSP in recovering unique structures in the lowest 7.5 kJ/mol region of the target landscapes from different template sets.

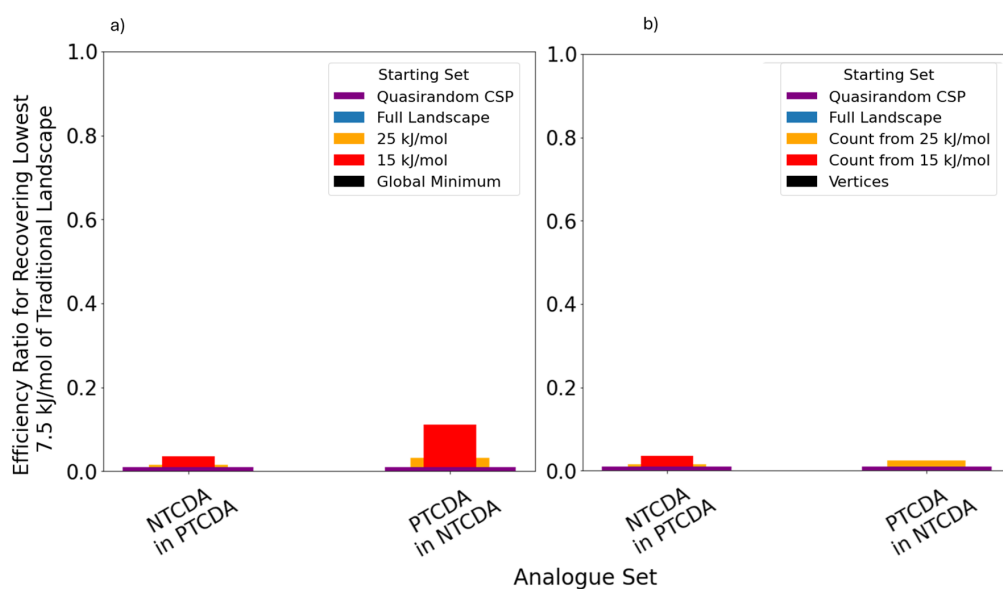


Figure 8.22: Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the second family of NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy



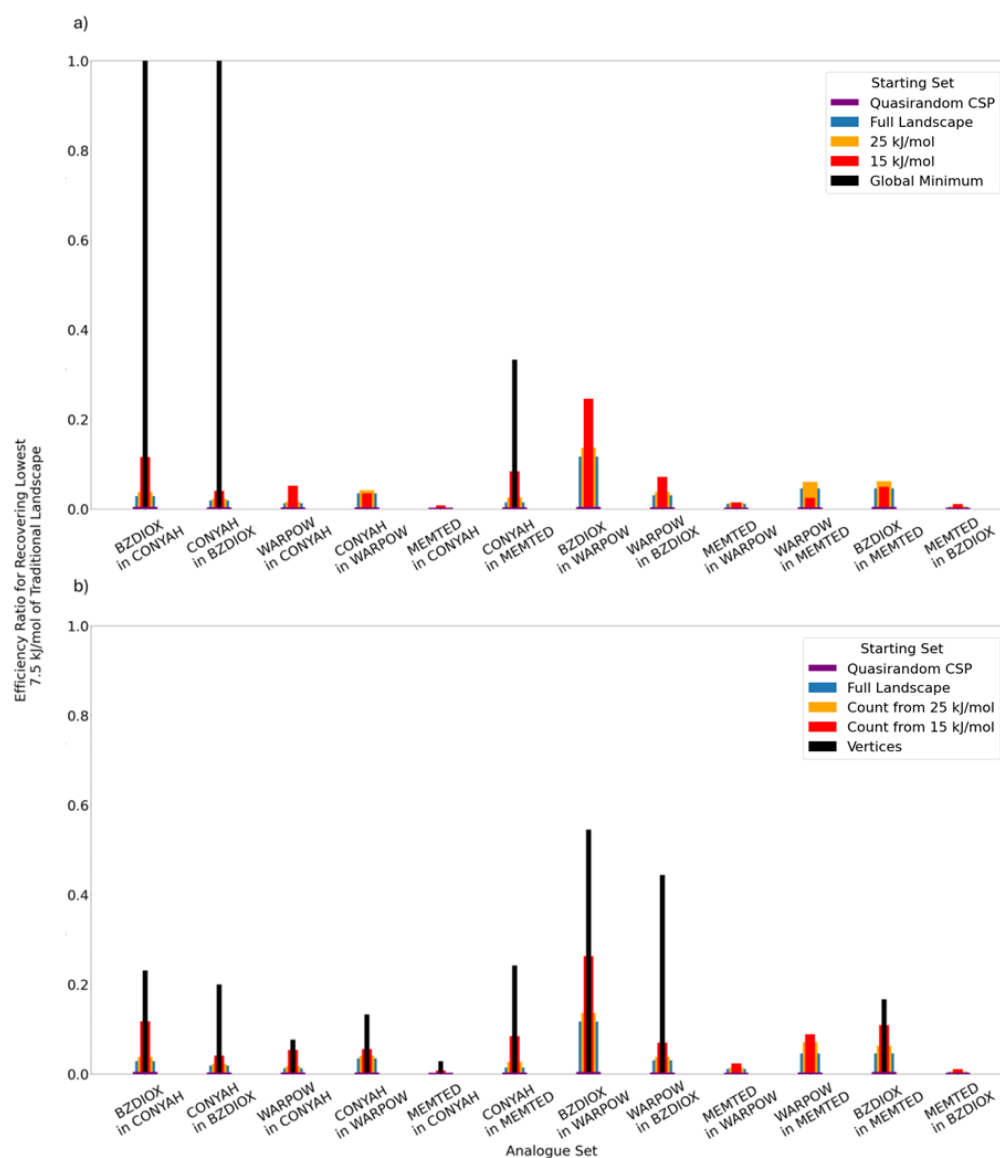


Figure 8.23: Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

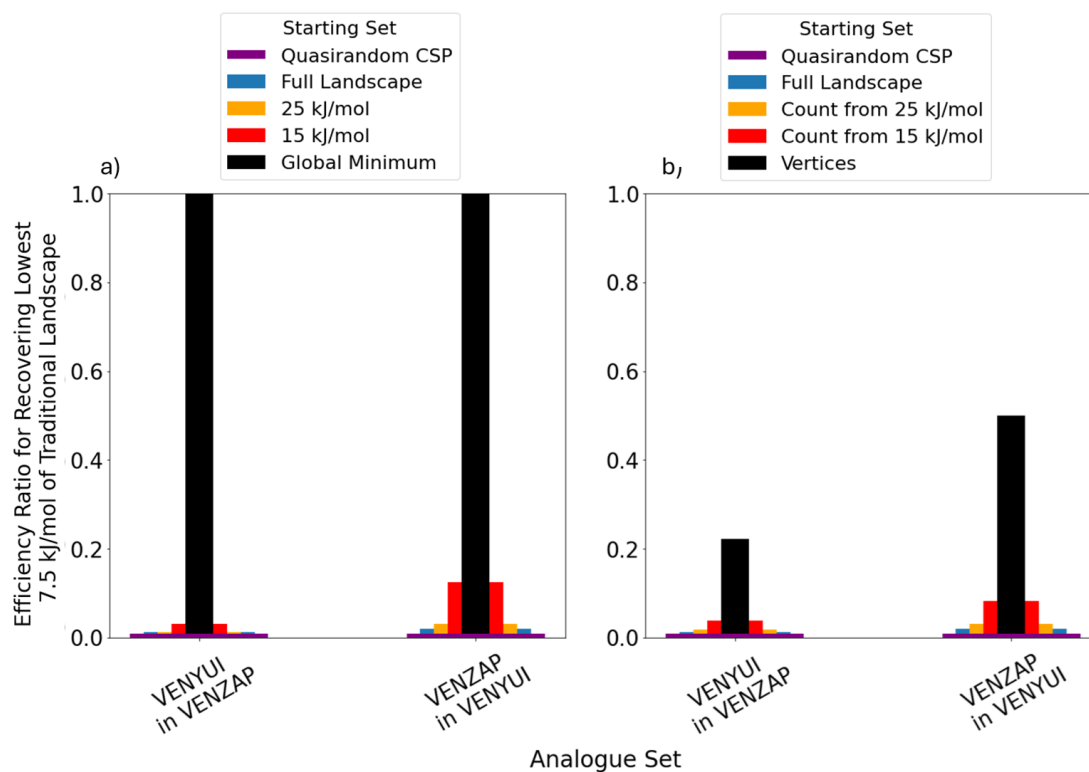


Figure 8.24: Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

Efficiency ratios differ significantly both between template-target pairings and between starting structure selections. The efficiency of the approach appears similar when extracting starting structures based upon dressed energy than when based upon relative energy.

Crucially, however, it can be seen that the templating CSP efficiency ratios are greater than the traditional CSP efficiency ratios in all cases. This is despite the fact that traditional CSP efficiency ratios have been defined generously. The ratios used assumed that the number of unique starting structures minimised was equal to the number of successfully optimised structures requested by the user when running CSP. Due to non-recording of errors, it is unknown precisely how many minimisations that were attempted in order to achieve this. The efficiency ratios for traditional CSP may not have significantly changed if failed minimisations were accounted for, but it is worth noting that the values provided for traditional CSP are likely to be an overestimate.

**Efficiency in Recovering the  $\leq 25$  kJ/mol Window**

Figures 8.25, 8.26, and 8.27 show the efficiency of templating CSP in recovering the 25 kJ/mol window of target landscapes. Again, efficiency can be seen to outperform traditional CSP in all cases. The distinction is far greater when exploring the 25 kJ/mol target window, with several starting sets leading to  $> 50\%$  efficiency, while all traditional efficiencies remain below 1%.

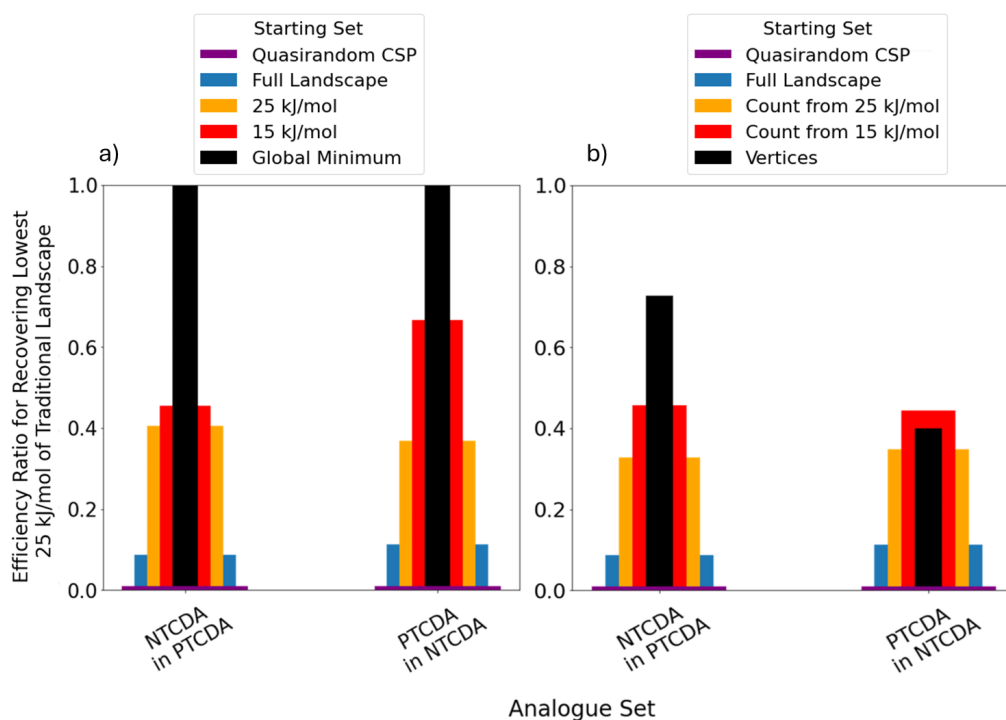


Figure 8.25: Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the NTCDA/PTCDA template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

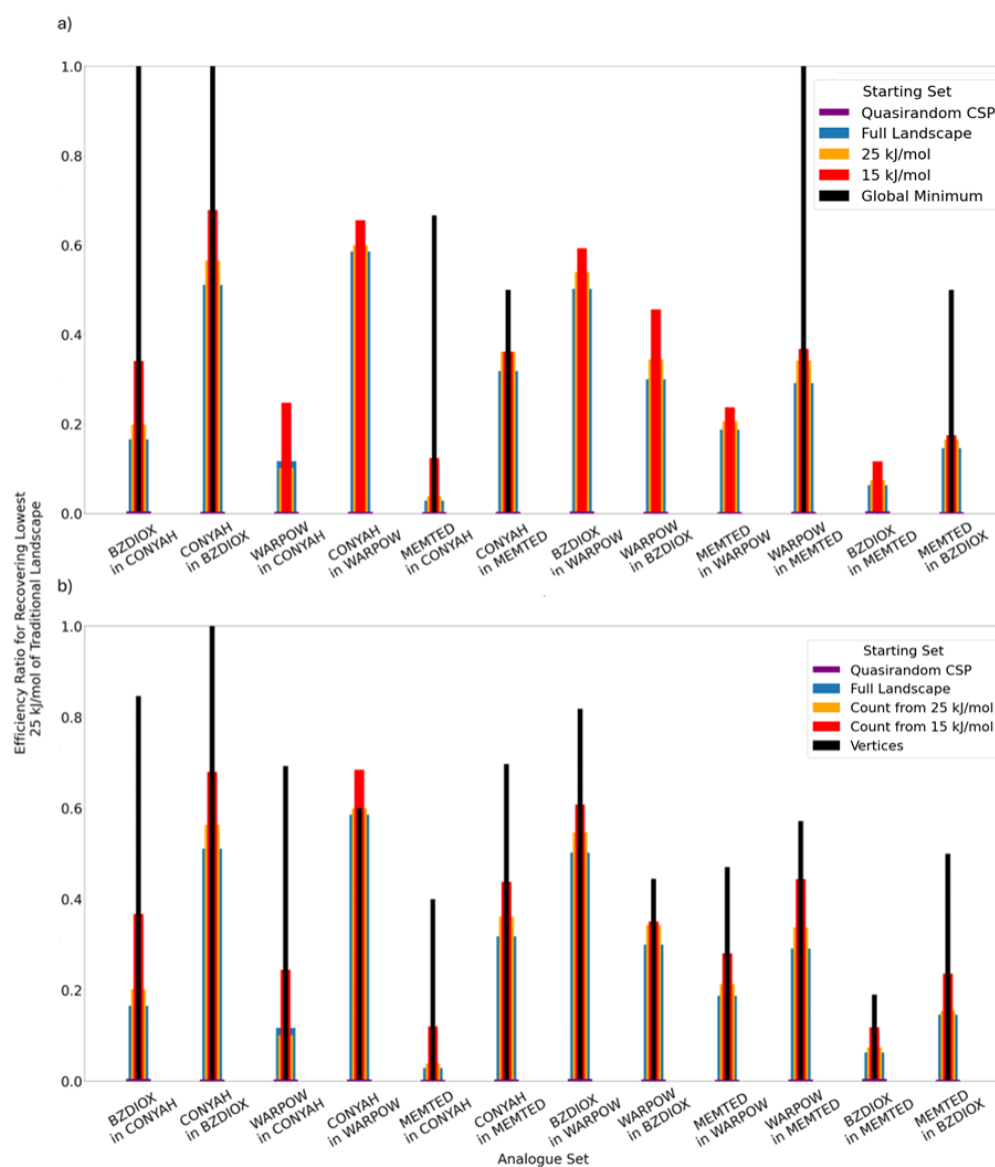


Figure 8.26: Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the first family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

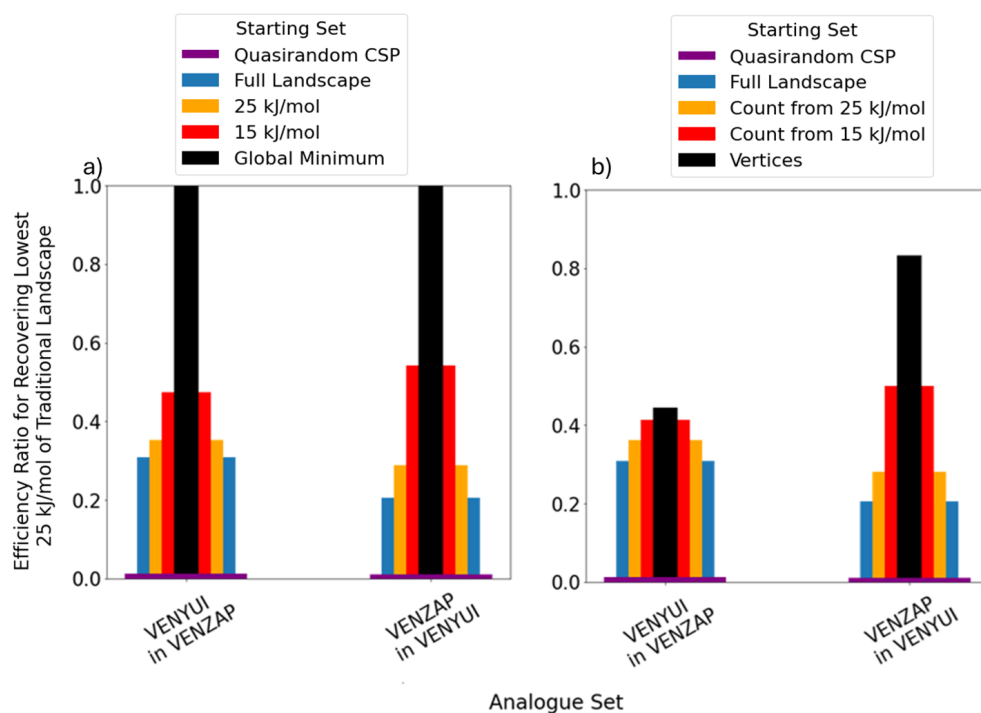


Figure 8.27: Efficiency of templating CSP in recovering the lowest 25 kJ/mol region of the target landscape for the second family of chemically substituted template-target pairs for each trialled starting set as extracted according to a) relative energy and b) dressed energy

The differences in efficiency highlighted by these results are of great importance to a fast-CSP method. Lattice-energy minimisation acts as the main bottleneck in CSP workflows, and so approaches that require fewer minimisations lead to significant speed-ups in the workflow, assuming equal cost of the individual minimisations. One promising result in light of this is that there was a 69 times improvement in the efficiency (0.00126 vs 0.087 efficiency ratio) for templating CSP over traditional CSP in recovering the 25 kJ/mol region of the NTCDA landscape. This is despite maintaining 80% recovery rate. The stand-out results such as these demonstrate strong potential for templating CSP.

### 8.5.6 Distribution of Matches

#### Overview

The statement that templating CSP could in some cases be 69 times more efficient compared to traditional CSP is a striking figure. However, it is known that successful traditional CSP oversamples beyond what is required to recover the low energy landscape. Therefore, it may be that templating CSP results thus far are merely demonstrating what could have been achieved via quasi-random

CSP with smaller sampling. This concern may be heightened by the discovery that there is little distinction between selecting starting structures via relative energy or via dressed energy. That is, the results thus far would appear to show that it does not matter for templating CSP **which** starting structures are selected - only **how many** are selected. However, both approaches have solid basis for starting structure selection, and it may simply be that both are equally successful. To investigate this, it is necessary to explore how both subset selection methods compare to random selection. Future work could explore the comparison between templating CSP and limited quasi randomly CSP more deeply, for instance by comparison of the effectiveness of templating CSP to CSP using limited quasirandom sampling. However, for proof of concept for the purposes of this thesis, investigation merely explored the templating approach - including **guided** extraction of starting templates - compared to **random** selection of templates from the original landscape. That is, it explored whether the position on the original landscape of the starting structures impacted the likelihood of recovering structures on the target landscape.

Qualitatively, this may to be the case, as can be seen from the efficiency ratio figures. The efficiency of sampling increases as the starting templates are extracted from lower energy regions, i.e lower energy template structures are more likely to lead to unique low energy target crystal structures. This suggests that the nature of selection is impactful, and that sampling via templating CSP may therefore outperform small (quasi)random sampling. An additional metric was added to quantify this. The distribution of matches metric explores the number of traditional low energy minima recovered from analogues of low-energy templates relative to the number of recovered structures that would be expected from an equal number of analogues - if formed from randomly selected structures across the original energy landscape. This is given by the distribution of matches ratio (DMR):

$$DMR = \frac{|\text{Set Recovered}|}{\text{Expected Recovery}} \quad (8.1)$$

$$\text{Expected Recovery} = \frac{|\text{Unique Analogue Set}|}{|\text{Full-Landscape Unique Analogue Set}|} \times |\text{Set Recovered by Full-Landscape Templating}|$$

DMR values therefore quantify the ratio between the expected number of recovered structures for a given starting set - based upon the full landscape templating recovery - and the actual number of recovered structures.

If there is a meaningful relationship between original and target landscapes guiding the success of templating - i.e if low energy structures are more likely to lead to low energy analogues - then

the number of structures recovered via templating from a low energy starting set should be greater than the expected recovery. A distribution of matches ratio greater than 1 for low-energy starting sets therefore indicates meaningful promise of the templating approach.

Of course, the expected number of recovered structures will not always be possible - i.e it is not possible to recover a non-integer number of structures. But the expected recovery can be viewed as the average number of recovered structures expected for a given size of starting set, e.g. an expected number of structures  $\ll 1$  renders recovery unlikely.

### Results

Figure 8.28 represents the DMR values for each template-target pair as heatmaps - considering the performance from different starting sets, and targeting the 7.5 kJ/mol and 25 kJ/mol regions of traditional target landscapes.

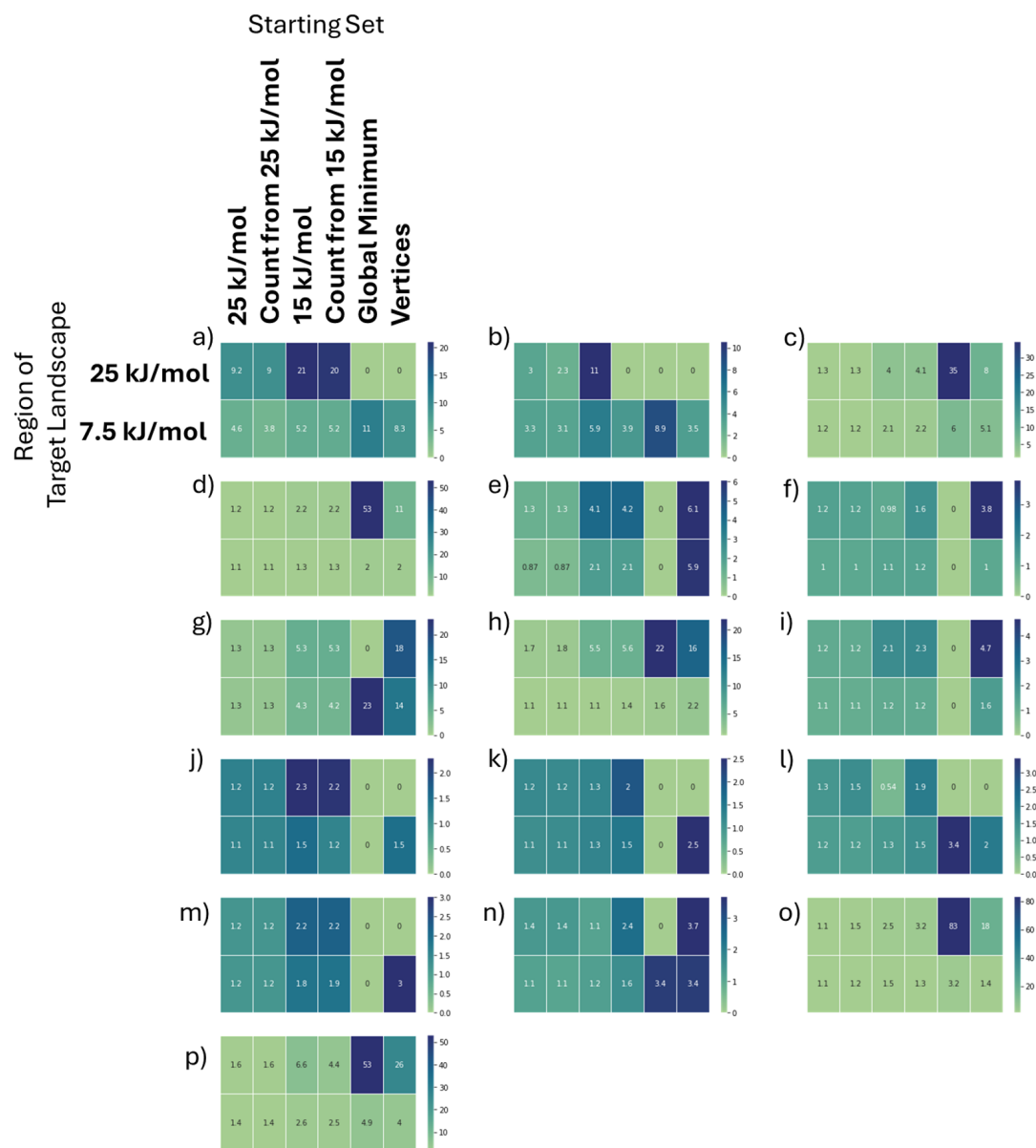


Figure 8.28: Heatmaps for each template-target pair, showing the DMR values of templating for each region of the target landscape and each trialled starting set. Maps for each system are given by a) NTCDA in PTCDA, b) PTCDA in NTCDA, c) BXDIOX in CONYAH, d) CONYAH in BZDIOX, e) WARPOW in CONYAH, f) CONYAH in WARPOW, g) MEMTED in CONYAH, h) CONYAH in MEMTED, i) BZDIOX in WARPOW, j) WARPOW in BZDIOX, k) MEMTED in WARPOW, l) WARPOW in MEMTED, m) MEMTED in BZDIOX, n) BZDIOX in MEMTED, o) VENYUI in VENZAP, p) VENZAP in VENYUI



The data confirms the trend that low energy templates are more likely to lead to unique low energy analogues - with almost all substantial starting subsets leading to DMR ratios  $> 1$ , i.e greater than expected recovery. Looking left to right across the heatmaps, it can be seen that the DMR values tend to increase towards the right - as templating uses lower-energy starting sets. This suggests that the CSP landscapes of similar molecules are likely to be related - sharing not just similar structures, but similar distributions of those structures across the energy landscape.

Unsurprisingly, cases in which vertex or minima- based starting sets have recovered target landscape structures lead to very high DMR ratios - with values reaching as high as 83. These represent cases in which on average, it was highly unlikely for a single starting structure or very small starting set to lead to recovery of a low-energy target landscape structure - but templating CSP still achieved this. It should be acknowledged that in cases with small starting sets, the expected recovery will be very low, and so even a single recovered structure leads to large DMR values - presenting something of an ‘all or nothing’ issue in the results from such sets. However, the appearance of these cases despite their low average likelihood still suggests a meaningful advantage to using low-energy starting structures - and so a relationship between original and target landscapes - especially given the number of such cases found.

Following the trend of prior results, there is little distinction between the cases of starting structure extraction via the hull or based upon relative energy.

One caveat to note is that in all cases in which no structures are recovered, the distribution of matches ratio will be equal to zero by construction, regardless of the number of expected recovered structures. Therefore, whilst the DMR data can be used to identify trends and compare the target landscape recovery from different successful starting sets, it cannot be used to discriminate between failed starting sets.

## 8.6 How Broadly Can Templating Be Applied?

It is natural at this point to question **how** similar molecules must be for templating CSP approaches to be effective. This is especially interesting in light of the previous investigations into the ‘box-model’ suggesting that molecules of merely similar shape may adopt similar crystal structures [179]. This question was not investigated extensively in this thesis, however, a short test was performed designed to push templating to its limits, by attempting inter-family templating CSP. Work predicted crystal structures of MEMTED via substitution of MEMTED molecules into templates from predicted NTCDA crystals (Figure 8.29). Figures 8.30 and 8.31 show the percentage recovery and efficiency ratios for the lowest 7.5 kJ/mol region of the target landscape, alongside the success of intra-family templating.

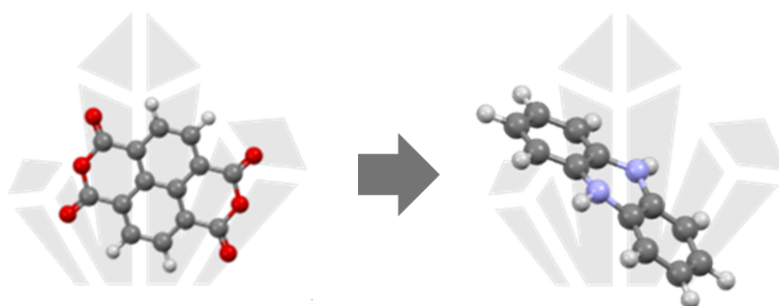


Figure 8.29: A challenging case for which templating was tested - despite key molecular differences

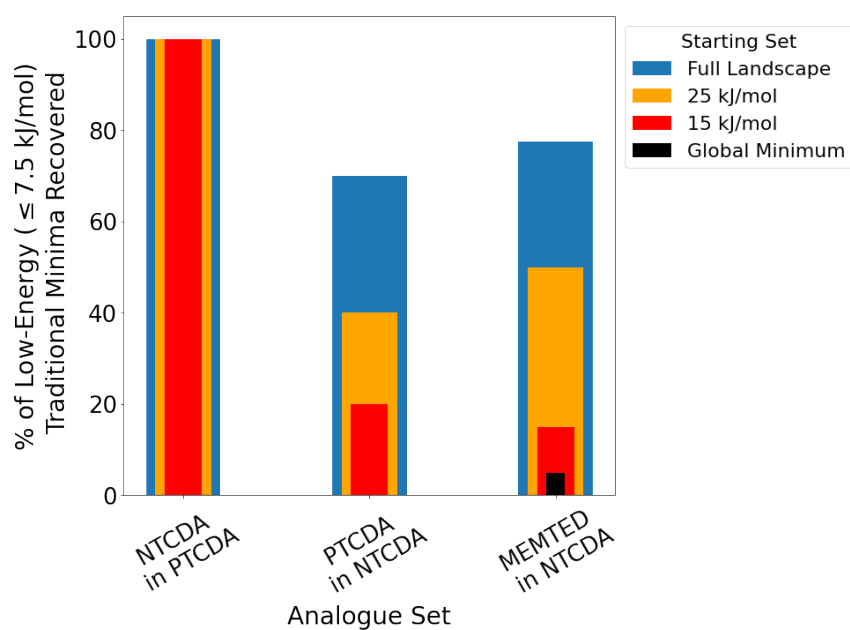


Figure 8.30: Percentage of the lowest 7.5 kJ/mol region of the target landscape recovered from templating CSP of the MEMTED/NTCDA and NTCDA/PTCDA template-target pairs for each trialled starting set

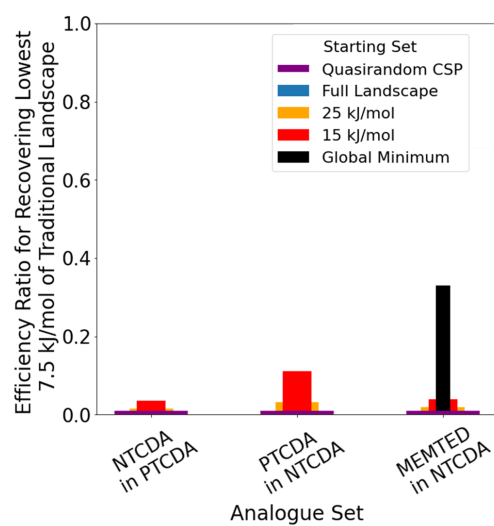


Figure 8.31: Efficiency of templating CSP in recovering the lowest 7.5 kJ/mol region of the target landscape for the MEMTED/NTCDA and NTCDA/PTCDA template-target pairs for each trialled starting set.

These findings demonstrate that the performance of intra-family templating in this case is on-par with inter-family templating. Further investigation into the reasons for this is needed. One possibility, however, is that the approach is more broadly applicable than originally thought. This may lend credence to theories such as the ‘box model’ that emphasise the surprisingly limited number of packing patterns adopted by organic molecules [179], and suggest wider promise of templating approaches.

## 8.7 Considerations and Concerns

### 8.7.1 Overview

As discussed, the templating CSP approach developed is presented as a proof-of concept and requires further development before implementation. Work aimed to formulate a generalisable approach to templating. However, limitations to this generalisability - of which some were known and others encountered during investigation - are discussed below.

### 8.7.2 Templates with $Z' > 1$

If a template has  $Z' > 1$ , this increases the number of possible analogues that could be formed, as the number of ways of forming valid analogues includes not only all reasonable substructure overlays of the old and new molecules, but all combinations of these overlays across the molecules of the respective asymmetric units.

Accounting for this would require adaptation of the existing code base in order to enact multiple defined substructure overlays within one crystal, and to enforce each required overlay onto a specific pair of molecules from the asymmetric units of the two crystals. The conceptual changes needed are not overly complex, but it is currently unknown how difficult the code implementation may prove.

### 8.7.3 Flexible Molecules

Where analogous molecules to be used in templating have flexibility, this introduces a further complexity to the task of producing analogues because if the molecule to be entered into a template has rotatable bonds, then it must be determined which molecular conformation should be added to the template. There could be several philosophies as to the best way of approaching this, and this may be considered in future work.

### 8.7.4 Substructure Limitations

#### Limitations of Graph Isomorphisms

One limitation of the approach that became apparent during the work is that whilst the use of isomorphic mappings of the molecular graphs is necessary to capture all valid molecular overlays, this can also incorporate additional overlays, which are unreasonable. For instance, if an isomorphic mapping or ‘renumbering’ of the atoms could not be achieved in physical space via rotation, and instead relies upon inversion or reflection, then the corresponding molecular overlay may be poor

- and not indicative of a meaningful analogue. Figure 8.32 shows an encountered example of this problem, in which a valid isomorphic mapping of the identified substructures of the two molecules would have led to the overlaying of the indicated atoms with each other. Such overlaying of the indicated atoms would be achievable via ‘flipping’ of one molecule, however, as the molecules form cages capped at both ends, when considering the entire mapping the corresponding overlay is neither feasible nor valid.

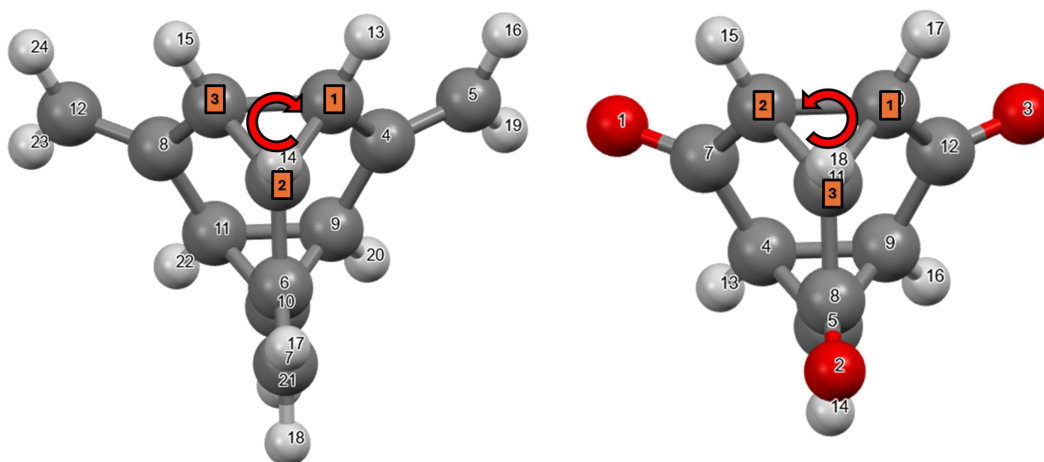


Figure 8.32: A case for which an isomorphic mapping between shared substructure instances in two molecules did not correspond to a viable overlay. The mapping required the atoms indexed in orange boxes to be overlaid (i.e atom 1- atom1)

However, it is also precisely this phenomenon that has allowed for consideration of all valid analogues in the case, for example, of CONYAH in BZDIOX (Placing a ‘bent’ molecule into a template crystal from a planar molecule) - this avoids arbitrarily inserting the bent molecule into the template ‘one way up’.

### Limitations of Substructure Identification

A further concern is that identified substructure instances may not correspond to meaningful shared molecular geometry - as the identification relies simply upon connectivity. Figure 8.33 shows an example of this problem, whereby the substructure overlay approach would in one case overlay the atoms according to the indexing shown in orange boxes - i.e overlaying atom 1 with atom 1 etc. This is indeed a case of shared substructure - the CHN containing chain indicated - but due to the differences in molecular geometry, this does not correspond to an **intuitive** or superimposable case of shared substructure.

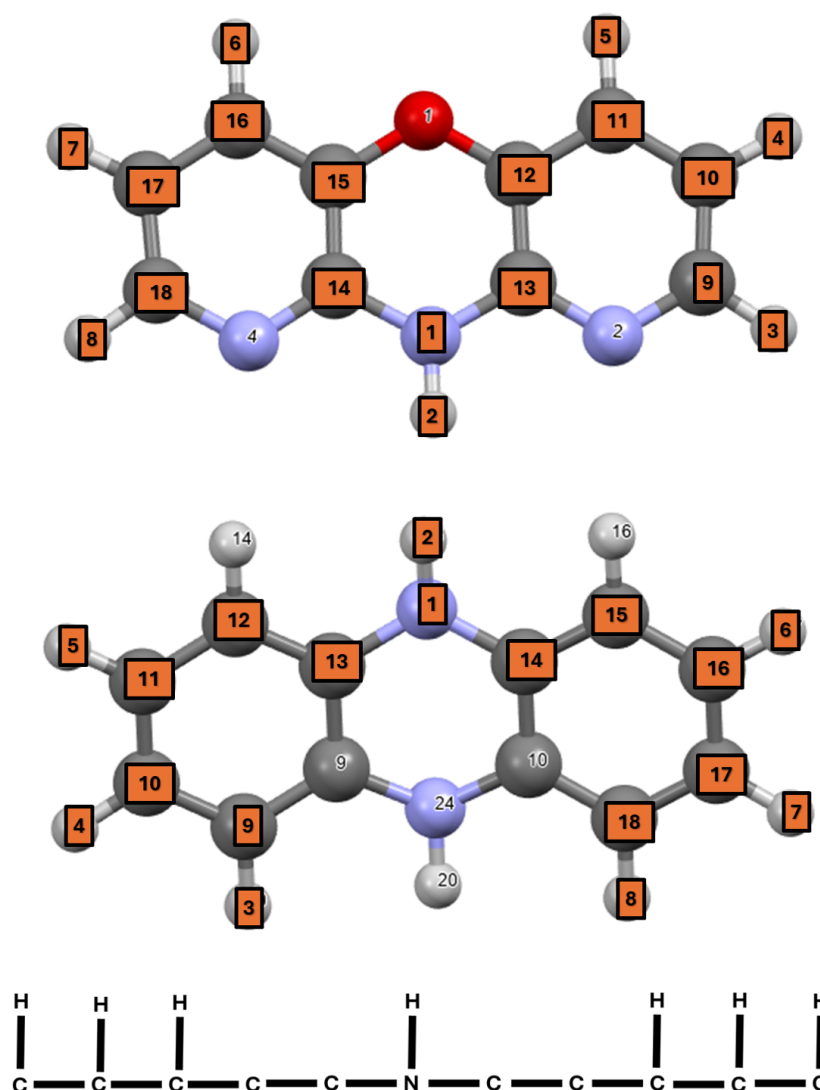


Figure 8.33: A case for which identified shared substructure - the indicated CHN chain- did not correspond to *intuitive* shared substructure (i.e shared motifs). The identified substructure instances in each case are labelled in equivalent order by the indexing in orange boxes

## Addressing Issues

For the purposes of the proof-of-concept in this thesis, these issues were not directly resolved. Instead, an RMSD criterion was enforced upon the overlays, and analogues were only formed where the corresponding substructure overlay was found to have an  $\text{RMSD} < 0.2 \text{ \AA}$ . It was assumed that substructure overlays achievable with an RMSD below this threshold would be theoretically reasonable, or their deviation from reasonable behaviour negligible. Therefore only analogues arising from valid substructure overlays were included in the results. Resolution of the issues was left to

future work, though some brief considerations are given below.

It could be said that substructure overlay is not always appropriate. One alternative approach would use off-the-shelf alignment software such as *align-3d*[180]. However, this approach does have its own limitations. For example, the selection of a single ‘best fit’ overlaying would not allow for construction of all valid analogues - and would be arbitrarily restrictive. Further, given its shape-based fitting, it may be overly impacted by molecular conformation. It is however possible that expansion upon this technique, coupled with considerations to enforce all valid overlays, could prove effective.

If continuing with the approach based on overlaying of substructure, it may be possible to identify cases where shared substructure represents ‘true’ and intuitive shared molecular geometry - for example based on feature identification via SMILES strings. It may also be possible to restrict the considered isomorphic mappings to those achievable via application of rotational symmetry operations.

### 8.7.5 Defining the Analogue

The question of what defines crystal structures as truly ‘analogous’ is to a certain extent a philosophical one. There are two main aspects of this question that were encountered during this work - namely whether molecules between claimed analogues should share centroid positions, and whether the analogues should share similar intermolecular separations.

#### Shifting Centroids

As discussed in Section 8.4.1, forming analogues by substructure overlay can result in structures in which the centroids of the molecules are positioned differently to in the template crystal. This can be corrected by shifting the molecules to regain the molecular centroid positions of the template.

Shifting the centroids, however, increases the duplication of analogues as more of the structures arising from different isomorphic overlays become equivalent. As analogue formation is quick and inexpensive, and duplicates can be efficiently removed using pXRD clustering prior to optimisation, the duplication problem is not one of cost. However, it does decrease the diversity of sampled initial structures - which could lead to poorer recovery of the target landscape.

Further, there could be debate as to the motivation behind positioning molecules such as to create



analogues worthy of investigation in templating CSP. Positioning molecules such that analogues share the geometric centroid positioning of their constituent molecules is one approach. However, an alternative philosophy would be to formulate analogues such that the positions of the important functional groups of the molecules are maintained (i.e without correcting the positions).

Work in this thesis formulated and investigated analogues that **did** share centroid positions, but the alternative could be investigated.

### Intermolecular Spacing

One question in defining what should be considered an analogue is that of whether analogues must share not just similar patterns of packing, but similar densities of packing. Whilst orienting the molecules and positioning their centroids to match those of the original structure goes a long way to defining an analogous packing, using a new molecule that has a different size along any given dimension can create an analogue that is more sparsely or densely packed. This creates analogues that have, in essence, the same packing pattern - but have been ‘spread out’ or compressed.

It would therefore represent a more intuitive analogue to re-adjust this intermolecular spacing. This task is not as straightforward as it may seem, as it first relies upon defining a useful measure of the spacing. Further, the required ‘adjustments’ to the spacing in-order to define a closer analogue would not necessarily align with a cell vector, making the task of reducing the cell and shifting the molecules more complex.

Initial work did look at exploring the average intermolecular spacing, along each unit cell axis, as a means to identifying dimensions that should be elongated or compressed. However, the average proved difficult to meaningfully define due to the presence of negative intermolecular ‘separations’ along any given axis.

Future work could potentially explore such corrections further, or it could be assumed that the optimisation process will reduce the differences, and progress towards a more analogous minimum.

#### 8.7.6 Optimisation

The true computational bottleneck in a CSP workflow is not the generation of trial structures, but rather the geometry optimisation of those structures. Therefore, it is too naïve to simply state that a

reduced number of required minimisations for successful CSP equates to a more efficient method - instead the overall minimisation **cost** must be reduced.

Due to time constraints, the relative minimisation time required for templating CSP was not investigated as a means of validating the proof of concept. Such investigation would not necessarily detract from the results in this thesis, indeed it is possible that beginning from a structure informed by previous predicted minima, may lead to faster geometry optimisations. However, this is a factor that should be tested before templating CSP is proposed as a ready alternative to quasi-random CSP methods.

## 8.8 Concluding Remarks

Work in this chapter constructed a proof of concept for the use of templating for fast CSP. A workflow was developed in which previously predicted crystal structures were extracted from a CSP landscape, overlaying of molecules by shared substructure was used to form analogous crystal structures containing a similar molecule, and those analogues were used as the trial structures for CSP of the new molecule. The performance of the approach was tested with regard to both its ability to recover structures that would have been predicted via more traditional methods, and the efficiency of the method in recovering those structures. This was tested across a range of similar molecule pairings and starting sets of initial templates extracted to form analogues. Work also explored the impact of selecting template structures according to their dressed energy - i.e via a Generalised Convex Hull, as compared to according to their relative energy.

Templating CSP, was found to have potential in recovering the low-energy regions of traditionally predicted CSP landscapes, validating the proof of concept for further investigation. However, the recovery of the wider CSP landscapes proved poorer - suggesting that the approach would be insufficient for full, confident solid-form screening. Further, the performance varied significantly between chosen molecule pairs.

It was found that, expectedly, the recovery was poorer when using smaller starting sets of templates extracted from low-energy windows on the original CSP landscape. However, the performance in these cases was greater than expected given the relative number of analogues formed from the sets - indicative of a relationship between the CSP landscapes of similar molecules.

Notably, the efficiency of the approach in identifying unique low-energy structures was significantly greater than quasi-random CSP in almost all cases - the exceptions being cases in which very few templates were used - reinforcing the promise of the approach, especially as a quick and approximate method.

However, little distinction was found between the cases in which templates were extracted based upon their relative-energy ranking and cases in which templates were extracted based upon their dressed energy ranking - meaning that incorporation of a GCH may not prove beneficial to the method.

Unfortunately, several issues were discovered with regard to the generalisability of the approach,

largely relating to the inappropriate derivation of sub-structure overlays in some cases - though this should not have impacted the results shown.

Future work should aim to address the identified issues, before broadening exploration of the approach. Possible avenues include incorporation of templating CSP with basin-hopping methods, or use of templates from multiple landscapes.

## **Chapter 9**

# **Conclusions and Future Work**

### **9.1 Overview**

This chapter acts as a summary of the thesis. First, the key findings and limitations of the work are discussed. Then possible avenues for further development and application of the work are explored.

## 9.2 Work Summary

Work in this thesis explored two key areas:

1. Construction and evaluation of a global SOAP kernel to analyse the similarity of predicted molecular crystal structures
2. Development and testing of a proof of concept for fast CSP using analogy to previously predicted structures of similar molecules

Construction of a new global SOAP kernel - called the adapted kernel - was achieved by creation of a wrapper around an average SOAP kernel calculation code in order to derive an ‘adapted’ average SOAP kernel. This was performed by limiting the included local kernels - i.e the considered atom-atom similarities so as to be theoretically reasonable. This relied upon identifying atoms in the respective crystal structures that could be said to represent the same atom of the underlying molecule, including those analogous when considering point group symmetry of the molecule.

The utility of the kernel was then assessed in terms of its efficiency in identifying stabilisable crystal structures via the GCH, the interpretability of ML descriptors derived from the kernel, and the performance of machine learning for energy prediction using the kernel. The ability to ‘identify stabilisable structures’ was assessed via the ability to identify observed structures as stabilisable. In these tests, the performance of the adapted kernel was assessed relative to that of the average kernel.

It was found that ML descriptors derived from the adapted kernel tended to be more closely related to intuitive descriptors of crystal structures than those derived from the adapted kernel. This was apparent both through visualisation of ML descriptor - intuitive descriptor relationships and through systematic evaluation of the relationship strength via machine learning. The advantage of the adapted kernel in this regard was particularly apparent for the case of the ROY system, where a classification of the underlying molecular conformation could be predicted from a single ML descriptor with notably greater than 0.8 balanced accuracy when using the adapted kernel, compared to  $\sim 0.8$  or lower balanced accuracy when using the average kernel. This improved performance was maintained regardless of the SOAP cut-off used. The findings suggest that the kernel may result in more interpretable ML descriptors, which is a positive sign with respect to the reasonableness of the kernel construction. It could also make the kernel more useful in materials discovery as it could help to propose constraints or selectively synthesising certain crystal structures. However, the differences in the ML-intuitive descriptor relationship strengths between

kernels in many cases were subtle.

With regard to different kernels' utility in machine learning of energies, it is possible that the adapted kernel may be able to provide accurate energy predictions over a wider energy range than the adapted kernel. When making energy predictions for a low-energy dataset, the performance gap shrank - though the adapted kernel still significantly outperformed the average. For example, when predicting energies of structures within a 30 kJ/mol window, use of the adapted kernel could achieve errors of  $1.459 \pm 0.034$  MAE compared to errors of  $1.780 \pm 0.046$  MAE using the average kernel. This is a positive result for the assessment of the adapted kernel construction, though the required training set sizes for satisfactory prediction across the full energy range (3500 training structures for the adapted kernel and  $> 6400$  for the average kernel) indicated that improvement may be needed for both implementations.

Unfortunately, results regarding the efficiency of identifying synthesisable structures were more complex and less conclusive. The efficiency of the GCH in identifying stabilisable crystal structures did not appear to be consistently improved by the kernel adaptations. In some cases the adapted kernel outperformed the average for identifying observed structures as stabilisable but the converse was true for other cases. In some cases, such as for the systems of T2 or galunisertib, results suggested that the GCH approach - using either kernel - could be more effective at identifying synthesisable structures than other landscape analysis approaches, but in other cases, such as for the ROY system use of a GCH in place of traditional landscape analysis methods increased the candidate pool.

Comparisons were further complicated by noted variability with underlying SOAP cut-off and with the dimensionality of hull constructions. Initial significance tests also suggested that the noted differences in candidate pool sizes may not be reflective of any meaningful performance gap between GCH approaches using different kernels.

These mixed results with regard to kernel comparisons are disappointing and perhaps unexpected. It would not be unreasonable to anticipate that a more theoretically reasonable kernel construction may lead to greater performance - but this could not be proven here.

Overall, the findings are not informative as to a 'superior' kernel construction with regards to efficiency at identifying stabilisable structures from prediction sets. But the kernel adaptations

have shown potential for purposes useful to materials discovery - in leading to interpretable ML descriptors and showing promise in energy prediction over a wider range of energies than the average SOAP kernel. The findings also underline the sensitivity of several approaches in materials discovery to the underlying construction of any similarity kernels used. This therefore suggests that results, such as stabilisability predictions using the GCH, should be viewed with appropriate caution and that researcher expertise may be needed in selecting a similarity kernel suited to the task at hand.

The use of analogy to previously predicted structures for fast CSP (templating CSP) showed more promising results. Work developed a proof of concept for templating CSP in which trial crystal structures of one molecule were created by forming analogues of previously predicted structures of similar molecules. Analogue formation relied upon ‘replacing’ the molecules in the unit cell, situating the new molecules based on their substructure overlay with the previous molecules. This was performed so as to form all valid analogues. Trial structures were then lattice energy minimised as in traditional CSP workflows.

The proof of concept was assessed based upon its ability to replicate low-energy CSP landscapes generated via more conventional methods, and its comparative efficiency in doing so. The approach performed well in both regards, being able to recover large percentages of the low energy landscape of many systems with significantly fewer lattice energy minimisations than quasi-random CSP. For example, one successful templating case recovered 89% of the lowest 7.5 kJ/mol of the target landscape using just 9529 unique analogues as trial structures, whereas the original traditional CSP search sampled 260,000 structures as trial structures. Some testing, albeit preliminary, suggested that the increased efficiency found was indeed due to a meaningful relation between the CSP landscapes of similar molecules.

There was, as expected, a variation in the performance dependent upon the pairs of similar molecules chosen. However, positively, a case in which the molecules has significant dissimilarities, still proved to be a success of the templating CSP method. This suggests that the method could be more widely applicable than previously thought. Overall, the findings suggest that - as a proof of concept- the approach has promise in facilitating fast and efficient CSP. The approach warrants further development to explore its full potential in materials discovery.



### 9.3 Issues

There remain some outstanding issues and limitations to the work.

The adapted kernel construction suffers from limitations to its generalisability - not being applicable to  $Z' > 1$  crystal structures of symmetrical molecules. Further, the neglect of local symmetry leads to some potentially reasonable local kernels being excluded from the global kernel construction.

The assessment of the adapted kernel also suffers from a lack of comparisons to the ReMatch kernel. This was due to a need to focus research efforts on development and on comparisons to the average kernel - with the adapted kernel (especially in its ‘average of possibilities’ formulation) being viewed as an altered average kernel. The value of comparisons to the performance of the ReMatch kernel is acknowledged and could be considered in future work. However, the results so far remain of interest as initial kernel comparisons. It can also be said that if the potential unreasonable local kernel sets used in ReMatch kernel construction are common, then the adapted kernel development remains of interest due to its theoretically reasonable construction alone. Should such comparisons prove rare, the adapted and ReMatch kernels may behave similarly.

There are also limitations to the templating CSP approach, particularly with regard to identification of reasonable overlays. The current approach could lead to inclusion of unreasonable analogues in some cases.

Due to time limitations, and ongoing code developments throughout the work, the analysis of both the adapted kernel construction and the proof-of-concept for templating CSP suffers from a lack of discussion of computational cost.

## 9.4 Potential for Future Work

One useful aspect of future work could be to address the aforementioned limitations of the methods, seeking to make the adapted kernel construction more generalisable and the approach to identifying reasonable molecular overlays for templating CSP more reliable. Further adaptations to the respective codes could also be made to optimise the efficiency and to create a more pythonic, and understandable codebase that could be used by a wider audience. One key element of such code developments would be to integrate the functionalities into the existing CSPy codebase.

Following final development of the codes, analysis could also be improved by exploration of the computational cost of the developed methods as well as comparison of the adapted and ReMatch kernels.

Further applications of the GCH method could also be investigated. One application could be in proposing diverse but likely stabilisable crystal structures to contribute to genetic algorithms for identifying promising molecules. Genetic algorithms (GA) are an approach, inspired by evolution, for discovering new functional materials. Genetic algorithms to propose new molecules have been used in the Day Group at the University of Southampton. This relies upon an iterative process of generating ‘new’ molecules by mutation or combination of the most promising molecules from previous steps. Work has shown that this can be improved by considering not just molecular properties but also likely solid state properties, when identifying the most promising molecules [181]. This leads to an interesting potential application of the GCH. If a required solid-state property is expensive to calculate then - on the timescale required for genetic algorithms - property calculations could only be performed for a small subset of predicted structures of each molecule. The GCH could be used to identify useful subsets of predicted structures for property calculations. The resulting predicted properties could then be averaged across the structures identified by the GCH to give the solid state property value used to identify promising molecules at each step of the GA.

The most promising avenue for future work is in the applications of templating CSP. One interesting area of investigation could be to explore how to identify promising pairs of systems for use in templating. Especially given the move of CSP towards ‘big data’, large numbers of CSP landscapes -i.e previously predicted structures- are becoming available. For example, one recent study produces CSP landscapes of over 1000 small organic molecules [54]. This data could be of great use in facilitating fast CSP via templating approaches. However, this requires a method/set

of rules for identifying which landscape to use as templates for a given prediction problem. Exploring possible methods/rules could be an interesting avenue of future work.

One further area of investigation for templating CSP could utilise recent work to identify the importance of different atomic and functional group interactions in determining crystal structure. This work identifies the importance of given interactions based upon their contribution to intermolecular energy, according to a metric derived via machine-learning [182]. An interesting area of investigation could be whether the determined importance of certain atoms/functional groups to determining intermolecular interactions aligns with which molecular pairs lead to successful templating CSP. For example, whether or not it is still possible to template from one system to another if an ‘important’ group has been changed between systems. If the methods are found to be aligned, this could offer a possible step towards identifying template systems to use in a given prediction problem.

Lastly, given the potential seen for templating CSP to quickly perform predictions, it could be used in applications where speed-ups to the prediction process are crucial. This could be cases in which CSP would typically require extensive quasi-random sampling or in applications where unusually fast CSP is needed, such as use of crystal structure prediction in genetic algorithms.

One particular application of interest could be in the prediction of crystal structures of porous salts. Porous salts have recently shown promise in capture of guest molecules - with possible applications to, for example, nuclear waste management. However, prediction of their crystal structures - which is crucial to their application - is complex and costly. This high-cost acts as a barrier to using genetic algorithms to identify promising porous salts. If not for the cost limitations, genetic algorithms could be a promising avenue for their development - optimising properties such as channel sizes and geometries. Fortunately, some porous salts, containing different organic molecules, were found to be iso-reticular [183]. This suggests that templating CSP could be a promising approach to their solid state structure prediction. If templating CSP can be used for these systems to quickly predict their crystal structures - this could facilitate the use of genetic algorithms in their design - leading to improved development of these materials. The use of templating CSP for porous salts - which would require some alterations and expansion of the method - is therefore an area of great interest in future work.

## Appendix A

# SOAP Parameters

### A.1 Parameters used in SOAP descriptor calculations

Parameter Controlled	Value Used
soap_type	'PowerSpectrum'
interaction_cutoff	User selected for each kernel
max_radial (Number of radial basis functions)	8
max_angular (Maximum angular momentum number $l$ used in the expansion)	6
Gaussian_sigma_constant (Å)	0.3
Gaussian_sigma_type	'Constant'
cutoff_smooth_width (Å)	0.5
radial_basis	'GTO'
inversion_symmetry	True
normalize	True

Table A.1: Table of SOAP descriptor parameters used within kernel calculations in this thesis. See librascal [113] documentation for further information

## Appendix B

# Key Scripts

### B.1 Script to Enforce Consistent Atom Indexing in Structure Sets

```
#####
#Reindexing Script – calls upon pre-existing functionality in CSPy
#Reindexes crystal structure files to match ordering in a reference molecule
#####

#####
#perform necessary imports
#####
from cspy.crystal import Crystal
from cspy.chem import Molecule
from zipfile import ZipFile
from multiprocessing import Pool
from functools import partial
#####

#####
#define function
#####

def reindex(struc, ref_mol):
    replace_mols=[]
#Include zipfile here for extracting structures, optimally this should not be hard ↘
→coded
        with ZipFile(<zip file of crystal structures>) as structures_zip:
            structures_zip.extract(struc)
            name_root=struc.split('.')[0]
#get the asymmetric unit of the crystal structure
            structure = Crystal.load(struc)
            molecules = structure.asym_mols()
```

```

#reindex the asymmetric unit molecules - by overlaying with reference
for i in range(len(molecules)):
    old_molecule = molecules[i]

    try:
        overlay = ref_mol.overlay(old_molecule)
        new_mol = overlay[0]
        new_name = 'new_' + name_root + '_mol_' + str(i) + '_xyz'
        new_mol.save(new_name)
    except:
        print('overlay fail:', struc)

#Using molecular replacement in CSP, replace molecules in original with the reindex \
→molecules
    try:
        original_structure = Crystal.load(struc)
    except:
        print('load crystal fail:', struc)

    for i in range(len(molecules)):
        load_name = 'new_' + name_root + '_mol_' + str(i) + '_xyz'
        mol = Molecule.load(load_name)
        replace_mols.append(mol)
    new_structure = original_structure.replace_molecules(replace_mols, \
→reorder_to='other')
    replaced_name = 'replaced_' + name_root + '.res'
    new_structure.to_shelx_file(replaced_name)
#####
#Running the process section
#####

#read in structure file of reference molecule with desired indexing
ref_mols= Molecule.load(<reference molecule xyz file>)

with ZipFile(<zip file of crystal structures>) as structures_zip:
    struc_list = structures_zip.namelist()

#Parallelised over cores, read in structures one at a time and reindex the file
with Pool(int(40)) as pool:
    reindexer=partial(reindex, ref_mol=ref_mols)
    jobs = pool.map(reindexer, struc_list)

```

## B.2 Script to Calculate Adapted Kernel for Systems with Rigid Underlying Molecules

```
#####
#adapted kernel generation script
#incorporates code adapted form the original gch libraries (ref [164])
#and the dch class of the scikitmatter library (ref [110,111])
#alongside unique code
#####
#IMPORTS SECTION
#####

import sys
import os
import time
from multiprocessing import Pool
from functools import partial
import argparse

import numpy as np
from pymatgen import io
from pymatgen.io import xyz
from pymatgen import symmetry
from pymatgen.symmetry import analyzer

import os
import ccdc
from ccdc import descriptors
import ase
import ase.io as aseio
from ase.spacegroup import crystal
from ase.symbols import Symbols
import rascal
from rascal.representations import SphericalInvariants as SOAP
from rascal.neighbourlist.structure_manager import (mask_center_atoms_by_id)
from zipfile import ZipFile

import itertools

import collections
import pandas as pd
import scipy.linalg as salg
from scipy.spatial import ConvexHull as chull

#####
```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
#start timing code – for possible analysis purposes
#####

start_time=time.time()

#####

#Functions to find symmetry mappings of molecule
#####

#READ MOLECULE FUNCTION

def read_molecule(underlying_mol):
    pymat_read=io.xyz.XYZ.from_file(underlying_mol)
    pymat_mol=pymat_read.molecule
    return pymat_mol

#WRITE TO FILE FUNCTION

def write_mol(mol_xyz,mol_name):
    mol_xyz.write_file(mol_name)

#GET CENTERED MOLECULE FUNCTION

def get_centered(underlying_mol,to_centre):
    centered=to_centre.get_centered_molecule()
    centered_xyz=io.xyz.XYZ(centered)
    #WRITING STEP – ideally could be removed for more optimal code
    mol_name='centered_' + underlying_mol.split('.')[0] + '.xyz'
    write_mol(centered_xyz,mol_name)
    return (centered,mol_name)

#GET OPERATORS FUNCTION

def get_operators(centered_mol):
    analyser=symmetry.analyzer.PointGroupAnalyzer(centered_mol)
    operators=analyser.get_symmetry_operations()
    return operators

#APPLY TRANSFORMATIONS FUNCTIONS – in more optimal code could be done
#smoother with less read/write

#getting vectors representing atom positions and species
def get_atom_vectors(centered_file):
    with open(centered_file,'r') as f:
        lines=f.readlines()
```



```
    atom_vectors=[]
    for i in range(2,len(lines)):
        atom_coordinate_list = lines[i].split()[1:]
        atom_vector= [float(x) for x in atom_coordinate_list]
        atom_el = lines[i].split()[0]
        atom_vector.insert(0,atom_el)
        atom_vectors.append(atom_vector)
        mol_atoms=len(atom_vectors)
    return (atom_vectors ,mol_atoms)

#doing transformation on an atom vector
def transform_atom(atom_vector , operator):
    #get co-ords
    coords = atom_vector[1:]
    coords = np.array(coords)
    #make new_vector
    new_vec = [atom_vector[0]]
    #rotating
    rot_matrix = operator.rotation_matrix
    rotated_atom = np.matmul(rot_matrix , coords)

    #translating
    tau = operator.translation_vector
    moved_atom = rotated_atom + tau

    #new vector updating
    for i in moved_atom:
        new_vec.append(i)
    return new_vec

#transforming all atoms in molecule
def transform_molecule(atom_vectors , operator):
    new_vecs=[]
    for vec in atom_vectors:
        new_vec=transform_atom(vec , operator)
        new_vecs.append(new_vec)
    return new_vecs

#writing transformed molecule file
def write_trans(new_vecs , new_filename):
    with open(new_filename , 'a') as f:
        f.writelines( str(len(new_vecs)) + '\n')
        atom_lines=[]
        for new_vec in new_vecs:
            str_changed_atom = [str(element) for element in new_vec]
            atom_line = join(str_changed_atom) + "\n"
            atom_line.append(atom_lines)
        f.writelines(atom_lines)
```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
#FIND MATCHUPS FUNCTIONS - to identify atom-atom mappings between original and \
→transformed

#set up method for writing the match lists
def write_match(match_list, map_string, i, j):
    if i==j:
        match = [i]
        match_list.append(match)
        map_string = map_string + str(i) + '_'
    else:
        match = [i, j]
        match_list.append(match)
        map_string = map_string + str(i) + str(j) + '_'
    return (match_list, map_string)

#set up method to gather operator matches and check for duplication
def gather_matches(all_maps, all_map_strings, match_list, map_string):
    if map_string not in all_map_strings:
        all_map_strings.append(map_string)
        all_maps.append(match_list)
    return (all_maps, all_map_strings)

#find the matches - based on shared coordinates and atom species
def get_matches(atom_vectors, new_vecs, tolerance):
    match_list=[]
    map_string='string -'
    for i in range(len(atom_vectors)):
        for j in range(len(new_vecs)):
            if (atom_vectors[i][0]==(new_vecs[j])[0]):
                old=np.array((atom_vectors[i])[1:])
                new=np.array((new_vecs[j])[1:])
                diff=abs(old-new)
                if all(x <= tolerance for x in diff):
                    update=(write_match(match_list, map_string, i, j))
                    match_list=update[0]
                    map_string=update[1]
    return (match_list, map_string)

#OVERALL FUNCTION TO RUN THE MAPPINGS SECTION

def overall_map(underlying_mol, tolerance):
    #initialise all_map_strings and all_maps
    all_maps=[]
    all_map_strings=[]
    #read in molecule
    pymat_mol=read_molecule(underlying_mol)
    #Centre molecule
    centered_mol=get_centered(underlying_mol, pymat_mol)[0]
```

```
centered_file=get_centered(underlying_mol,pymat_mol)[1]
#Get operators
operators=get_operators(centered_mol)
#get atom vectors and mol atoms for original
atom_vectors_run=get_atom_vectors(centered_file)
atom_vectors=atom_vectors_run[0]
molecule_atoms=atom_vectors_run[1]
#Do transformations and mapping for each operator
for k in range(len(operators)):
    operator=operators[k]
    new_vecs=transform_molecule(atom_vectors,operator)
    match_step=get_matches(atom_vectors,new_vecs,tolerance)
    match_list=match_step[0]
    map_string=match_step[1]
    #gather up all the mappings
    gather=gather_matches(all_maps,all_map_strings,match_list,map_string)
    all_maps=gather[0]
    all_map_strings=gather[1]
#save the mapping data
all_maps_array=np.array(all_maps,dtype=object)
np.save('all_maps.npy',all_maps_array)
return (all_maps,molecule_atoms)

#####
#SPACEGROUP/SETTINGS FINDER SECTION – needed for setting up atoms objects
#####
#FUNCTION TO GET SPACEGROUPS
def get_spacegroups(filename):
    crystal_reader=ccdc.io.CrystalReader(filename)
    crystal = crystal_reader[0]
    spacegroup = crystal.spacegroup_number_and_setting
    if spacegroup[1]>2:
        print('ASE cannot handle this setting – conversion needed')
        sys.exit()
    return spacegroup

#####
#MAKING INPUTS FROM RES FILES ETC SECTION – this is only releavnt to cspy output files
#also obtains energy and asymmetric unit length for later use
#####
def file_neatener(filename):
    with open(filename,"r") as f:
        lines = f.readlines()
#Read in total energy from top line, convert to au per atom and add to list
    energy_line = lines[0].split()
```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
total_energy = energy_line[2]
atom_energy = (float(total_energy))
#one less/start part identifies where the co-ordinate part begins-i.e which lines to \
→edit
search = 'SFAC'
one_less = [lines.index(line) for line in lines if search in line]
start = one_less[0] + 1
asymm_length = len(range(start,(len(lines))))
#looping over coordinate section lines to be edited
# edit the lines to add zeros columns and remove the numbers from the element \
→column

for i in range(start,(len(lines))):
    col_list = (lines[i]).split()
    #take only the letters from the first column
    element=[char for char in col_list[0] if char.isalpha()]
    col_list[0]="".join(str(x) for x in element)
    #add two columns of zeros
    col_list.append(0)
    col_list.append(0)
    str_col_list = [str(col) for col in col_list]
    lines[i]= " ".join(str_col_list) + "\n"

return (lines ,atom_energy ,asymm_length)

#FUNCTION TO MAKE ATOMS OBJECT
def make_atoms(lines ,spacegroup):
    with open('correct_res.res','w') as f:
        f.writelines(lines)
    atom_struc=aseio.read('correct_res.res')
    spacegrp=spacegroup[0]
    print('spacegroup detected')
    print(spacegrp)
    setting=spacegroup[1]
    # add the spacegroup info and use crystal structure to apply bulk/crystal info to the \
→relevant atoms object
    atom_struc = crystal(symbols=atom_struc ,spacegroup=spacegrp ,setting=setting ,pbc=\
→True)
    os.remove('correct_res.res')
    return atom_struc

#FUNCTION TO DO OVERALL MAKING INPUTS – start with zipfile of res structures
def make_inputs(structures_zip_name):
    atoms_list=[]
```

```

energy_list=[]
asymm_list=[]
with ZipFile(structures_zip_name,'r') as structures_zip:
    #get list of filenames
    structure_list = structures_zip.namelist()
    for i in range(len(structure_list)):
        filename = structure_list[i]
        structures_zip.extract(filename)
        spacegroup = get_spacegroups(filename)
        neatener=file_neatener(filename)
        os.remove(filename)
        lines=neatener[0]
        energy=neatener[1]
        asymm_length=neatener[2]
        struc_atoms = make_atoms(lines,spacegroup)
        #Append each entry to corresponding list
        atoms_list.append(struc_atoms)
        energy_list.append(energy)
        asymm_list.append(asymm_length)
    return (atoms_list,energy_list,asymm_list)

#####
# RE-ORDERING FUNCTION – needed later to reorder energies properties based on
#Z', as structures are reordered that way
#####

def reorderer(property_list,asymm_list,Z_prime_list,molecule_atoms):
    prop_lists=dict([(k, []) for k in Z_prime_list])
    for i in range(len(asymm_list)):
        if asymm_list[i]/molecule_atoms in Z_prime_list:
            prop_lists[(int(asymm_list[i]/molecule_atoms))].append(property_list[i])
        else:
            print('reordering error-structure is of unfamiliar z_prime')
            sys.exit()
    ordered_prop_lists=collections.OrderedDict(sorted(prop_lists.items(), key=lambda t:
→ t[0]))
    reordered_list=np.concatenate(list(ordered_prop_lists.values()))
    return reordered_list

#####\
→
#KERNEL CALCULATION SECTION – has functions needed for SOAP descriptor and kernel \
→calculation
#####\
→

```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
#FUNCTION TO 'SETUP KERNEL' WITH IMPORTANT PARAMETERS OF CHOICE
def initialise_kernel(cut_off):
    HYPERS = {'soap_type': 'PowerSpectrum', 'interaction_cutoff': cut_off, 'max_radial': \
→8, 'max_angular': 6, 'gaussian_sigma_constant': 0.3, 'gaussian_sigma_type': 'Constant', '\
→cutoff_smooth_width': 0.5, 'radial_basis': 'GTO', 'inversion_symmetry': True, 'normalize'\
→: True}

    features = SOAP(**HYPERS)
    kernel = rascal.models.Kernel(features, kernel_type='Full', target_type='Structure \
→', zeta=1)
    return (kernel, features)

#FUNCTION TO SORT STRUCTURES BY Z_PRIME -
def sort_z_prime(atoms_list, asymm_lengths, mol_name, molecule_atoms):
    z_prime_list=[]
    mini_lists=[]
    for i in range(0, len(atoms_list)):
        #establish z-prime for each structure
        atoms = atoms_list[i]
        asymm = asymm_lengths[i]
        z_prime = int(asymm/molecule_atoms)
        if z_prime != asymm/molecule_atoms:
            print('structure is of non-integer Z prime. This dataset cannot be handled')
            sys.exit()
        #add structure to relevant mini list amd record z prime value
        if z_prime not in z_prime_list:
            z_prime_list.append(z_prime)
            mini_lists.append((z_prime, []))
        for mini in mini_lists:
            if mini[0]==z_prime:
                mini[1].append(atoms)
    for mini in mini_lists:
        filename = 'z_prime' + str(mini[0]) + '_' + mol_name + '.xyz'
        aseio.write(filename, mini[1])
    return (z_prime_list, mini_lists)

#FUNCTION TO CALC NUMBER OF MOLECULES IN A STRUCTURE (Z) - needed for identifying atoms \
→ later
def get_num_mol(single_struc, molecule_atoms):
    all_atoms = len(single_struc)
    num_mol = int(all_atoms/molecule_atoms)
    return num_mol

#FUNCTION TO TAKE THE ATOM INDEX AND RETURN THE ATOM IDS THAT NEED TO BE MASKED IN THE \
→ORIGINAL STRUCTURE
def to_mask(atom_index, zp, num_mol, molecule_atoms):
```

```
masked = []
number_asymm = int(num_mol/zp)
#Get overall index of molecular atom index in each molecule in assymmetric unit, \
→accounting for possible multiple copies of asymm unit in file
for i in range(zp):
    to_mask_start =int((atom_index + molecule_atoms*i) * number_asymm)
    to_mask = [i for i in range(to_mask_start , (to_mask_start + 1))]
    masked = masked + to_mask
    print('masking in A')
    print(masked)
return masked

#FUNCTION TO TAKE THE ATOM INDICES (FROM MAPPING LISTS) AND RETURN THE ATOM IDS THAT \
→NEED TO BE MASKED IN THE TRANSFORMED STRUCTURE

def to_mask_adv( atom_index_list , zp , num_mol , molecule_atoms ):
    masked = []
    number_asymm = int(num_mol/zp)
    #sanity check- atom_index_list is of correct length
    try:
        atom_index_list[zp-1]
    except IndexError:
        print('Atom_index_list is too short. It should have Z prime elements')

    try:
        check_list = ['check']*zp
        check_list[(len(atom_index_list)-1)]
    except IndexError:
        print('Atom_index_list is too long. It should have Z prime elements')
    #work out overall file inidices for the desired molecular atom indices in each
    #same process as before but masking off a different molecular atom index (according\
→ to desired mappings) in each molecule of asymmetric unit
    for i in range(zp):
        to_mask_start =int((atom_index_list[i] + molecule_atoms*i) * number_asymm)
        to_mask = [i for i in range(to_mask_start , (to_mask_start + 1))]
        masked = masked + to_mask
        print('masking in B')
        print(masked)
    return masked

#FUNCTION TO GET Z' COMBINATIONS OF POSSIBLE MAPPINGS – Temporarily defunct as causes \
→issues

def get_combos( atom_index , zp_max , symm_opp_list ):
    #Generate the list of indices that could be equivalent to a given atom_index , one \
→mapping at a time
```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
possible_equivalents = []
#To generate possible equivalents take atom index - search for it in each mapping ↘
→ pull out what it maps to and add to equivalents list
for symm_opp in symm_opp_list:
    for i in symm_opp:
        if i[0] == atom_index:
            try:
                possible_equivalents.append(i[1])
            except IndexError:
                possible_equivalents.append(i[0])

#Generate the possible z_prime combinations of possible equivalent indices
if zp_max > 1:
    symm_combos=[combo for combo in itertools.product(possible_equivalents, repeat=↘
→zp_max)]
if zp_max == 1:
    symm_combos = [[combo] for combo in possible_equivalents]

return symm_combos

#SET UP PARTIAL FUNCTION TO DO KERNEL CALCULATION FOR EACH MAPPING FOR SINGLE ORIGINAL ↘
→ATOM INDEX (or Z' combination of mappings)

def one_symm(symm_combo, struct_A, struct_B, file_A, file_B, molecule_atoms, zp_max, zp_min, ↘
→atom_index, features, kernel):
    combo = symm_combo
    combo = [atom for atom in combo]

    #Mask off atoms corresponding to the relevant atom index in original structure ↘
→ relies on to_mask function
    for s in struct_A:
        s.wrap(eps=1e-18)
        all_atoms = len(s)
        number_molecules = int(get_num_mol(s, molecule_atoms))
        to_masks = to_mask(atom_index, zp_min, number_molecules, molecule_atoms)
        mask_center_atoms_by_id(s, to_masks)

    for s in struct_B:
        #mask off atoms corresponding to the symmetry combination in transformed ↘
→structure
        s.wrap(eps=1e-18)
        all_atoms = len(s)
        number_molecules = int(get_num_mol(s, molecule_atoms))
        to_masks = to_mask_adv(combo, zp_max, number_molecules, molecule_atoms)
        mask_center_atoms_by_id(s, to_masks)

    #Take kernel for atom contribution and relevant symmetry combination - add to ↘
→corresponding total kernel
    struct_A = features.transform(struct_A)
    struct_B = features.transform(struct_B)
```



```

        print('show descriptors ')
        print(struct_A.get_features(features))
        print(struct_B.get_features(features))
        results=kernel.__call__(struct_A,struct_B)
#reset the structure lists
        struct_A=[]
        struct_B=[]
        struct_A = aseio.read(file_A,index=':')
        struct_B = aseio.read(file_B,index=':')
        print('done symm opp ')

    return results

#FUNCTION TO GET 'FULL' ADAPTED KENREL FOR EACH Z' PAIR
def zprime_pair_kernel(zp_max,zp_min,file_A,file_B,symm_opp_list,molecule_atoms,\
→features,kernel,cores_requested):

    #Read in 'mini structure lists' for each z_prime
    struct_A = aseio.read(file_A,index=':')
    struct_B = aseio.read(file_B,index=':')
    number_combos = len(symm_opp_list)**zp_max
    total_kernels=[np.zeros([len(struct_A),len(struct_B)])]*number_combos
    #Loop over contributions from each atom in the molecule
    for atom_index in range(0,molecule_atoms):
        #Generate the possible symmetry combos – in terms of molecular atom indexes
        symm_combos=get_combos(atom_index,zp_max,symm_opp_list)
        cores_required=len(symm_combos)
        core_options=[cores_required,cores_requested]
        cores_needed=min(core_options)
        print('got combos ')
        #perform the single atom contribution kernel calculations
        with Pool(int(cores_needed)) as pool:
            symm_combo = partial(one_symm,struct_A=struct_A,struct_B=struct_B,file_A=
→file_A,file_B=file_B,zp_max=zp_max,zp_min=zp_min,molecule_atoms=molecule_atoms,\
→atom_index=atom_index,features=features,kernel=kernel)

            atom_kernels = pool.map(symm_combo,symm_combos)

        for i in range(0,number_combos):
            total_kernels[i] = total_kernels[i] + atom_kernels[i]
        print('done ', atom_index)
    #average over the atom index contributions
    average_kernels=[np.zeros([len(struct_A),len(struct_B)])]*number_combos
    for i in range(0, len(symm_combos)):
        average_kernels[i] = total_kernels[i]/molecule_atoms
    average_kernels_array = np.array(average_kernels)

    #Take the mean value for each structure pair over all the possible symmetry

```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
→combinations
    final_kernel = get_mean_kernel(average_kernels , len(struct_A) , len(struct_B))
    return final_kernel

#Function to get average of possibilities kernel (over different mappings)

def get_mean_kernel(kernel_possibilities , length_A , length_B):
    correct_kernel=np.zeros([length_A , length_B])
    #For each structure pair , loop over the possibilities and take the mean
    for A in range(0,length_A):
        for B in range(0,length_B):
            possibilities = [chance[A,B] for chance in kernel_possibilities]
            value = np.mean(possibilities)
            correct_kernel[A,B] = value
    return correct_kernel

#function to run the kernel calculator to make each z' pair kernel

def make_all_kernel_bits(Z_prime_list , mol_name , symm_opp_list , molecule_atoms , features , \
→kernel , cores_requested):
    for zp1 in Z_prime_list:
        for zp2 in Z_prime_list:
            if zp1 >= zp2:
                file_B = 'z_prime' + str(zp1) + '_' + mol_name + '.xyz'
                file_A = 'z_prime' + str(zp2) + '_' + mol_name + '.xyz'
                zprime_kernel = zprime_pair_kernel(zp1 , zp2 , file_A , file_B , symm_opp_list , \
→molecule_atoms , features , kernel , cores_requested)
                np.save('z_prime_' + str(zp1) + '_' + str(zp2) + '_' + mol_name + '.npz' , \
→zprime_kernel)

#####
#KERNEL RECOMBINATION SECTION
#####

def kernel_combiner(z_prime_list , mol_name):
    #piece together kernel sections in Z' order
    made_full=False
    for i in z_prime_list:
        made_column=False
        for j in z_prime_list:
            if i <= j:
                filename = 'z_prime_' + str(j) + '_' + str(i) + '_' + mol_name + '.npz'
                part = np.transpose(np.load(filename))
            else:
                filename = 'z_prime_' + str(i) + '_' + str(j) + '_' + mol_name + '.npz'
                part = np.load(filename)
            if made_column:
                column = np.concatenate((column , part) , axis=0)
            else:
```

```

        column=part
        made_column=True
    if made_full:
        full=np.concatenate((full,column), axis=1)
    else:
        full=column
        made_full=True
    normalised = full.copy()
    #normalise kernels
    for i in range(0,full.shape[0]):
        for j in range(0,full.shape[0]):
            normalised[i,j] = full[i,j]/((full[i,i]*full[j,j])**0.5)

    kernel_name='final_average_possibilities_kernel_' + mol_name + '.npy'
    np.save(kernel_name ,normalised)
    return normalised

#####
#GETTING DRESSED ENERGIES FROM KERNEL SECTION
#####

#may want to adapt this when making full thing to avoid read-write and take kernels etc ↘
→ as straight variables

#####
#PROJECTION MAKING FUNCTIONS
#####
#kpca function from original GCH repository (ref [164]) - including possible centering ↘
→step error
def kpca(kernel,ndim):
    """ Extracts the first ndim principal components in the space
    induced by the reference kernel (Will expect a square matrix) """
    #Centering step
    k = kernel.copy()
    cols=np.mean(k,axis=0);
    rows=np.mean(k,axis=1);
    mean=np.mean(cols);
    for i in range(len(k)):
        k[:,i]-=cols
        k[i,:]-=rows
    k += mean
    # Eigensystem step
    eval, evec = salg.eigh(k,eigvals=(len(k)-ndim,len(k)-1))
    eval=np.flipud(eval); evec=np.fliplr(evec)
    print(eval)
    pvec = evec.copy()
    print(pvec)
    for i in range(ndim):

```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
pvec[:,i] *= 1./np.sqrt(eval[i])

# Projection step
return np.dot(k, pvec)

#function to take kernel, make projection, and join it with the energies of choice
def make_kpca_data(kernel,proj_size,energies):
    kern=kernel
    projection=kpca(kern,proj_size)
    projection =pd.DataFrame(projection)
    energies =pd.DataFrame(energies)
    data=pd.concat((energies,projection),axis=1,ignore_index=True)
    data=data.to_numpy()
    return data

#function to do overall kpca_generation - ideally don't need this as a seperate \
→function
def do_kpca_process(kernel,proj_size,energy_array):
    data = make_kpca_data(kernel,proj_size,energy_array)

    return data

#####
# HULL TAKING/DRESSED ENERGY DATA ETC FUNCTIONS
#####

#function to cut down the data to desired dimensions and take required hull
def take_hull(data,dimensions):
    points = data[:,0:dimensions]
    hull = chull(points)
    return (hull,points)

#function to get chemically relevant hull - based on facet equations
def get_relevant(hull):
    slist = hull.simplices
    snormals=hull.equations
    bad_equations=[]
    vlist=[]
    for i in range(len(slist)):
        if snormals[i,0] > 0.:
            bad_equations.append(i)
        else:
            vlist = np.union1d(vlist, slist[i])
    equations = np.delete(snormals, bad_equations, axis=0)
    return (equations, vlist)
```

```
#####
#functions to get dressed energies – adapted from DCH code (ref [110,111])
#####
def get_all_distances(points, equations):
    """
    Computes the distance of the points to the planes defined by the equations
    with respect to the direction of the first dimension.
    equations : ndarray of shape (n_facets, n_dim)
                each row contains the coefficientst for the plane equation of the form
                equations[i, 0]*x_1 + ...
                + equations[i, -2]*x_{n_dim} = equations[i, -1]
                -equations[i, -1] is the offset
    points : ndarray of shape (n_samples, n_dim)
            points to compute the directional distance from

    Returns
    -----
    directional_distance : ndarray of shape (nsamples, nequations)
                        closest distance wrt. the first dimension of the point to the planes
                        defined by the equations
    """
    orthogonal_distances = -(points @ equations[:, :-1].T) - equations[:, -1:].T
    return -orthogonal_distances / equations[:, :1].T

#function to get dressed energies (closest distances to hull for all points)

def get_dressed(points, equations):
    distances = get_all_distances(points, equations)
    # we get negative distances for each plane to check if any distance is below the ↘
→threshold
    below_directional_convex_hull = np.any(distances < -0.000001, axis=1)
    # directional distances to corresponding plane equation
    dressed_energies = np.zeros(len(points))
    dressed_energies[~below_directional_convex_hull] = np.min(distances[~↘
→below_directional_convex_hull], axis=1)
    # some distances can be negative if tolerances allow it to be outside of hull, so ↘
→we take the max of all negative distances for the corresponding
    # point to be the dressed energy in that case
    negative_directional_distances = distances.copy()
    negative_directional_distances[distances > 0] = -np.inf
    dressed_energies[below_directional_convex_hull] = np.max(↘
→negative_directional_distances[below_directional_convex_hull], axis=1)
    return dressed_energies
```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

```
#function to get dressed energies if starting from a projection
def get_dressed_energies_analysis(projection, dimensions):
    hull=take_hull(projection, dimensions)
    #get relevant part for hull
    relevant=get_relevant(hull[0])
    equations=relevant[0]
    points=hull[1]
    dressed_energies=get_dressed(points, equations)
    return dressed_energies

#function to get dressed energies if starting from kernel
def get_dressed_energies_full(kernel, proj_size, energy_array, dimensions, mol_name):
    data=do_kpca_process(kernel, proj_size, energy_array)
    projection_name='kPCA_projection_for_average_possibilities_'+ mol_name + '.\n
→numpy'
    np.save(projection_name, data)
    dressed_energies=get_dressed_energies_analysis(data, dimensions)
    return dressed_energies

#####
#FUNCTION TO RUN COMPLETE PROCESS
#####
def from_start_to_end(molecule, mol_name, tolerance, struc_zip, proj_size, cut_off, \n
→dimensions, job_type, projection_file, cores_requested):
    if job_type=='full':
        mappings=overall_map(molecule, tolerance)
        symm_opp_list=mappings[0]
        molecule_atoms=mappings[1]

        inputs=make_inputs(struc_zip)
        atoms_list=inputs[0]
        energy_list=inputs[1]
        asymm_list=inputs[2]

        initial=initialise_kernel(cut_off)
        kernel=initial[0]
        features=initial[1]
        z_prime_data = sort_z_prime(atoms_list, asymm_list, mol_name, molecule_atoms)
        Z_prime_list=z_prime_data[0]
        kernel_bits=make_all_kernel_bits(Z_prime_list, mol_name, symm_opp_list, \n
→molecule_atoms, features, kernel, cores_requested)

        final_kernel=kernel_combiner(Z_prime_list, mol_name)
    #reorder the energies list
    correct_energy_list=reorderer(energy_list, asymm_list, Z_prime_list, molecule_atoms \n
```

```

→)
    energy_name='correct_order_energies_' + mol_name + '.numpy'
    np.save(energy_name, correct_energy_list)
    values=get_dressed_energies_full(final_kernel, proj_size, correct_energy_list, \
→dimensions, mol_name)
    dressed_name='dressed_energies_' + mol_name + '_' + str(cut_off) + '.numpy'
    np.save(dressed_name, values)
    if job_type=='analysis':
        projection=np.load(projection_file)
        values=get_dressed_energies_analysis(projection, dimensions)
        dressed_name='dressed_energies_' + mol_name + '_' + str(cut_off) + '.numpy'
        np.save(dressed_name, values)
    return(values)

#####
parser = argparse.ArgumentParser(
    prog='GCH Dressed Energy Generator',
    description='Takes inputs of the molecular geometry xyz and a zip-
→file of (CSPY FORMAT!) res files of predicted structures and returns an array of hull
→energies for each structure (ordered by original order split into Z prime sections)',
    epilog='GCH for the win!')

parser.add_argument('-sz', '--struc_zip', default='none', help='Zip file of predicted
→structure res files. For now MUST be Cspy format res')
parser.add_argument('-mf', '--mol_file', default='none', help='xyz file of underlying
→molecular geometry. For now can only be one (Rigid CSP)')
parser.add_argument('-mn', '--mol_name', default='none', help='name you want to give to
→the system')
parser.add_argument('-tol', '--tolerance', type=float, default=0.3, help='Tolerance value
→for calculation of atom index mappings (default=0.3)')
parser.add_argument('-ps', '--proj_size', type=int, default=32, help='Number of kPCA
→components to be calculated (default=32, unlikely to need bigger)')
parser.add_argument('-co', '--cut_off', type=float, default=4, help='SOAP cut-off for
→descriptor calculation in Angstroms (default=4)')
parser.add_argument('-d', '--dimensions', type=int, default=2, help='Number of dimensions
→including energy dimension- desired for hull construction (default=2)')
parser.add_argument('-jt', '--job_type', type=str, default='full', help='Type of job to run
→ - either "full" (full process) or "analysis" - (just dressed energy calculations -
→you must provide projection)')
parser.add_argument('-pj', '--projection_file', type=str, default='none', help='File
→containing the ready-made projection - only for use if job-type is analysis')
parser.add_argument('-j', '--core_req', type=int, default=1, help='Number of requested
→cores to parallelise over - in actual usage terms - only number of cores equal to
→eventual number of symmetry operator combinations will be used - so limit this to
→something sensible e.g 20 to avoid too much wastage')

```

## B.2. SCRIPT TO CALCULATE ADAPTED KERNEL FOR SYSTEMS WITH RIGID UNDERLYING MOLECULES

---

```
args = parser.parse_args()

#####
#RUN PROCESS
#####

answer = from_start_to_end(args.mol_file, args.mol_name, args.tolerance, args.struc_zip, \
→args.proj_size, args.cut_off, args.dimensions, args.job_type, args.projection_file, args.\
→core_req)
print(answer)

end_time=time.time()
run_time=end_time-start_time
print('actual python took:', run_time, 'seconds')

#####

#THE END
```



### B.3 Script to Calculate the Direct Product of Symmetry Mappings

```
#####
#Imports
#####

from pymatgen import io
from pymatgen.io import xyz
from pymatgen import symmetry
from pymatgen.symmetry import analyzer
from zipfile import ZipFile
import numpy as np
import os
from itertools import chain, combinations

#####
#Functions to get mappings for a single molecule
#####
def read_molecule(underlying_mol):
    pymat_read=io.xyz.XYZ.from_file(underlying_mol)
    pymat_mol=pymat_read.molecule
    return pymat_mol

def write_mol(mol_xyz, mol_name):
    mol_xyz.write_file(mol_name)

def get_centered(underlying_mol, to_centre):
    centered=to_centre.get_centered_molecule()
    centered_xyz=io.xyz.XYZ(centered)
    mol_name='centered_' + underlying_mol.split('.')[0] + '.xyz'
    write_mol(centered_xyz, mol_name)
    return (centered, mol_name)

def get_group(centered_mol):
    analyser=symmetry.analyzer.PointGroupAnalyzer(centered_mol)
    point_group=analyser.get_pointgroup()
    return point_group

def get_symm_one(centered_mol):
    analyser=symmetry.analyzer.PointGroupAnalyzer(centered_mol)
    symm_version=analyser.symmetrize_molecule()
    return symm_version
```

```

def overall_group(underlying_mol):
    all_maps=[]
    all_map_strings=[]
    pymat_mol=read_molecule(underlying_mol)
    group = get_group(pymat_mol)
    case_name = (underlying_mol , group.sch_symbol)
    return case_name

def get_operators(centered_mol):
    analyser=symmetry.analyzer.PointGroupAnalyzer(centered_mol)
    operators=analyser.get_symmetry_operations()
    return operators

def transform_atom(atom_vector , operator):
    #get co-ords
    coords = atom_vector[1:]
    coords = np.array(coords)
    #make new_vector
    new_vec = [atom_vector[0]]
    #rotating
    rot_matrix = operator.rotation_matrix
    rotated_atom = np.matmul(rot_matrix , coords)

    #translating
    tau = operator.translation_vector
    moved_atom = rotated_atom + tau

    #new vector make
    for i in moved_atom:
        new_vec.append(i)
    return new_vec

def get_atom_vectors(centered_file):
    with open(centered_file , 'r') as f:
        lines=f.readlines()
        atom_vectors=[]
        for i in range(2,len(lines)):#hardoced 2 here is to make sure starts at \
→corrctet line of file
            atom_coordinate_list = lines[i].split()[1:]
            atom_vector= [float(x) for x in atom_coordinate_list]
            atom_el = lines[i].split()[0]
            atom_vector.insert(0,atom_el)
            atom_vectors.append(atom_vector)
            mol_atoms=len(atom_vectors)
    return (atom_vectors , mol_atoms)

```

```
def transform_molecule(atom_vectors, operator):
    new_vecs=[]
    for vec in atom_vectors:
        new_vec=transform_atom(vec, operator)
        new_vecs.append(new_vec)
    return new_vecs

#writing transformed
def write_trans(new_vecs, new_filename):
    with open(new_filename, 'a') as f:
        f.writelines(str(len(new_vecs)) + '\n')
        atom_lines=[]
        for new_vec in new_vecs:
            str_changed_atom = [str(element) for element in new_vec]
            atom_line = join(str_changed_atom) + "\n"
            atom_line.append(atom_lines)
        f.writelines(atom_lines)

def write_match(match_list, map_string, i, j):
    if i==j:
        match = (i,)
        match_list.append(match)
        map_string = map_string + str(i) + '_'
    else:
        match = (i, j)
        match_list.append(match)
        map_string = map_string + str(i) + str(j) + '_'
    return (match_list, map_string)

def gather_matches(all_maps, all_map_strings, match_list, map_string):
    if map_string not in all_map_strings:
        all_map_strings.append(map_string)
        all_maps.append(match_list)
    return (all_maps, all_map_strings)

def get_matches(atom_vectors, new_vecs, tolerance):
    match_list=[]
    map_string='string -'
    for i in range(len(atom_vectors)):
        for j in range(len(new_vecs)):
            if (atom_vectors[i][0]==(new_vecs[j][0]):
                old=np.array((atom_vectors[i][1:]))
                new=np.array((new_vecs[j][1:]))
                diff=abs(old-new)
```

```

        if all(x <= tolerance for x in diff):
            update=(write_match(match_list, map_string, i, j))
            match_list=update[0]
            map_string=update[1]
    return (match_list, map_string)

def overall_map(underlying_mol, tolerance):
    #initialise all_map_strings and all_maps
    all_maps=[]
    all_map_strings=[]
    #read in molecule
    pymat_mol=read_molecule(underlying_mol)
    #Centre molecule
    centered_mol=get_centered(underlying_mol, pymat_mol)[0]
    centered_file=get_centered(underlying_mol, pymat_mol)[1]
    #Get operators
    operators=get_operators(centered_mol)
    #get atom vectors and mol atoms for original
    atom_vectors_run=get_atom_vectors(centered_file)
    atom_vectors=atom_vectors_run[0]
    molecule_atoms=atom_vectors_run[1]
    for k in range(len(operators)):
        operator=operators[k]
        new_vecs=transform_molecule(atom_vectors, operator)
        match_step=get_matches(atom_vectors, new_vecs, tolerance)
        match_list=match_step[0]
        map_string=match_step[1]
        #gather up all the mappings
        gather=gather_matches(all_maps, all_map_strings, match_list, map_string)
        all_maps=gather[0]
        all_map_strings=gather[1]
    all_maps_array=np.array(all_maps, dtype=object)
    return (all_maps, molecule_atoms)

#####
#Eliminating subgroups functions – needed to simplify direct product task
#####

#Powerset function – to get all subgroups of the mapping groups
def powerset(set_name):
    s = list(set_name)
    return set(chain.from_iterable(combinations(s, r) for r in range(len(s)+1)))

```

```

#check if group of mappings is subgroup of a group already covered
def check_sub(one_map, covered, max_list):
    removals=[]
    #Make sets of mappings into frozen sets so that can have set of (frozen)sets
    ops=frozenset(tuple(i) for i in one_map)
    #if group of mappings not a subgroup of existing – add it – and all its subgroups
    if ops not in covered:
        subs=powerset(ops)
        subs.remove(())
        for sub in subs:
            covered.update([frozenset(sub)])
    #remove any previously 'needed' master groups that are now a subgroup of the new
→ one
    for big in max_list:
        if big in [frozenset(sub) for sub in subs]:
            removals.append(big)
    for bad in removals:
        max_list.remove(bad)
    #add new group of mappings to the list of 'master' groups needed for direct
→product calculation
    max_list.append(ops)
    return covered, max_list

#dictionary of point group subgroup – not needed if using groups of mappings method
pg_dict={"C1":["C1"], "Ci":["Ci", "C1"], "C2":["C2", "C1"], "Cs":["Cs", "C1"], "C2h":["C2h", "C2", "Cs", "C1", "Ci"], "D2":["D2", "C2", "C1"], "C2v":["C2v", "C2", "Cs", "C1"], "D2h":["D2h", "C2v", "D2", "C2h", "C2", "Cs", "C1"], "C4":["C4", "C2", "C1"], "S4":["S4", "C2", "C1"], "C4h":["C4h", "C4", "S4", "C2h", "C2", "Cs", "C1"], "D4":["D4", "C4", "D2", "C2", "C1"], "C4v":["C4v", "C4", "C2v", "C2", "Cs", "C1"], "D2d":["D2d", "S4", "C2v", "D2", "C2", "Cs", "C1"], "D4h":["D4h", "D2d", "C4v", "D4", "C4h", "C4", "S4", "D2h", "Cev", "D2", "C2h", "Cs", "C2", "Ci", "C1"], "C3":["C3", "C1"], "C3i":["C3i", "C3", "Ci", "C1"], "D3":["D3", "C3", "C2", "C1"], "C3v":["C3v", "C3", "Cs", "C1"], "D3d":["D3d", "C3v", "D3", "C3i", "C3", "C2h", "Cs", "C2", "Ci", "C1"], "C6":["C6", "C3", "C2", "C1"], "C3h":["C3h", "C3", "Cs", "C1"], "C6h":["C6h", "C3h", "C6", "C3i", "C3", "C2h", "Cs", "C2", "Ci", "C1"], "D6":["D6", "C6", "D3", "C3", "D2", "C2", "C1"], "C6v":["C6v", "C6", "C3v", "C3", "C2v", "C2", "Cs", "C1"], "D3h":["D3h", "C3h", "C3v", "D3", "C3", "C2v", "C2", "Cs", "C1"], "D6h":["D6h", "D3h", "C6v", "D6", "C6h", "D3d", "C3h", "C6", "C3v", "D3", "C3i", "C3", "D2h", "C2v", "D2", "C2h", "C2", "Cs", "Ci", "C1"], "T":["T", "C3", "D2", "C2", "C1"], "Th":["Th", "T", "C3i", "C3", "D2h", "C2v", "D2", "C2h", "C2", "Cs", "Ci", "C1"], "O":["O", "T", "D3", "D4", "C4", "C3", "D2", "C2", "C1"], "Td":["Td", "T", "C3v", "C3", "D2d", "S4", "C2v", "D2", "C2", "Cs", "C1"], "Oh":["Oh", "Td", "O", "Th", "T", "D3d", "C3v", "D3", "C3i", "C3", "D4h", "D2d", "C4v", "D4", "C4h", "S4", "C4", "D2h", "C2v", "D2", "C2h", "C2", "Cs", "Ci", "C1"]}

#####
#Take direct product functions
#####

```

```

#function to combine any two MAP (not group of mappings)
def combine_maps(map_1,map_2,molecule_atoms):
    combo_map=[]
    for atom_index in range(molecule_atoms):
        for i in map_1:
            if i[0] == atom_index:
                try:
                    intermed=(i[1])
                except IndexError:
                    intermed=(i[0])
        for j in map_2:
            if j[0] == intermed:
                try:
                    combined=(j[1])
                except IndexError:
                    combined=(j[0])
        if atom_index!=combined:
            match = [atom_index ,combined]
        else:
            match = [atom_index]
        combo_map.append(match)
    return combo_map

#get required groups of mappings to form product

maps_sets=[]
covered=set(())
max_list=[]

with ZipFile(<zip file of conformation xyz files>) as conf_zip:
    names = conf_zip.namelist()
    for name in names:
        conf_zip.extract(name)
        one_map = overall_map(name,<tolerance>)[0]#0.3 is usually sufficient as a \
→tolerance
        checks=check_sub(one_map,covered , max_list)
        covered=checks[0]
        max_list=checks[1]
        os.remove(name)

```

```
#initialise list of groups to 'add' to direct product, so can then loop through one by \
→one
maps_sets=max_list.copy()
combo_maps=list(maps_sets[0])

#loop over groups to 'add' forming direct product of previous direct product and new \
→group
for i in maps_sets[1:]:
    #make list of 'combined maps' defining this direct product and fill it up
    new_combo_maps=[]
    for first in combo_maps:
        for second in i:
            combo = combine_maps(first,second,<num atoms in molecule>)
            new_combo_maps.append(combo)
    #update 'current' direct product
    combo_maps=new_combo_maps.copy()

#save final direct product of mappings
combo_maps_array = np.array(combo_maps)
np.save('combined_maps.npy',combo_maps_array)

#####
```

# References

- (1) A. Anelli, E. A. Engel, C. J. Pickard and M. Ceriotti, *Phys. Rev. Mater.*, 2018, **2**, 103804.
- (2) Online Dictionary of Crystallography, 2021, <https://dictionary.iucr.org/Crystal> (accessed 12/09/2025).
- (3) R. DeSando and R. Lange, *Journal of Inorganic and Nuclear Chemistry*, 1966, **28**, 1837–1846.
- (4) Jonas Nyman, Ph.D. Thesis, University of Southampton, 2017.
- (5) C. Hammond, *The Basics of Crystallography and Diffraction*, Oxford University Press, 2015.
- (6) J. Law and R. Rennie, *Dictionary of Chemistry*, Oxford University Press, Eighth Ed., 2020.
- (7) C. Clapham and J. Nicholson, *The Concise Oxford Dictionary of Mathematics*, Oxford University Press, Fifth Ed., 2014.
- (8) M. von Raumer and R. Hilfiker, in *Polymorphism in the Pharmaceutical Industry*, ed. M. von Raumer and R. Hilfiker, John Wiley & Sons, Ltd, 2018, ch. 1, pp. 1–30.
- (9) A. Landi, *The Journal of Physical Chemistry C*, 2019, **123**, 18804–18812.
- (10) A. Pulido, L. Chen, T. Kaczorowski, D. Holden, M. A. Little, S. Y. Chong, B. J. Slater, D. P. McMahon, B. Bonillo, C. J. Stackhouse, A. Stephenson, C. M. Kane, R. Clowes, T. Hasell, A. I. Cooper and G. M. Day, *Nature*, 2017, **543**, 657–664.
- (11) R. L. Greenaway, V. Santolini, A. Pulido, M. A. Little, B. M. Alston, M. E. Briggs, G. M. Day, A. I. Cooper and K. E. Jelfs, *Angew. Chem., Int. Ed. Engl.*, 2019, **58**, 16275–16281.
- (12) Q. Zhu, J. Johal, D. E. Widdowson, Z. Pang, B. Li, C. M. Kane, V. Kurlin, G. M. Day, M. A. Little and A. I. Cooper, *J. Am. Chem.*, 2022, **144**, 9893–9901.
- (13) J. Yang, S. De, J. E. Campbell, S. Li, M. Ceriotti and G. M. Day, *Chem. Mater.*, 2018, **30**, 4361–4371.
- (14) R. Mathew, K. A. Uchman, L. Gkoura, C. J. Pickard and M. Baías, *Magn. Reson. Chem.*, 2020, **58**, 1018–1025.



- (15) R. M. Bhardwaj, J. A. McMahon, J. Nyman, L. S. Price, S. Konar, I. D. H. Oswald, C. R. Pulham, S. L. Price and S. M. Reutzel-Edens, *J. Am. Chem. Society*, 2019, **141**, 13887–13897.
- (16) C. R. Taylor, M. T. Mulvey, D. S. Perenyi, M. R. Probert, G. M. Day and J. W. Steed, *Journal of the American Chemical Society*, 2020, **142**, 16668–16680.
- (17) L. M. Hunnisett et al., *Acta Crystallographica Section B*, 2024, **80**, 517–547.
- (18) L. M. Hunnisett et al., *Acta Crystallographica Section B*, 2024, **80**, 548–574.
- (19) C. J. Pickard and R. J. Needs, *J. Phys. : Condens. Matter*, 2011, **23**, 053201.
- (20) D. H. Case, J. E. Campbell, P. J. Bygrave and G. M. Day, *J. Chem. Theory Comput.*, 2016, **12**, 910–924.
- (21) A. Banerjee, D. Jasrasaria, S. P. Niblett and D. J. Wales, *J. Phys. Chem. A*, 2021, **125**, 3776–3784.
- (22) F. Curtis, X. Li, T. Rose, A. V M, S. Bhattacharya, L. M. Ghiringhelli and N. Marom, *J. Chem. Theory Comput.*, 2018, **14**, 2246–2264.
- (23) C. W. Glass, A. R. Oganov and N. Hansen, *Computer Physics Communications*, 2006, **175**, 713–720.
- (24) D. S. Coombes, S. L. Price, D. J. Willock and M. Leslie, *J. Phys. Chem.*, 1996, **100**, 7352–7360.
- (25) E. O. Pyzer-Knapp, H. P. G. Thompson and G. M. Day, *Acta Crystallogr. Sect. B*, 2016, **72**, 477–487.
- (26) M. P. Metz, M. Shahbaz, H. Song, L. Vogt-Maranto, M. E. Tuckerman and K. Szalewicz, *Cryst. Growth Des.*, 2022, **22**, 1182–1195.
- (27) M. Neumann, *The Journal of Physical Chemistry. B*, 2008, **112**, 9810–29.
- (28) J. Nyman, O. S. Pundyke and G. M. Day, *Phys. Chem. Chem. Phys.*, 2016, **18**, 15828–15837.
- (29) E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev and A. R. Oganov, *Phys. Rev. B*, 2019, **99**, 064114.
- (30) Z. Ye, N. Wang, J. Zhou and D. Ouyang, *The Innovation*, 2024, **5**, 100562.
- (31) R. J. Clements, J. Dickman, J. Johal, J. Martin, J. Glover and G. M. Day, *MRS Bull.*, 2022, **47**, 1054–1062.
- (32) J. Halebian and W. McCrone, *Journal of Pharmaceutical Sciences*, 1969, **58**, 911–929.

- 
- (33) J. Bauer, S. Spanton, R. Henry, J. Quick, W. Dzikowski, W. Porter and J. Morris, *Pharm Res*, 2001, **18**, 859–866.
- (34) G. R. Desiraju, *J Chem Sci*, 2010, **122**, 667–675.
- (35) J. Bernstein and J. MacAlpine, in *Polymorphism in the Pharmaceutical Industry*, ed. M. von Raumer and R. Hilfiker, John Wiley & Sons, Ltd, 2018, ch. 16, pp. 469–483.
- (36) G. J. O. Beran, I. J. Sugden, C. Greenwell, D. H. Bowskill, C. C. Pantelides and C. S. Adjiman, *Chem. Sci.*, 2022, **13**, 1288–1297.
- (37) J. Nyman and G. M. Day, *CrystEngComm*, 2015, **17**, 5154–5165.
- (38) A. R. Oganov, C. J. Pickard, Q. Zhu and R. J. Needs, *Nat. Rev. Mater.*, 2019, **4**, 331–348.
- (39) J. P. M. Lommerse, W. D. S. 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 and D. E. Williams, *Acta Crystallographica Section B*, 2000, **56**, 697–714.
- (40) 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 and D. E. Williams, *Acta Crystallographica Section B*, 2002, **58**, 647–661.
- (41) 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 and P. Verwer, *Acta Crystallographica Section B*, 2005, **61**, 511–527.
- (42) 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 and B. Schweizer, *Acta Crystallographica Section B*, 2009, **65**, 107–125.
- (43) 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 and I. K. Zhitkov, *Acta Crystallographica Section B*, 2011, **67**, 535–551.
- (44) 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 and C. R. Groom, *Acta Crystallographica Section B*, 2016, **72**, 439–459.
- (45) G. J. O. Beran, *Chem. Sci.*, 2023, **14**, 13290–13312.
- (46) J. A. Schmidt, J. A. Weatherby, I. J. Sugden, A. Santana-Bonilla, F. Salerno, M. J. Fuchter, E. R. Johnson, J. Nelson and K. E. Jelfs, *Crystal Growth & Design*, 2021, **21**, 5036–5049.
- (47) C. Wang, C. Zhang, B. You and X. Xue, *Crystal Growth & Design*, 2024, **24**, 38–44.
- (48) M. Kilgour, J. Rogal and M. Tuckerman, *Journal of Chemical Theory and Computation*, 2023, **19**, PMID: 37053511, 4743–4756.
- (49) C. Zeni, R. Pinsler and D. Zügner, *Nature*, 2025, DOI: <https://doi.org/10.1038/s41586-025-08628-5>.
- (50) S. L. Price, *Acta Crystallographica Section B*, 2013, **69**, 313–328.
- (51) S. L. Price, *Phys. Chem. Chem. Phys.*, 2008, **10**, 1996–2009.
- (52) N. Galanakis and M. Tuckerman, *Nature Communications*, 2024, **15**, 9757.
- (53) J. C. Cole, C. R. Groom, M. G. Read, I. Giangreco, P. McCabe, A. M. Reilly and G. P. Shields, *Acta Crystallographica Section B*, 2016, **72**, 530–541.

- 
- (54) C. Taylor, P. Butler and G. Day, *Faraday Discuss.*, 2025, **256**, 434–458.
- (55) G. J. O. Beran, I. J. Sugden, C. Greenwell, D. H. Bowskill, C. C. Pantelides and C. S. Adjiman, *Chem. Sci.*, 2022, **13**, 1288–1297.
- (56) D. A. Lopes, V. Kocevski, T. L. Wilson, E. E. Moore and T. M. Besmann, *Journal of Nuclear Materials*, 2018, **510**, 331–336.
- (57) A. Anelli, PhD Thesis, EPFL, 2020.
- (58) A. Vriza, PhD Thesis, University of Liverpool, 2022.
- (59) A. J. Cruz-Cabeza, S. Karki, L. Fábíán, T. Frišćić, G. M. Day and W. Jones, *Chem. Commun.*, 2010, **46**, 2224–2226.
- (60) J. E. Campbell, J. Yang and G. M. Day, *J. Mater. Chem. C*, 2017, **5**, 7574–7584.
- (61) M. Ceriotti, G. A. Tribello and M. Parrinello, *Proc. Natl. Acad. Sci.*, 2011, **108**, 13023–13028.
- (62) F. Musil, S. De, J. Yang, J. E. Campbell, G. M. Day and M. Ceriotti, *Chem. Sci.*, 2018, **9**, 1289–1300.
- (63) J. Yang, N. Li and S. Li, *CrystEngComm*, 2019, **21**, 6173–6185.
- (64) B. Schölkopf, A. Smola and K.-R. Müller, *Neural Computation*, 1998, **10**, 1299–1319.
- (65) I. Borg and P. Groenen, *Journal of Educational Measurement*, 2003, **40**, 277–280.
- (66) S. M. Moosavi, H. Xu, L. Chen, A. I. Cooper and B. Smit, *Chem. Sci.*, 2020, **11**, 5423–5433.
- (67) C. Zhao, L. Chen, Y. Che, Z. Pang, X. Wu, Y. Lu, H. Liu, G. M. Day and A. I. Cooper, *Nat Commun*, 2021, **12**, 817.
- (68) A. P. Bartók, R. Kondor and G. Csányi, *Phys. Rev. B*, 2013, **87**, 184115.
- (69) mol-CSPy GitLab, <https://gitlab.com/mol-cspy/mol-cspy>, (Accessed: 2025-03-18).
- (70) M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M.

- Klone, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski and D. J. Fox, *Gaussian 09 Revision A.2*, 2009.
- (71) Cramer, C.J, *Essentials of Computational Chemistry Theories and Models*, Wiley, Second Ed., 2004.
- (72) F. Jensen, *Introduction to Computational Chemistry*, John Wiley & Sons, 2017.
- (73) J. P. Perdew, M. Ernzerhof and K. Burke, *J. Chem. Phys.*, 1996, **105**, 9982–9985.
- (74) C. Adamo and V. Barone, *J. Chem. Phys.*, 1999, **110**, 6158–6170.
- (75) R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.
- (76) S. Grimme, *WIREs Computational Molecular Science*, 2011, **1**, 211–228.
- (77) R. A. Buckingham and J. E. Lennard-Jones, *Proc. R. Soc. London, Ser. A*, 1938, **168**, 264–283.
- (78) S.R.Cox, L-Y.Hsu, D.E.Williams, *Acta Crystallogr.* 1981,**37**,293-301.
- (79) S. L. Price, M. Leslie, G. W. A. Welch, M. Habgood, L. S. Price, P. G. Karamertzanis and G. M. Day, *Phys. Chem. Chem. Phys.*, 2010, **12**, 8478–8490.
- (80) A Stone and M Alderton, *Molecular Physics*, 2002, **100**, 221–233.
- (81) A. J. Stone, *J. Chem. Theory Comput.*, 2005, **1**, 1128–1132.
- (82) A. Stone, *The Theory of Intermolecular Forces*, Oxford University Press, 2013.
- (83) P. Pracht, F. Bohle and S. Grimme, *Phys. Chem. Chem. Phys.*, 2020, **22**, 7169–7192.
- (84) J, Bramley, PhD Thesis, University of Southampton, 2025 (submitted).
- (85) H. P. G. Thompson and G. M. Day, *Chem. Sci.*, 2014, **5**, 3173–3182.
- (86) P. J. Winn, G. G. Ferenczy and C. A. Reynolds, *The Journal of Physical Chemistry A*, 1997, **101**, 5437–5445.
- (87) G. G. Ferenczy, P. J. Winn and C. A. Reynolds, *The Journal of Physical Chemistry A*, 1997, **101**, 5446–5455.
- (88) P. Kratzer and J. Neugebauer, *Frontiers in Chemistry*, 2019, **7**, DOI: [10.3389/fchem.2019.00106](https://doi.org/10.3389/fchem.2019.00106).
- (89) R. M. Martin, in *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, 2nd edn., 2020, ch. 4, 81–108.

- 
- (90) D. S. Sholl and J. A. Steckel, in *Density Functional Theory: A Practical Introduction*, Wiley, 2009, ch. 3, pp. 49–81.
- (91) G. Kresse and J. Hafner, *Phys. Rev. B*, 1993, **47**, 558–561.
- (92) G. Kresse and J. Furthmüller, *Computational Materials Science*, 1996, **6**, 15–50.
- (93) G. Kresse and J. Furthmüller, *Phys. Rev. B*, 1996, **54**, 11169–11186.
- (94) J. Moellmann and S. Grimme, *The Journal of Physical Chemistry C*, 2014, **118**, 7615–7621.
- (95) D. Porezag, T. Frauenheim, T. Köhler, G. Seifert and R. Kaschner, *Phys. Rev. B*, 1995, **51**, 12947–12957.
- (96) M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haugk, T. Frauenheim, S. Suhai and G. Seifert, *Phys. Rev. B*, 1998, **58**, 7260–7268.
- (97) B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshayé, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu and T. Frauenheim, *The Journal of Chemical Physics*, 2020, **152**, 124101.
- (98) I. Batatia, D. P. Kovacs, G. Simm, C. Ortner and G. Csanyi, *Advances in Neural Information Processing Systems*, ed. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh, Curran Associates, Inc., 2022, vol. 35, pp. 11423–11436.
- (99) D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole and G. Csányi, *arXiv*, 2025, preprint, <https://arxiv.org/abs/2312.15211> (Accessed 2025-03-30).
- (100) R. J. G. B. Campello, D. Moulavi, A. Zimek and J. Sander, *ACM Trans. Knowl. Discov. Data*, 2015, **10**, DOI: [10.1145/2733381](https://doi.org/10.1145/2733381).
- (101) A. L. Spek, *J. Appl. Crystallogr.*, 2003, **36**, 7–13.
- (102) Y. Iwasaki, A. Kusne and I. Takeuchi, *npj Comput Mater*, 2017, **3**, DOI: <https://doi.org/10.1038/s41524-017-0006-2>.
- (103) P. Cui, D. P. McMahon, P. R. Spackman, B. M. Alston, M. A. Little, G. M. Day and A. I. Cooper, *Chem. Sci.*, 2019, **10**, 9988–9997.
- (104) J. Chisholm and S. Motherwell, *Journal*, 2005, **38**, 228–231.

- (105) R. A. Sykes, N. T. Johnson, C. J. Kingsbury, J. Harter, A. G. P. Maloney, I. J. Sugden, S. C. Ward, I. J. Bruno, S. A. Adcock, P. A. Wood, P. McCabe, A. A. Moldovan, F. Atkinson, I. Giangreco and J. C. Cole, *Journal of Applied Crystallography*, 2024, **57**, 1235–1250.
- (106) J O’ Rourke, *Computational Geometry in C*, Cambridge University Press, Second Ed., 1998.
- (107) P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, *Nature Methods*, 2020, **17**, 261–272.
- (108) C. B. Barber, D. P. Dobkin and H. Huhdanpaa, *ACM Trans. Math. Softw.*, 1996, **22**, 469–483.
- (109) SciPy API reference, <https://docs.scipy.org/doc/scipy/reference/index.html>, (Accessed: 2025-03-19).
- (110) Scikit-Matter: DCH Functionality, [https://github.com/lab-cosmo/scikit-matter/blob/main/src/skmatter/sample\\_selection/\\_base.py](https://github.com/lab-cosmo/scikit-matter/blob/main/src/skmatter/sample_selection/_base.py), (commit 133abe8).
- (111) A. Goscinski, VP. Principe, G. Fraux et al., scikit-matter : A Suite of Generalisable Machine Learning Methods Born out of Chemistry and Materials Science. Open Res Europe 2023, 3:81., [10.12688/openreseurope.15789.2](https://openreseurope.org/10.12688/openreseurope.15789.2).
- (112) S. De, A. P. Bartók, G. Csányi and M. Ceriotti, *Phys. Chem. Chem. Phys.*, 2016, **18**, 13754–13769.
- (113) C.B. Mahmoud, M. Ceriotti, F. Gilberti, K. Goldshtein, T. Junge, M. Stricker, F. Musil and M. Veit, Librascal, 2019, <https://cosmo-epfl.github.io/librascal/index.html> (Accessed 2025-03-30).
- (114) J. Lever, M. Krzywinski and N. Altman, *Nat. Methods*, 2017, **14**, 641–642.
- (115) I. T. Jolliffe and J. Cadima, *Phil. Trans. R. Soc. A.*, 2016, **374**, 20150202.
- (116) S. Raschka and V. Mirjalili, *Python Machine Learning - Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 -3rd Ed*, Packt, 2019.
- (117) M. Deisenroth, A. Faisal and O. C.S., *Mathematics for Machine Learning*, Cambridge University Press, 2020.

- (118) J. Wang, *Computing in Science & Engineering*, 2023, **25**, 4–11.
- (119) A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng and K. W. Jacobsen, *Journal of Physics: Condensed Matter*, 2017, **29**, 273002.
- (120) LAB-COSMO Research Group Website, <https://www.epfl.ch/labs/cosmo/>, (Accessed: 2025-02-03).
- (121) S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson and G. Ceder, *Computational Materials Science*, 2013, **68**, 314–319.
- (122) V. I. Minkin, *Pure and Applied Chemistry*, 1999, **71**, 1919–1981.
- (123) A. A. Hagberg, D. A. Schult and P. J. Swart, Proceedings of the 7th Python in Science Conference, ed. G. Varoquaux, T. Vaught and J. Millman, Pasadena, CA USA, 2008, pp. 11–15.
- (124) C. R. Groom, I. J. Bruno, M. P. Lightfoot and S. C. Ward, *Acta Crystallogr., Sect. B*, 2016, **72**, 171–179.
- (125) C. F. Macrae, I. Sovago, S. J. Cottrell, P. T. A. Galek, P. McCabe, E. Pidcock, M. Platings, G. P. Shields, J. S. Stevens, M. Towler and P. A. Wood, *Journal of Applied Crystallography*, 2020, **53**, 226–235.
- (126) C. Schutte, *The Wave Mechanis of Atoms, Molecules and Ions - An Introduction for Chemistry Students*, Edward Arnold Publishers Ltd, London, 1968.
- (127) R. Nader, A. Bretto, B. Mourad and H. Abbas, *The Journal of Physical Chemistry Letters*, 2019, **755**, 13–28.
- (128) J. F. Humphreys, *A Course in Group Theory*, Oxford University Press, 1996.
- (129) R. Devarapalli, S. B. Kadambi, C.-T. Chen, G. R. Krishna, B. R. Kammari, M. J. Buehler, U. Ramamurty and C. M. Reddy, *Chemistry of Materials*, 2019, **31**, 1391–1402.
- (130) M. Hasegawa and N. Sato, *Mol. Cryst. Liq. Cryst. Sci. Technol., Sect. A*, 1997, **296**, 409–426.
- (131) A. Hamd Hssain, B. Gündüz, A. Majid and N. Bulut, *Chemical Physics Letters*, 2021, **780**, 138918.



- (132) Tojo and J. Mizuguchi, *Zeitschrift für Kristallographie - New Crystal Structures*, 2002, **217**, 253–254.
- (133) A. D. Becke, *J. Chem. Phys.*, 1993, **98**, 5648–5652.
- (134) P. J. Stephens, F. J. Devlin, C. F. Chabalowski and M. J. Frisch, *J. Chem. Phys.*, 1994, **98**, 11623–11627.
- (135) C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B*, 1988, **37**, 785–789.
- (136) R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.
- (137) P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta*, 1973, **28**, 213–222.
- (138) W. J. Hehre, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, 1972, **56**, 2257–2261.
- (139) L Born and G Heywang, *Zeitschrift für Kristallographie*, 1990, **190**, 147–152.
- (140) G. R. Krishna, R. Devarapalli, G. Lal and C. M. Reddy, *J. Am. Chem.*, 2016, **138**, 13561–13567.
- (141) D. G. R. Yeates and R. A. Palmer, *Acta Crystallographica Section B*, 1975, **31**, 1077–1082.
- (142) D. Haubenberger and M. Hallett, *New England Journal of Medicine*, 2018, **378**, PMID: 29742376, 1802–1810.
- (143) R. Payne, R. J. Roberts, R. C. Rowe, M. McPartlin and A. Bashal, *Int. j. pharm*, 1996, **145**, 165–173.
- (144) H.-H. Chen and C.-S. Su, *Org. Process Res. Dev.*, 2016, **20**, 878–887.
- (145) G. M. Day, W. D. S. Motherwell and W. Jones, *Phys. Chem. Chem. Phys.*, 2007, **9**, 1693.
- (146) R. Cuadrado, J. Glover, J. Bramley, C. Taylor and G. Day, *In preperation*.
- (147) M. Gaus, A. Goez and M. Elstner, *Journal of Chemical Theory and Computation*, 2013, **9**, PMID: 26589037, 338–354.
- (148) Y. Le Page, *Journal of Applied Crystallography*, 1987, **20**, 264–269.
- (149) Automeris.io : WebPlotDigitizer Software, <https://automeris.io/>, (Accessed: 2025-02-18).
- (150) J. Evans, in *Crystal Structure Analysis: Principles and Practice (Chapter 17)*, Oxford University Press, 2009.
- (151) *Personal Communication from Chie-Shaan Su to Graeme Day*, (2023-08-17).
- (152) R. A. Mayo, M. Marczenko, Katherine and E. R. Johnson, *Chem. Sci.*, 2023, **14**, 4777–4785.

- (153) A. O. de-la Roza, E. R. Johnson and V. Luaña, *Computer Physics Communications*, 2014, **185**, 1007–1018.
- (154) T. Stolar, J. Alić, I. Lončarić, M. Etter, D. Jung, O. K. Farha, I. Đilović, E. Meštrović and K. Užarević, *CrystEngComm*, 2022, **24**, 6505–6511.
- (155) O. Elishav, R. Podgaetsky, O. Meikler and B. Hirshberg, *The Journal of Physical Chemistry Letters*, 2023, **14**, 971–976.
- (156) A.K.Rappe, C.J.Casewit, K.S.Cowell, W. III and W.M.Skiff, *Journal of the American Chemical Society*, 1992, **114**, 10024–10035.
- (157) M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek and G. R. Hutchin-son, *Journal of Chemical Informatics*, 2012, **4**.
- (158) A. T. Nielsen, A. P. Chafin, S. L. Christian, D. W. Moore, M. P. Nadler, R. A. Nissan, D. J. Vanderah, R. D. Gilardi, C. F. George and J. L. Flippen-Anderson, *Tetrahedron*, 1998, **54**, 11793–11812.
- (159) ASE:Release Notes, <https://wiki.fysik.dtu.dk/ase/releasenotes.html>, (Accessed: 2025-02-18).
- (160) P. Eastman, P. K. Behara, D. L. Dotson, R. Galvelis, J. E. Herr, J. T. Horton, Y. Mao, J. D. Chodera, B. P. Pritchard, Y. Wang, G. D. Fabritiis and T. E. Markland, *Sci Data*, 2023, **10**, DOI: <https://doi.org/10.1038/s41597-022-01882-6>.
- (161) D. I. A. Millar, H. E. Maynard-Casely, A. K. Kleppe, W. G. Marshall, C. R. Pulham and A. S. Cumming, *CrystEngComm*, 2010, **12**, 2524–2527.
- (162) D. E. Widdowson and V. A. Kurlin, *Crystal Growth & Design*, 2024, **24**, 5627–5636.
- (163) *GRACE (version 2.4)*, Avant-garde Materials Simulation Deutschland GmbH.
- (164) A. Anelli, GCH, 2018, <https://github.com/andreanelli/GCH>, (Accessed 2025-02-23).
- (165) A. M. Reilly and A. Tkatchenko, *The Journal of Chemical Physics*, 2013, **139**, 024705.
- (166) M. G. Kendall, *Biometrika*, 1938, **30**, 81–93.
- (167) F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *Journal of Machine Learning Research*, 2011, **12**, 2825–2830.
- (168) B. Tian, H. Hao, X. Huang, T. Wang, J. Wang, J. Yang, X. Li, W. Li, L. Zhou and N. Wang, *Crystal Growth & Design*, 2023, **23**, 7266–7275.

## REFERENCES

---

- (169) T. Lohith, M. Hema, C. Karthik, S. S, J. R. Rajabathar, M. Karnan, N. Lokanath, L. Malle-sha, P. Mallu and M. Sridhar, *Journal of Molecular Structure*, 2023, **1289**, 135841.
- (170) T. F. Willems, C. H. Rycroft, M. Kazi, J. C. Meza and M. Haranczyk, *Microporous and Mesoporous Materials*, 2012, **149**, 134–141.
- (171) P. Cui, D. P. McMahon, P. R. Spackman, B. M. Alston, M. A. Little, G. M. Day and A. I. Cooper, *Chem. Sci.*, 2019, **10**, 9988–9997.
- (172) R. Thakuria, B. Sarma and A. Nangia, in *Comprehensive Supramolecular Chemistry II*, ed. J. L. Atwood, Elsevier, Oxford, 2017, pp. 25–48.
- (173) U. Timothy C., *Statistics in Plain English*. Routledge, 2022, vol. Fifth edition.
- (174) sklearn User Guide Website, [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html), (Accessed: 2025-03-10).
- (175) O. Egorova, R. Hafizi, D. C. Woods and G. M. Day, *The Journal of Physical Chemistry A*, 2020, **124**, PMID: 32881496, 8065–8078.
- (176) sklearn github issue : precomputed kernels, <https://github.com/scikit-learn/scikit-learn/issues/8445>, (Accessed: 2025-01-30).
- (177) M. R. Ward, C. R. Taylor, M. T. Mulvee, G. I. Lampronti, A. M. Belenguer, J. W. Steed, G. M. Day and I. D. H. Oswald, *Crystal Growth & Design*, 2023, **23**, 7217–7230.
- (178) X. Yin and C. E. Gounaris, *Current Opinion in Chemical Engineering*, 2022, **35**, 100726.
- (179) E. Pidcock and W. D. S. Motherwell, *Chem. Commun.*, 2003, 3028–3029.
- (180) PubChem-align3d Github Repository, <https://github.com/ncbi/pubchem-align3d>, (Accessed: 2025-02-12).
- (181) J. Johal and G. Day, *unknown*, unpublished work.
- (182) R. K. Cersonsky, M. Pakhnova, E. A. Engel and M. Ceriotti, *Chem. Sci.*, 2023, **14**, 1272–1285.
- (183) M. O’Shaughnessy, J. Glover, H. Roohollah, M. Barhi, R. Clowes, S. Chong, S. Argent, G. Day and A. Cooper, *Nature*, 2023, **630**, 102–108.