Computational Approaches to Drug Profiling and Drug-Protein Interactions

Oliver B. Scott

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

March 2022
Declaration of Authorship

I, OLIVER SCOTT, declare that this thesis titled, ‘Computational Approaches to Drug Profiling and Drug-Protein Interactions’ and the work presented in it are my own. 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 previously been submitted for a degree or any other qualification 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 the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:  

Date:  31/03/2022
Despite substantial increases in R&D spending within the pharmaceutical industry, de-novo drug design has become a time-consuming endeavour. High attrition rates led to a long period of stagnation in drug approvals. Due to the extreme costs associated with introducing a drug to the market, locating and understanding the reasons for clinical failure is key to future productivity. As part of this PhD, three main contributions were made in this respect. First, the web platform, LigNFam enables users to interactively explore similarity relationships between ‘drug like’ molecules and the proteins they bind. Secondly, two deep-learning-based binding site comparison tools were developed, competing with the state-of-the-art over benchmark datasets. The models have the ability to predict off-target interactions and potential candidates for target-based drug repurposing. Finally, the open-source ScaffoldGraph software was presented for the analysis of hierarchical scaffold relationships and has already been used in multiple projects, including integration into a virtual screening pipeline to increase the tractability of ultra-large screening experiments. Together, and with existing tools, the contributions made will aid in the understanding of drug-protein relationships, particularly in the fields of off-target prediction and drug repurposing, helping to design better drugs faster.
Impact Statement

As part of this PhD two formulations of geometric deep learning have been applied to protein structural data. Both methods explore novel paradigms that have previously not been applied to protein data. The second proposed method utilises the protein solvent excluded surface, a representation that has been underused with respect to the application of deep learning on protein structures. We hope that our ideas help inspire others to explore the potential that this technique may hold within the field of drug-discovery. We believe that the approach could be applied at numerous different stages of the drug discovery process.

The newly developed Python library ScaffoldGraph, aids the construction of drug discovery workflows, specifically granting the ability to perform complex analyses on hierarchical scaffold relationships. The software is open source, with an emphasis on usability and flexibility, being able to drop into existing workflows with ease. The software has already found use in an industrial setting as part of a pipeline for the discovery of central nervous system treatments. The software was able to process 280 million molecules in order to generate a diverse set appropriate for virtual screening within a tractable time frame.
Acknowledgements

First and foremost, I would like to thank my research supervisor, Dr. Edith Chan for her invaluable advice, knowledge, and tenacity throughout my PhD study. Her broad knowledge and experience have been integral throughout this process.

I would also like to thank Professor Dave Selwood and Dr. Roman Laskowski for their invaluable insight and outside angle. All the members in my lab, with a special thanks to Clara Gathmann, for listening to my outlandish ideas, for her kind words and support, especially towards the end.

To my incredible friends Vic Saunders, Abbie Rogan and Pejvak Khorram Din, whose help, support and friendship have made my study and life at UCL unforgettable. I express eternal gratitude to my parents Clare and Andrew, two brothers Christian and Alister, Grandfather Andrew and long-term partner Samantha, without their understanding and encouragement, it would have been impossible for me to complete my study.

Finally, I would like to extend thanks to the LIDo program and my industry partner Cresset for the opportunity to undertake this rewarding project. Working with Cresset and Dr. Mark Mackey as part of an internship programme was extremely rewarding and has helped shine a light on the next stages of my career.
Table of Contents

I LIGNFAM: AN INTERACTIVE INTERFACE FOR THEN EXPLORATION OF DRUG-PROTEIN RELATIONSHIPS 1

1 Chapter 1: LigNFam 1

1.1 Introduction and Motivation 1

1.1.1 Drug repurposing 2

1.1.1.1 The Three Paradigms of Drug Repurposing 5

1.1.1.2 Systematic Drug Repurposing 6

1.1.2 Molecular Similarity 9

1.1.2.1 Molecular Representation 11

1.1.2.2 Superposition 13

1.1.2.3 Molecular/Chemical Descriptors 14

1.1.2.4 Similarity Coefficients 17

1.1.3 Application of Similarity Principles to Biological Molecules 19

1.1.3.1 Protein Homology 20

1.1.3.2 Sequence-Based Classification 22

1.1.3.3 Structure-Based Classification 24

1.1.3.4 Functional Classification 25

1.1.3.5 Binding Site Similarity 27

1.2 Aims 28

1.3 Methodology 30

1.3.1 Database Construction 32

1.3.1.1 Ligand database 32

1.3.1.2 BLAST database 33

1.3.1.3 Protein-ligand interaction database 34

1.3.2 Ligand-based search 35

1.3.2.1 Similarity searching 35

1.3.2.2 Clustering 39

1.3.3 Protein-Based Search 40

1.3.3.1 Similarity Searching 41

1.3.3.2 Clustering 42

1.3.3.3 Web Interface 46

1.3.3.4 Backend 46

1.4 Results and Discussion 49

1.4.1.1 Database 51

1.4.1.2 Ligand-Based Search 55

1.4.1.3 Protein-Based Search 56

1.4.1.4 Interface 59

1.5 Conclusions 64
## Chapter 2: Introduction and Motivation

### 2.1 The Anatomy of a Binding Site Comparison

### 2.2 Evaluation of Binding Site Comparison Algorithms

### 2.3 Binding Site Databases

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>ProSPECTs</td>
</tr>
<tr>
<td>2.3.2</td>
<td>TOUGH-M1</td>
</tr>
<tr>
<td>2.3.3</td>
<td>TOUGH-C1</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Vertex</td>
</tr>
<tr>
<td>2.3.5</td>
<td>sc-PDB</td>
</tr>
<tr>
<td>2.3.6</td>
<td>Dataset Redundancy</td>
</tr>
</tbody>
</table>

### 2.4 The Application of Machine Learning to Binding Site Comparisons

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>2.4.1.1</td>
<td>Supervised Learning</td>
</tr>
<tr>
<td>2.4.1.2</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>2.4.1.3</td>
<td>Weakly-Supervised Learning</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>2.4.2.1</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>2.4.2.2</td>
<td>Gradient Decent</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Geometric Deep Learning</td>
</tr>
<tr>
<td>2.4.3.1</td>
<td>Symmetries, invariants and equivariant</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Application of Machine Learning to Biological/Chemical problems</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Application of Machine Learning to Binding Site Comparison</td>
</tr>
<tr>
<td>2.4.5.1</td>
<td>DeeplyTough</td>
</tr>
<tr>
<td>2.4.6</td>
<td>Proposal</td>
</tr>
</tbody>
</table>

## Chapter 3: ToughGraph: Weakly Supervised Learning of Binding Site Representations Using Graph Neural Networks

### 3.1 Introduction

### 3.1.1 Graph Data

### 3.1.2 Graph Similarity

### 3.1.3 Graph Neural Networks

### 3.1.4 Graph Matching Neural Networks

### 3.1.5 Approach

### 3.2 Methodology

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1</td>
<td>Modelling Protein Binding Sites</td>
</tr>
<tr>
<td>3.2.1.1</td>
<td>Pseudocenter Construction</td>
</tr>
<tr>
<td>3.2.1.2</td>
<td>Feature Computation</td>
</tr>
<tr>
<td>3.2.1.3</td>
<td>Graph Construction</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Model</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>Graph Embedding</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>Graph Matching</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Training Dataset</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Training Procedure and Parameters</td>
</tr>
</tbody>
</table>

### 3.3 Results and Discussion
IV PART 3: SCAFFOLDGRAPH: HIERARCHICAL SCAFFOLD RELATIONSHIPS
Chapter 5: ScaffoldGraph Software

5.1 Introduction

5.1.1 Molecular Scaffolds

5.1.2 Scaffold Definition

5.1.3 Hierarchical Scaffold Relationships

5.2 ScaffoldGraph

5.2.1 Technical Details

5.2.2 Features

5.2.2.1 Representations

5.2.2.2 Analysis

5.2.2.3 Visualisation

5.2.2.4 Miscellaneous

5.2.2.5 Command Line Interface

5.2.3 Benchmarking

5.2.3.1 Technical Details

5.2.3.2 Results

5.3 Case Studies

5.3.1 Diverse Compound Selection

5.3.2 Privileged Scaffold Identification

5.4 Conclusion

REFERENCES
List of Figures

Figure 1. 1 A bar plot depicting the number of articles within PubMed mentioning drug repositioning or a related term within the title or abstract between 2008 and 2021... 4
Figure 1. 2 An illustration of the master pipeline for the LigNFam project.......................... 31
Figure 1. 3 The backend logic for the LigNFam web interface............................................. 48
Figure 1. 4 An illustration of the implemented ligand search pipeline.............................. 57
Figure 1. 5 An illustration of the implemented ligand search pipeline.............................. 58
Figure 1. 6 Screenshots of the LigNFam interface ................................................................. 61
Figure 1. 7 Screenshots of the LigNFam results interface.................................................... 63
Figure 1. 8 Screenshot of the LigNFam error page ............................................................... 64

Figure 2. 1 An illustration of the protocol used by Govindaraj and Brylinski\textsuperscript{123} to construct the TOUGH-M1 dataset .............................................................................. 81
Figure 2. 2 A Venn diagram displaying the redundancy within binding site comparison benchmark datasets based on Uniprot accession codes............................................................. 86
Figure 2. 3 Illustrates a three-layer artificial neural network (ANN) with one hidden layer .................................................................................................................................................. 95

Figure 3. 1 An illustration of the proposed graph models, adapted from Li et al.\textsuperscript{183} ...... 111
Figure 3. 2 ROC (a) and PR (b) plots with associated AUC and AP values for the evaluation of the performance of binding site comparison algorithms on validation folds of TOUGH-M1. ........................................................................................................ 129
Figure 3. 3 Virtual screening results for three estrogen receptors (ER\textalpha{}, ER\textbeta{} and ERR\textgamma{}) bound to 4-hydroxy-tamoxifen ......................................................................................... 142
Figure 3. 4 A flavone-like scaffold, and two drug molecules; Flavopiridol and Genistein containing it. ...................................................................................................................................... 145
Figure 4. 1 An illustration of a point-pair feature (PPF) calculation defined between two surface points $m_1$ and $m_2$................................................................. 156
Figure 4. 2 Illustration depicting the two-ring neighbourhood of vertex $v$................. 160
Figure 4. 3 Auto-encoder architecture for the unsupervised learning of transformation invariant local protein surface descriptors. .......................................................... 164
Figure 4. 4 Illustration of the Bag-of-Features (BoF) approach for the computation of a global descriptor from a collection of features sampled at local regions on a shape or molecular surface.............................................................. 170
Figure 4. 5 Illustration of Bag of Features (BoF) histograms for three protein-ligand binding sites ........................................................................................................ 173
Figure 4. 6 Training results for the PPF-FoldNet model trained on local regions of molecular surfaces. .................................................................................. 182
Figure 4. 7 A PaCMAP\textsuperscript{220} dimensionality reduction calculated from encoded local regions (patches) on protein molecular surfaces........................................ 184
Figure 4. 8 Visualisation of latent features upon the molecular surfaces of two protein-ligand binding sites of the enzyme carbonic anhydrase II; (a) 1A42, (b) 1ZE8.... 185
Figure 4. 9 Elbow plot displaying the inertia for the various cluster sizes used to compute a BoF codebook. ................................................................................. 186
Figure 4. 10 Grid search results (AUC) for SurfsUp using a soft-quantization approach and no size dependent scaling, on each ProSPECCTs subset.................... 189
Figure 4. 11 Grid search results (AP) for SurfsUp using a soft-quantization approach and no size dependent scaling, on each ProSPECCTs subset. ................... 190
Figure 4. 12 Boxplots displaying the distance dependent distinction between positive and negative pairs of binding sites, in the P1 dataset (a) and the P7 dataset (b). ....... 196
Figure 4. 13 ROC and PR curves illustrating the retrieval results of carbonic anhydrase II (EC 4.2.1.1) binding sites using the SurfsUp algorithm. ......................... 208
Figure 4. 14 The top three kinase inhibitor hits against synapsin-1 identified by\textsuperscript{228} not including the pan-kinase inhibitor staurosporine......................................... 213
Figure 4. 15 A RANSAC surface alignment between synapsin-1 (1AUX), bound to ATP-$\gamma$S and casein kinase II subunit $\alpha$ (1PJK) bound to phosphomethylphosphonic acid adenylate ester (ANP). ................................................................. 217
Figure 4. 16 Virtual screening results for a collagenase-3 (MMP-13) binding site (1FM1) screened versus the sc-PDB database. ................................................................. 223

Figure 4. 17 A RANSAC surface alignment between MMP-13 (1FM1) and ADAMTS5 (2RJP) ........................................................................................................... 226

Figure 4. 18 Re-scoring of virtual screening results for a collagenase-3 (MMP-13) binding site (1FM1) screened versus the sc-PDB database. ................................. 228

Figure 5. 1 An illustration of various hierarchical scaffold relationships constructed from the molecule, Amodiaquine. .................................................................................. 243

Figure 5. 2 A screen capture of ScaffoldsGraphs Cytoscape-based visualisation tool which functions within a Jupyter notebook environment. ........................................... 248

Figure 5. 3 Performance benchmark for the scaffold tree, network and HierS algorithms implemented in ScaffoldGraph. ................................................................. 252

Figure 5. 4 Performance benchmark for the scaffold tree, network and HierS algorithms implemented in ScaffoldGraph, parallelised over multiple CPU cores. .............. 253

Figure 5. 5 An illustration of the pipeline used by Orion Pharma for selecting a diverse CNS relevant subset from the Enamine REAL enumerated database. .................. 256

Figure 5. 6 Chemical structures of 2-biphenyl-tertrazole and biphenyl, both identified as privileged scaffolds using a bioactivity enrichment-based approach .................. 262
List of Tables

Table 1. 1 A broad summary of data sources commonly integrated into repositioning pipelines. ........................................................................................................... 8
Table 1. 2 Lipinski\textsuperscript{62} and extended compound filtering rules\textsuperscript{90} applied to retain only ‘drug like’ molecules in the constructed database................................................. 33
Table 1. 3 A list of the pharmacophore feature types computed by the Align-it (Pharao) software............................................................................................................. 38

Table 2. 1 Examples of binding site comparison tools using diverse approaches......... 71
Table 2. 2 Summary of datasets contained within the ProSPECCTs benchmark dataset collection\textsuperscript{130} to analyse the strengths and weaknesses of binding site comparison tools. ........................................................................................................... 79
Table 2. 3 Summary of the TOUGH-C1 dataset\textsuperscript{151} for distinguishing protein binding sites by interacting ligand classification. ................................................................. 83
Table 2. 4 Table showing the number of entries and unique entries in terms of Uniprot accession codes, evaluating the redundancy within the datasets themselves. .......... 87

Table 3. 1 AUC values for the ToughGraph models on each of the 10 ProSPECCTs data subsets. ............................................................................................................. 133
Table 3. 2 Average precision values for the ToughGraph models on each of the 10 ProSPECCTs data subsets ............................................................................................................. 134
Table 3. 3 AUC and AP values for evaluating ToughGraph and other binding site comparison algorithms on the vertex dataset........................................................................ 137
Table 3. 4 The number of estrogen and estrogen-related receptors included in the sc-PDB database............................................................................................................. 140
Table 3. 5 Virtual screening results using an estrogen receptor alpha binding site (3ERT) against the sc-PDB database........................................................................ 143
Table 3. 6 Virtual screening results using an estrogen receptor beta binding site (1X7B) against the sc-PDB database. ................................................................. 143
Table 3. 7 Virtual screening results using an estrogen-related receptor gamma binding site (1S9Q) against the sc-PDB database. ......................................................... 144

Table 4. 1 The Kyte-Doolittle scale\textsuperscript{208} for defining amino acid hydrophobicity, originally intended for identifying both surface-exposed and transmembrane regions from a protein sequence.................................................................................................................. 159
Table 4. 2 AUC results for DeeplyTough\textsuperscript{131}, ToughGraph and all variants of the SurfsUp algorithm. ...................................................................................................................... 191
Table 4. 3 Average precision results for DeeplyTough\textsuperscript{131}, ToughGraph and all variants of the SurfsUp algorithm............................................................................................................. 192
Table 4. 4 Sensitivity and specificity analysis for datasets P1 and P7. ......................... 198
Table 4. 5 Performance of various algorithms for classifying ligand binding sites included in the TOUGH-C1 dataset\textsuperscript{151}. ........................................................................................................ 202
Table 4. 6 Enrichment factors for the SurfsUp algorithm with respect to the retrieval of nucleotide (A) and heme (B) binding sites within a diverse set of control structures defined within the TOUGH-C1 dataset\textsuperscript{151}. ............................................. 204
Table 4. 7 The ability of SurfsUp (BoF) to retrieve binding sites with the same Enzyme Commission (EC) number within the top ranks. ......................................................... 209
Table 4. 8 \textit{In vitro} screening results of various kinase inhibitors against synapsin-1 as reported by Defranchi et al.\textsuperscript{228} ....................................................................................................................... 212
Table 4. 9 The top ten ranked hits from a virtual screening against the sc-PDB database\textsuperscript{147} using a synapsin-1 query (1AUX) bound to phosphothiophosphoric acid-adenylate ester (ATP-\(\gamma\)S). .......................................................................................................................... 214
Table 4. 10 Iterative AUC scores for select protein kinases (\(N > 10\)) when performing a virtual screen using a synapsin-1 binding site (1AUX) against the sc-PDB database. ....................................................................................................................... 216
Table 4. 11 A summary of metalloproteinases within the metzincin superfamily contained within the sc-PDB database. ......................................................................................... 220
Table 4. 12 Iterative AUC scores for metalloproteinase ($N > 10$) enzymes when performing a virtual screen using a collagenase-3 (MMP-13) enzyme binding site (1FM1) against the sc-PDB database.

Table 4. 13 Runtimes for each stage of a SurfsUp computation.

Table 5. 1 Results of the privileged scaffold identification approach, displaying the number of subfamilies for each super family, the number of molecules, bioactivity data points, scaffolds and identified privileged scaffolds within the superfamily.

Table 5. 2 Identified privileged scaffolds: (a) biphenyl, and (b) biphenyl-tetrazole.

Abbreviations

ANN - Artificial Neural Network
BOF - Bag-Of-Features
BM - Bemis And Murcko Scaffold/Framework
CLI - Command-Line Interface
CSE - Compound Set Enrichment
CNN - Convolutional Neural Network
CSK - Cyclic Skeleton
DAG - Directed Acyclic Graph
ESP - Electrostatic Potential Value
ECDF - Empirical Cumulative Distribution Function
EF - Enrichment Factor
EC - Enzyme Commission
ECFP - Extended Connectivity Fingerprint
FPR - False Positive Rate
GDL - Geometric Deep Learning
GMN - Graph Matching Network
GUI - Graphical User Interface
HET - Heterogen Atoms
HMM – Hidden Markov Model
HTS - High Throughput Screening
KS - Kolmogorov-Smirnov
MCS - Maximum Common Substructure
MLP - Multi-Layer Perceptron
NLP - Natural Language Processing
NN - Nearest Neighbour
PPF - Point Pair Features
PS - Privileged Scaffold
PDB - Protein Data Bank
RMSE - Root-Mean-Square-Error
SN - Scaffold Network
ST - Scaffold Tree
SPP - Similarity Property Principle
SAS - Solvent-Accessible Surface
SES - Solvent-Excluded Surface
SDF - Structure Data File
SAR - Structure-Activity Relationship
TC - Tanimoto Coefficient
TFPS - Target-Family Privileged Scaffolds
TPR - True Positive Rate
Chapter 1: LigNFam

1.1 Introduction and Motivation

*De-novo* drug discovery is an expensive and time-consuming process. On average bringing a drug to from development to market can cost upwards of two to three billion dollars in investment, taking 9-15 years from the initial discovery to the approval stage. The process is also high-risk with very high attrition rates. In fact, only 10% of the drugs that make it to clinical trials are approved by the regularity agencies, with the rest failing due to low efficacy or high toxicity due to the limited predictive power of disease models in preclinical studies. Despite the large increases in research and development spending, the number of new drug approvals became stagnant even with investment into novel drug discovery technologies including: combinatorial chemistry, structure-based drug design, high-throughput screening and genomics promising to increase productivity. This stagnation has since seen a turn-around with drug approvals in 2017 hitting a two-decade peak. With these associated costs, time and risk a clinical failure is costly, and thus improving R&D productivity is one of, if not the most, important priority within the industry. One of the
main drivers behind high rates of attrition is the promiscuity of small molecules\textsuperscript{7}. We are now aware that the traditional “one drug, one target” paradigm is fundamentally outdated with studies proposing, that on average molecules interact with at least six to eleven targets within biological systems\textsuperscript{8}. The lack of selectivity can lead to adverse reactions including: side-effects, toxicity and drug-drug interactions\textsuperscript{9}. Non-intended targets which interact with a drug are known as off-targets, and the prediction of these will aid in the design of more selective drugs and boost productivity. In addition to, the associated adverse effects, off-targets may also present a unique opportunity for target hopping, in which a target is implicated in a different disease pathway. This process is one of the methods often used for drug-repositioning, with opportunities identified using data-driven approaches.

1.1.1 Drug repurposing

Ashburn and Thor\textsuperscript{7} defined drug repurposing as: “The process of finding new uses outside the scope of the original medical indication for existing drugs”. The term is often used interchangeably with; drug repositioning (occasionally also referred to as ‘drug reprofiling’, ‘drug-retasking’ and ‘therapeutic switching’)\textsuperscript{8}. While most commonly repurposing applies to approved drugs, repurposing may also be applied to drugs in pre-clinical development, drugs which have failed phase I or II clinical trials due to efficacy issues, shelved drugs due to lack of investment or by academic institutions\textsuperscript{10} and drugs which may have been withdrawn due to safety concerns for a particular patient cohort\textsuperscript{7}. 
The premise behind repositioning is thus simple; considering that approved compounds will have known absorption, distribution metabolism and excretion (ADME) properties, constructed safety-profiles, known manufacturing procedures and are also likely to have a degree of characterized pharmacology, identifying new indications for these compounds will substantially cut the associated costs and time taken for the drug to reach the market. Due to the simple nature of this concept, it may not be surprising that in 2017 30% of newly approved drugs were repositioned while only 6% were first-in-class agents. This data is further reinforced by the literature with the number of articles relating to drug repositioning increasing every year since 2008 as shown in Figure 1.1.

The benefits repositioning has over *de-novo* discovery is clear. Repositioning candidates will have, to an extent, known pharmacological profiles, preclinical research data, safety profiles and potentially a developed formulation, thus, accelerating research into late-stage development. This initial acceleration can significantly shorten the development process and associated risk, considering approximately 30% of drugs fail due to toxicity issues. In addition to safety and time benefits, the cost of bringing a repositioned drug to market is significantly less than traditional discovery. While there are discrepancies in the literature, repositioning has been estimated to cost approximately $300 million compared to up to $3 billion for a first-in-class therapeutic. Repositioning is also particularly beneficial in the case of rare diseases. Currently, there are greater than 7000 recognized rare diseases with 95% lacking an FDA approved treatment. These diseases are often poorly characterized pathophysiologically, with knowledge of disease development pathways, lacking or non-existent.
Figure 1. A bar plot depicting the number of articles within PubMed mentioning drug repositioning or a related term within the title or abstract between 2008 and 2021.

For the most part repositioning has been achieved through serendipity, through retrospective clinical analysis or has resulted from an increased understanding of a drug’s mechanism of action. Indeed, two of the most famous repositioning examples were discovered by chance; Sildenafil was originally developed by Pfizer as an angina medication in 1989 and repurposed and marketed as Viagra for the treatment of erectile dysfunction\textsuperscript{7}, and, Thalidomide was repurposed as a leprosy treatment after the tragic effects of its initial launch in 1957, as a treatment for morning sickness\textsuperscript{13}. Viagra has since also been approved for the treatment of pulmonary hypertension\textsuperscript{14}, while Thalidomide has been rebranded under the trade name, Thalidomid, for the treatment of multiple cancers\textsuperscript{15}. Due to successes of repositioning, systematic approaches attempting to identify candidates, utilising existing knowledge are becoming increasingly popular\textsuperscript{16}. The rapid increase of
publicly available data and computational processing power has certainly helped facilitate this development.

1.1.1.1 The Three Paradigms of Drug Repurposing

Three different paradigms enable the systematic identification of putative drug repurposing candidates. The first has been previously mentioned and relates to the inherent promiscuity of drug molecules. This observation is referred to as promiscuity, where a target other than the intended primary target is called on off-target. Identifying these off-target effects can provide opportunities for drug repositioning. A bibliometric study of PubMed found that over 60% of drugs or drug candidates have been studied in more than one disease with 189 of these studied in over 300 diseases each\textsuperscript{17}.

A second observation is that a particular biological entity may also be involved in multiple biological processes, performing multiple functions. Traditionally, it was believed that the best way to develop a therapeutic was to design compounds towards a specific disease relevant target. However, biological systems are inherently complex whereby proteins often function in the context of signaling pathways forming complex communication networks\textsuperscript{18}. As such, an understanding of relevant signaling cascades can aid in the discovery of repositioning opportunities.

The final observation relates to drugs in clinical use; to get to this stage a drug must have been through three phases of clinical trials. These trials are usually related to a particular disease, although it is possible for a drug to be effective without an understanding
of target interactions. In a clinical environment, it is common for the prescription of a drug to evolve for use in other indications, regardless of its legally approved application, through clinical observation. This practice is known as ‘off-label prescription’.

1.1.1.2 Systematic Drug Repurposing

Systematic drug repurposing approaches make use of the aforementioned paradigms in order to evaluate putative candidates. These approaches can be broadly categorized into experimental and computational approaches although both streams are inherently synergistic. A full review of all proposed repurposing prediction methods is beyond the scope of this thesis; however, data sources and broadly applicable computational approaches will be briefly discussed. After which the principles of molecular similarity are reviewed since it plays a large role in the LigNFam interface and indeed the entirety of this thesis. The reader is made aware of recent in-depth reviews highlighting the progress within the field.

Since experimental repositioning evaluation is a time consuming, costly and inefficient endeavor, computational approaches have been popularized within the drug repurposing domain. The advent of databases containing large amounts of biological and chemical data, and the application of computational methods including: machine learning, network analysis, text mining, and high throughput data analysis has increased the potential to identify repositioning opportunities. Computational methods aim to predict an association between a disease indication and compound through the analysis of biomedical
data. This can be achieved from a drug-centric perspective by identifying novel indications for existing drugs, or from a disease-centric perspective by identifying drugs for a particular disease. Most methods rely on a derived similarity-metric from the properties being studied, to build a predictive model establishing new links between drugs and novel disease indications. Approaches exist integrating genomic\textsuperscript{23–25}, phenomic\textsuperscript{26–28} chemical structure\textsuperscript{29–33} and protein structure\textsuperscript{34–36} data. Often approaches may consider multiple data sources to increase prediction capability. In Table. 1.1 a list of potential data sources for drug repositioning is displayed although this is by no means exhaustive.
Table 1. A broad summary of data sources commonly integrated into repositioning pipelines. Data sources may either be used for formulating or validating a generated prediction.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Gene Expression Omnibus (GEO)</em>[^37]</td>
<td>Functional Genomics Data</td>
</tr>
<tr>
<td><em>Gene Expression Atlas</em>[^38]</td>
<td>Gene Expression Patterns under various conditions</td>
</tr>
<tr>
<td><em>The Cancer Genome Atlas (TGCA)</em>[^39]</td>
<td>Genomic data of &gt;10000 patient tissue samples</td>
</tr>
<tr>
<td><em>The Connectivity map (CMAP)</em>[^40,^41]</td>
<td>Gene expression profiles of &gt;1000 drugs</td>
</tr>
<tr>
<td><em>Library of Integrated Network-based cellular signatures (LINCS)</em>[^42]</td>
<td>&gt;1 million gene expression profiles of drugs and genetic perturbagens</td>
</tr>
<tr>
<td><em>GeneSigDB</em>[^43]</td>
<td>Collections of annotated gene signatures</td>
</tr>
<tr>
<td><em>SIDER</em>[^44]</td>
<td>Registry and results databases of clinical studies</td>
</tr>
<tr>
<td><em>ClinicalTrials.gov</em>[^45]</td>
<td>Genetic variation on drug response</td>
</tr>
<tr>
<td><em>PharmGKB</em>[^46]</td>
<td>3D structures of proteins and nucleic acids</td>
</tr>
<tr>
<td><em>Protein Data Bank (PDB)</em>[^47]</td>
<td>Detailed approved drug pharmacological data</td>
</tr>
<tr>
<td><em>DrugBank</em>[^48]</td>
<td>Literature derived database of molecules and molecule-protein interactions</td>
</tr>
<tr>
<td><em>ChEMBL</em>[^49]</td>
<td>Biological assay result database</td>
</tr>
<tr>
<td><em>PubChem</em>[^50]</td>
<td>Potential drug targets associated with disease</td>
</tr>
<tr>
<td><em>OpenTargets</em>[^51]</td>
<td>Protein-Protein interaction, analysis, and networks</td>
</tr>
<tr>
<td><em>STRING</em>[^52]</td>
<td>68,000 chemicals, interactions and over 1.5 million proteins in 373 species</td>
</tr>
<tr>
<td><em>STITCH</em>[^53]</td>
<td>Collection of databases dealing with genomes, biological pathways, diseases, drugs, and chemical substances</td>
</tr>
<tr>
<td><em>KEGG</em>[^54]</td>
<td>Collection of approved and failed drug-disease associations</td>
</tr>
<tr>
<td><em>RepoDB</em>[^55]</td>
<td></td>
</tr>
</tbody>
</table>
Computational approaches have garnered a significant amount of attention to date, all requiring some form of validation. The ideal validation is clearly a measure of how successful the technique is in producing clinically efficacious results. This, however, is unfeasible, due to obvious intractability. Currently predictions are often validated in *in-vitro* assays, though this may not always be feasible, or appropriate in certain circumstances. The alternate validation protocol is to measuring the ability of a protocol to reproduce known cases of repositioning, or their overlap with known drug indications. One issue that may occur with such a validation is various sources of bias since it is difficult to ascertain whether the data from which a prediction has been made has been created due to the discovery of that particular case. Modern approaches using heterogenous graph data often remove connections from a knowledge database and attempt to re-construct these using machine learning based techniques. For now, at least, indirect validation seems like the most viable source and robust validation remains an open problem.

### 1.1.2 Molecular Similarity

The concept of molecular similarity is fundamental to modern drug discovery and a key concept in chemical informatics. Comparing molecules and their properties is a fundamental part of pharmaceutical research and can answer commonly asked questions such as, is a hit-compound too similar to a competitor’s intellectual property position? Or how can we complement our screening libraries with further, dissimilar compounds? These questions all require establishing a method for quantifying the similarity between compounds. Since the introduction of molecular similarity analysis, the “similarity-
property principle” (SPP) materialised, based upon the observation that similar molecules are more likely to share similar properties than dissimilar ones²⁹.

Predominantly similarity analysis and the application of the SPP is used by pharmaceutical companies when estimating properties such as adsorption, distribution, metabolism, excretion and toxicity (ADME/Tox) in virtual screening and the prediction of physiochemical properties such as solubility, partitioning or, most commonly, bioactivity⁶⁰. Vast increases in computation power, technological advances in high-throughput screening and synthesis combined with the desire from pharmaceutical companies to reduce high failure rates and discover novel, patentable lead compounds has seen rapid adoption and development of molecular similarity-based applications in the hope of fast and reliable selection of lead candidates.

Despite widespread adoption, the assessment of molecular similarity remains a complicated problem. While it is true that retrieving compounds from a database sharing a common substructure is trivial and unambiguous, defining a quantitative measure of similarity is far from trivial due to subjectivity and context-dependence²⁹. For example, an orange is similar to a banana in that they are both fruits; however, they are dissimilar from the perspective of colouration. Definitively, the representation of a compound, the weighting of features within a representation and the function (coefficient) used to calculate a quantitative measure of similarity will determine a model’s prediction, just as much, if not more, than the way the model processes this information to formulate a prediction.
Based on this information, it is clear that for similarity calculations to have utility, they must be assessed in a formally consistent manner.

1.1.2.1 Molecular Representation

Although a cognitive and a computational assessment of molecular similarity might be fundamentally different, both require some form of symbolic representation of structural information. For example, a medicinal chemist may make a judgement based on an observation of two-dimensional representations of chemical structure, further influenced by their additional knowledge and experience (chemical intuition)\textsuperscript{59}.

Although there may be differences in how humans and \textit{in-silico} systems perceive information, any type of structural representation must include distinctive features or properties for underlying patterns to be identified. With estimations of the size of “drug-like” chemical space being as large as $10^{60}$ molecules\textsuperscript{61,62}, and with data complexity scaling with the number of possible patterns, it is evident that chemical space fast becomes intractable for human perception. To this end, numerous algorithms for computing molecular representations have been developed, along with coefficients for evaluating the similarity between these representations\textsuperscript{63}. Choosing the correct algorithm is context-dependent since no single canonical or invariant molecular representation exists. A good descriptor for a particular application will separate a pair of molecules in metric space with a distance related to the property of interest. Since different structural features may be more
or less important in different contexts, the quality of any model is dependent on the representation selected.

Due to many potential applications, a vast quantity of molecular representations and similarity coefficients exist, ranging from superposition-based algorithms, one-dimensional property-based comparisons, and two-dimensional descriptions of molecular topology to complex rotationally invariant three-dimensional descriptors. Due to the sheer amount of methods, it would be impossible to discuss them all here, hence the reader is directed to more comprehensive reviews\textsuperscript{59,60}. It is imperative to state that while the choice of an algorithm may be context-dependent, most representations are constant and context-independent, hence they can be applied across multiple applications.

The molecular representation space can be broken down broadly into constitution, configuration, and conformation descriptions\textsuperscript{60}. Constitution describes a particular arrangement of bonds and atoms in space with representations constructed to describe topology, the existence (or lack thereof) of distinct chemical fragments and other expressions of molecular features in two dimensions. Since molecules are inherently three-dimensional, representations of constitution fail to differentiate between conformations of the same molecule; thus, descriptions of configuration represent the spatial arrangement of atoms or properties in the form of distinct points, angles or ‘fields’ in three dimensions. Accounting for all conformations and orientations of a molecule is much more challenging, thus, comparatively few methods exist for this purpose, with many utilising spectral shape properties\textsuperscript{64}, which are isometry invariant descriptors derived from the
eigenvalues/eigenfunctions of the Laplace-Beltrami operator (a generalisation of the Laplacian to functions derived on a manifold).

1.1.2.2 Superposition

Superposition based methods attempt to map one molecular representation to another. In a two-dimensional context, this often involves finding a correspondence between atoms in a pair of molecular graphs. Maximum common substructure (MCS) search is a well-known algorithm that entails the identification of the largest connected component (MCS) appearing in a pair (or set) of molecules.\textsuperscript{65,66} MCS/correspondence searching for similarity assessment is intuitive since significant correspondences are likely related to activity, plus results are easily visualised due to an explicitly calculated bijection. Despite these advantages, calculating MCS becomes intractable at a large scale since the problem is non-deterministic polynomial-time (NP)-complete.\textsuperscript{65}

Extending superposition to three dimensions necessitates calculating a spatial alignment where a molecule's flexibility may or may not be accounted for. Despite the additional complexity, various 3D alignment methods have been developed for applications, including: virtual screening, target prediction and scaffold-hopping, the process of replacing a molecule's core structure, or “scaffold” (Section 5.1.1), while retaining similar pharmacological properties to the reference molecule. Alignment-based methods are often successful due to the fact that shape is a fundamental aspect of molecular recognition.\textsuperscript{67} While it is possible to construct a 3D superposition from a least-squares alignment, given a bijective map of atom correspondences, algorithms tend to utilise
alternative molecular representations to improve efficiency and negate the requirement for rigid correspondence criteria. The majority of recent developments decompose the all-atom molecular representation into either: shape, pharmacophore (labelled feature points) or interaction field-based representations. The most successful of these techniques use a Gaussian-sphere model\(^{68,69}\) where the volume of a molecule is represented as a set of overlapping Gaussian spheres centered on atoms, pharmacophores\(^{70}\) or at field-extrema. The volume overlap between a pair of molecular representations can be optimized to produce an alignment without the requirement for bijective correspondences.

Traditionally, superposition methods have been limited due to computational complexity. However, with the significant increase in computer power, massively parallel computing and the use of graphics processing units (GPUs), shape-based screening can be performed on a vast scale with the ability to process millions of molecules in a matter of seconds\(^{71,72}\).

1.1.2.3 Molecular/Chemical Descriptors

Molecular descriptors are the most popular way to represent molecules for similarity computation\(^{63,73}\). A descriptor can be considered a point in a molecular ‘latent space’, whereby molecules that resemble each other more closely will be positioned closer together within this space. Compared to superposition-based similarity, computation is vastly more efficient since descriptors may be pre-computed, and pairwise similarities/distances, calculated using simple coefficients. This efficiency, however, comes at the cost of losing
explicit equivalence relationships between molecules, such as atom-atom correspondences. Descriptors most commonly represent descriptions of constitution (topology), calculated directly from a molecular graph, although some algorithms also include 3D configuration information in a way that is invariant to rigid transformations. Accounting for flexibility in an invariant manner is much more complex, and a relatively small amount of progress has been made in this regard.

The simplest descriptors represent global properties of a molecule, such as: molecular weight, logP, number of hydrogen bond donors etc. Such representations can be modelled numerically using a variety of statistical models. One may argue that similarity based on this descriptor type represents chemical similarity rather than molecular similarity, defined using structural properties. Despite the simplicity, these descriptors are sufficient in many cases, and hence a modeller must utilise domain knowledge when deciding which descriptors are required. For example, logP (experimentally or computationally derived) and molecular weight are likely sufficient when modelling membrane permeability, whereas structural information is critical when considering bioactivity prediction. More complex global properties may also be derived considering topology or 3D shape, including: BCUT parameters and the Kier flexibility index. Given a collection of descriptors, the similarity may be represented by an inverse function of the distance between them.

A further subclass of descriptors represents molecules using binary indices (presence/absence) or counts, representing structural features of a molecule. Structural
features commonly account for: fragments/substructures or pharmacophore points but can represent any structural feature. Structural features may be predefined, as in the case of MACCS (Molecular ACCess System) keys\textsuperscript{79}, binary bit strings in which each bit encodes the presence (1) or absence (0) of a pre-defined structural feature, or generated on the fly using atomic neighbourhoods as in the case of extended connectivity fingerprints (ECFP)\textsuperscript{74}. The similarity between a pair of fingerprints is simply defined as the number of common elements normalised by the total number of elements in each. Extending this concept to three dimensions may involve binning angles and distances between atoms, pharmacophore points or triplets of either\textsuperscript{80}. When constructing a binning scheme, one should consider ‘fuzziness’ so that a descriptor may retrieve most molecules with the same activity without too many false-positive results. Due to the sheer number of fingerprint descriptors often, QSAR studies will utilise multiple fingerprints during the optimisation of a model, retaining the best performing fingerprint or constructing an ensemble of models each considering a different fingerprint type.

The molecular surface is another way to represent a molecule considering three dimensions. Calculating similarity based on molecular surfaces is a good way to reveal similarities in physical and biological properties\textsuperscript{67}. Surfaces are defined in multiple ways, frequently making approximations since quantum definitions of the molecular surface is not practical in this context. A standard approximation considers a molecule as a set of hard spheres, centred at each atom, with a radius equal to the corresponding van der Waals radius. A solvent-accessible (SAS)\textsuperscript{81}, solvent-excluded (SES)\textsuperscript{82}, or van der Waals surface can be computed from this representation, with the surface represented by an occupancy
grid, a cloud of evenly distributed points or a triangulated polygonal mesh. Surface representations are rarely compared directly, with descriptors being calculated from the surface for comparison\textsuperscript{67,83}. A further advantage of surface-based methods is that they are easy to extend for computing the shape complementarity between a binding site and a ligand molecule\textsuperscript{84}. Despite advantages, surface-based methods are still in their infancy due to implementation complexity and the difficulty in describing local rather than just global similarity.

1.1.2.4 Similarity Coefficients

Molecular representation forms only the first half of similarity computation. The Second half, involved in producing a numerical readout, is equally important and can significantly impact the interpretation of similarity. Any function that rationalises similarity from a comparison must be consistent, increase consensus and be superior to an objective measurement\textsuperscript{58}. The choice of a coefficient is dependent mainly on the properties of the molecular representation; in many cases, the inverse of a standard distance function (Euclidean, Manhattan, cosine etc.) is sufficient. However, the disadvantage of distance metrics in terms of human perception is that the readout does not lie within a standardised range. This disadvantage is likely why the Tanimoto coefficient (Tc) has become the most popular metric when assessing the similarity of molecules\textsuperscript{85}. The Tanimoto coefficient not only returns output in a standard range ($0 < Tc(A,B) < 1$) but is also simple to implement and very efficient. It is commonly associated with molecular fingerprints but has also been adapted to other domains such as shape similarity. The Tanimoto coefficient is typically defined as:
\[ T_{c}(A, B) = \frac{c}{a + b - c} \]

Where \( a \) and \( b \) represent the number of (unique) features present in molecules \( A \) and \( B \), respectively, and \( c \) is the number of features shared by both molecules. This can intuitively be interpreted as the ratio of the number of features common to both \( A \) and \( B \) to the total number of unique features (intersection over union). Consider that as \( A \) and \( B \) become more similar, they possess a larger number of shared features, approaching the total number of features, \( c \to a, b \). Correspondingly the number of unique features will approach zero since at the limit \( A \) and \( B \) are equivalent i.e. \( a = b = c \) resulting in \( T_{c}(A, B) = 1 \). Conversely as \( A \) and \( B \) become less similar, the number of shared features approaches zero, and thus at the limit \( T_{c}(A, B) = 0 \). Recently the Tversky coefficient (Tv) has been growing in popularity, it is defined by:

\[ Tv_{\alpha, \beta}(A, B) = \frac{c}{\alpha(a - c) + \beta(b - c) + c} \]

Notice that the coefficient is almost identical to Tanimoto, other than the parameters \( \alpha > 0 \) and \( \beta > 0 \) that weight the number of unique features in molecules \( A \) and \( B \). The values of these parameters typically lie in the unit interval \([0,1]\) of the real line. This weighting scheme introduces asymmetry into the similarity computation, such that when \( \alpha > \beta \)
unique features of the reference molecule A are weighted more heavily than unique features in the query molecule B and vice-versa. Weights may be tuned to analyse the relative importance of common and unique features. While the coefficients mentioned above represent the most common metrics for similarity computation in chemical informatics, many more exist. Clearly care must be taken when using coefficients. An understanding of how coefficients relate to each other is imperative, especially in the case of asymmetry where relationships are non-monotonic.

1.1.3 Application of Similarity Principles to Biological Molecules

Molecular similarity and the similarity property principle (SPP) is fundamental to chemical informatics, but do these principles translate into the biological domain? Protein comparison represents an elementary operation in biological informatics, whether at the sequence or structural level, and is used to model evolutionary relationships, classify structures for functional inference, and discover templates for homology modelling. Indeed, one can draw many parallels between small molecule and protein similarity, particularly when considering structural comparisons. The biggest difference however, is that protein similarity is typically considered in the context of evolution, while small molecules completely escape evolutionary study. A further complication is that proteins are composed of one or more evolutionary units known as domains. The combination and interaction of these domains in a protein determine its structure and function, thus complicating sequence-structure-function relationships.
1.1.3.1 Protein Homology

Protein homology refers to the evolutionary relationship between either a pair or set of proteins that share a common ancestor. Homology is identified by observing sequence or structural similarities that are greater than what would be expected by chance, indicating evolutionary divergence. This concept is particularly important given that homologous proteins often share similar structure and function, and thus existing annotations can be utilised to predict the properties of uncharacterised proteins. Hierarchical classification of proteins into homologous groups called families, has become a particularly popular resource for this purpose. Aside from functional inference, homology also has practical applications in drug discovery including target identification, off-target identification, and selectivity analysis.

Given that a protein's primary structure, is the primary driver of structure and function, sequence alignments are commonly used as a quantitative measure of homology. These algorithms identify continuous regions of similar characters between two sequences using dynamic programming. Alignment methods fall broadly into two categories: global and local. Global alignment algorithms e.g. Needleman-Wunsch$^{89}$ perform an end-to-end alignment, aligning each residue, and are thus commonly used when the length of sequences that are being compared are roughly equivalent. Local alignment methods e.g., Smith-Waterman$^{90}$, do not assume that pairs of sequences are similar over the full length of the sequence, and thus are more useful when searching for more distantly related proteins where similarity may only exist in smaller regions within the global context. Due to the large size of modern sequence databases, algorithms relying on heuristics to accelerate searching have been developed including, BLAST$^{91}$ and FASTA$^{92}$ amongst many others.
The similarity between a pair of sequences is commonly quantified in terms of their percentage sequence identity. As a rule of thumb, a greater sequence identity indicates a more recent common ancestor, and a greater chance of functional similarity. Although a sequence identity threshold of 30% is commonly used to infer homology, in reality this threshold can miss many homologues. While it is true that sequences with high identity adopt similar structures, through 3D structural alignments it has been shown that the majority of structurally homologous proteins share an identity of less than 12%. This marks a potential issue with using sequence alignment to detect remote homology. BLAST and FASTA can confidently identify homology in sequences with >40% identity, however, they frequently fail to produce accurate alignments between pairs with approximately 20-35% identity, the so called “twilight-zone”.

Detecting remote homology is particularly important in applications including, structure and function prediction. As a result, methods utilising profile hidden Markov Models (HMMs), have been developed (e.g. HMMER, SAM), offering a probabilistic assessment of distantly related sequences. These methods particularly sensitive, with the ability to identify homologues within the “twilight” and even “midnight” zones (8-10% identity). The power of these methods comes from the fact that they consider profiles of protein families which can better model residue insertions and deletions.

Despite a wealth of excellent sequence-based resources, structure is considerably more conserved than sequence, and thus only direct structural comparison can identify very remote evolutionary relationships. The 3D comparison/alignment of protein structures is now a standard operation, based on the superposition of atoms, residues or feature points. Due to the larger size of proteins compared to small molecules, using traditional 3D
alignment techniques to screen large databases can be computationally demanding. To mitigate the shortcomings of these types of alignments, several methods have been developed that utilise shape-based similarity algorithms to detect global or local similarities between protein structures. The classification of these algorithms is broadly identical to those developed for small molecules, including: Gaussian overlap, surface, and harmonics-based approaches\(^67\). Many of these protein similarity methods have been directly adapted from those developed for small molecules and vice-versa. The use of such methods has enabled the detection of very distant homologues and the accurate identification of domain boundaries in structures containing multiple domains\(^99\). Due to the relative lack of protein structural data however, the integration of both structure and sequence data is important to improve our understanding of evolution and function.

1.1.3.2 Sequence-Based Classification

The vast wealth of knowledge accumulated about sequence, structure and function has necessitated the construction of hierarchical classification schemes in order to aid researchers in the understanding evolutionary relationships, protein function and molecular interactions. Classifications may also be used to annotate uncharacterised proteins where annotations can be transferred based on the group the query is assigned to. Such schemes commonly classify homologues proteins into superfamilies, whereby these groupings are subdivided into smaller more closely related clusters known as families. The criteria for subdivision is context dependent but is often related to sequence or functional similarity.
Due to the broad utility of such classification schemes, there exists a multitude of protein databases using various classification schemes. A large number of these databases classify proteins into families and domains based on protein “signatures” which represent locally conserved sequence patterns, built from multiple sequence alignments. These signatures allow models to factor in the degree of conservation of each residue position, which is crucial as residues important for function or structure tend to be more conserved within a protein family. As protein signatures can represent conserved regions even when sequence identity is low, they are more efficient in identifying distant homologues, providing a kind of “representation” of a protein family, domain or motif that can be linked to function or structure.

The difference between most sequence-based resources is in how these signatures are computed. Signatures can be computed with a variety of algorithms including regular expressions\textsuperscript{100}, profile matching\textsuperscript{101} and HMMs\textsuperscript{102}. The Pfam database\textsuperscript{103} containing, families, domains and functional units, is possibly the most comprehensive and well-known database, containing greater than 19,000 domain families. Each family is represented by a profile and two HMMs and is manually annotated with functional information, domain architecture and links to other databases. Other sequence-based databases include PROSITE\textsuperscript{100}, PRINTS\textsuperscript{104} and SMART\textsuperscript{105}. A comprehensive review of available resources can be found elsewhere\textsuperscript{106}. An integrated resource, InterPro\textsuperscript{107}, contains data from multiple data-sources, with the aim to cover a greater portion of the available sequence and taxonomic space than could be covered by a single database alone, minimising redundancy by collapsing overlapping entries into a single data point.
1.1.3.3 Structure-Based Classification

The development of structure-based protein classification methods was driven by the fact that structure is considerably more conserved than sequence\textsuperscript{98}. Direct structural comparison can reveal more distant homologues, painting a clearer picture of how evolution has utilised combinations of existing domains to develop novel phenotypes\textsuperscript{108}. Several methods have been proposed classifying proteins into the levels: “class”, “folds”, “super-families” and “families”, independent of function. The most comprehensive of these resources are the Structural Classification of Proteins (SCOP)\textsuperscript{109} and CATH (Class, Architecture, Topography and Homology)\textsuperscript{110}.

The two databases share fundamental similarities but differ in the way they define domains and how they are maintained. SCOP primarily relies on manual curation, whereas CATH utilises a semi-automatic approach that involves automated pipelines complemented by manual validation. Irrespective of their differences, both methods first separate proteins into discrete globular domains, prior to classification since these individual units can have distinct functional roles. Separated domains are classified hierarchically, with “class” forming the top level which is usually determined by secondary structure composition. There are three broad classes: mainly $\alpha$ proteins, mainly $\beta$ proteins, and mixed $\alpha$ - $\beta$ proteins. Within these top-level classes proteins are divided into “folds” determined by the topology of secondary structure elements (connectivity and arrangement), and then into “superfamilies” which contain protein domains sharing similar function. Superfamilies often lie at a level where homology cannot be inferred by sequence identity and are thus further separated into “families” comprising groups of proteins with similar function and similar sequence. Additions to SCOP and CATH, Superfamily\textsuperscript{111} and
Gene3D\textsuperscript{112,113} respectively, extend the structural domain to sequence databases through the generation of HMMs to represent each superfamily. By searching various sequence resources with the HMM models, domain assignments and functional annotations can be provided for proteins with unknown structures. As with the sequence-based classification methods, both CATH and SCOP are integrated into the InterPro database\textsuperscript{107}.

1.1.3.4 Functional Classification

The relationship between sequence, structure and function is far from trivial, especially given that function is context-dependent and can be viewed from multiple perspectives. Function can be defined in terms of: the molecular function of the protein, participation in particular biochemical pathways and the cellular location where the protein performs its function. Many language-based annotations in protein function databases are often vague and unspecific; hence several resources have been developed for functional annotation. The most popular existing resources include both the Enzyme Commission (EC) classification system\textsuperscript{114} and the Gene Ontology (GO)\textsuperscript{115} which provide hierarchical classifications of protein function (also known as gene function).

The EC system is one of the earliest classification systems consisting of a four-level hierarchical numerical classification, where each level represents the catalytic reaction that it performs. The top level digit represents the enzymes class, based on the reaction it catalyses, while the following digits provide more specific detail about substrate and product specificity. Although well established, the EC system is fundamentally limited
in that it cannot classify non-enzymatic proteins or describe proteins with multiple roles and the various interactions it partakes in.

The GO system is a more comprehensive scheme, assigning functional annotations to both enzymes and non-enzymes. These annotations are derived from three non-overlapping ontologies: the Molecular Function Ontology (MFO) which describes the biochemical activity of the protein, the Biological Process Ontology (BPO) which describes the proteins involvement in biochemical pathways and cellular processes and the Cellular Component ontology (CCO) which described the cellular location within which the proteins function is performed. Each ontology is organised into a directed acyclic graph (DAG), where traversal towards the root, reveals more general annotations. Upon assignment of a particular GO term to a protein or gene, due to the hierarchical nature of the ontology the protein implicitly inherits the terms ancestors. This ontology structure is powerful in that it enables a quantitative measure of functional similarity using content-based and graph-based methods\textsuperscript{116,117}.

Despite the widespread use of both the EC and GO methods for functional annotation, the exponential growth in the amount of sequence data collected means that very few have been experimentally characterised and thus many are not annotated. This lack of annotation has necessitated the development of automated functional annotation methods. A biologically relevant clustering should be able to group proteins with the same function where homology can also be demonstrated. In structure-based classifications like SCOP and CATH, while proteins in a superfamily share a core structural unit, the largest of these superfamilies cover most known proteins\textsuperscript{118}, meaning that only a subset of the superfamily members share the same function. These functional deviations occur as a result
of evolution, through mutations in individual domains, where even a single residue change can result in a functional change, potentially by removing or obstructing an important residue in the active site\textsuperscript{119}. Larger structural embellishments can also alter the geometry, electrostatics and surface properties of the active site, resulting in a change of substrate binding specificity\textsuperscript{119}. Clustering these superfamilies into further functional groups, with automated methods, can aid the prediction of protein function and aid in the understanding of the evolution of function.

To address this need, the developers of CATH introduced an algorithm for functional classification of homologous superfamilies (FunFHMMer) into functional families or “FunFams”\textsuperscript{120}. The generated functional families, can be used for downstream annotation of protein function and can also predict functionally important residues within active sites\textsuperscript{121}. FunFams were validated for functional purity by the authors\textsuperscript{120}, using known functional information (EC numbers) against other domain-based approaches including Pfam\textsuperscript{103}. The groupings generated by FunFHMMer were shown to be more functionally coherent indicating its ability to predict protein function and structure for uncharacterised proteins.

1.1.3.5 Binding Site Similarity

One of the most exciting applications of protein similarity computation, especially in drug discovery, involves its application to binding-site comparison. Comparison of local structural regions is essential for this application since global sequence and structural alignments will often fail to retrieve similar sites across different protein folds. The
The conservation level of binding site structure and chemistry is significantly increased relative to the protein as a global entity, therefore, consistent and accurate local similarity calculations have the potential to predict protein function\textsuperscript{122}, ligand polypharmacology\textsuperscript{34} and suggest candidates for drug repurposing\textsuperscript{123}. Objectively, it is clear that the SPP can also be applied based on the assumption that similar local regions of protein structure are likely to share similar properties, where function or binding of a common ligand are most frequently the properties of interest. A more comprehensive review of binding site similarity and the tools available is presented in Section 2.1.

The pairwise similarity between a molecule and a binding site can be considered a further extension of both domains, where similarity is considered in terms of complementarity, i.e. a perfect complementarity between two sampled values within the range, is achieved when both values lie at either end of the range and perfect anti-complementarity is achieved when both values are equal. Complementarity calculations have been applied successfully in virtual screening experiments, outperforming docking experiments in some cases\textsuperscript{124}, and can be used for visualisation during structure-based drug design, indicating areas of a molecule that may be optimised for increased activity\textsuperscript{84}.

1.2 Aims

This project aims to design and construct a web-based platform/graphical user interface (GUI) based on the use of molecular similarity searches of the Protein Data Bank (PDB)\textsuperscript{47} to provide a user-friendly way to interact with drug-protein relationship data. The
platform will aid users in identifying: off-target interactions, distantly related proteins exhibiting similar binding characteristics, potential drug repurposing candidates, and help understand structure-activity relationships. Although many tools exist for similarity searching within ligand and protein databases, as far as we are aware, no tool currently exists for the interactive exploration of drug-protein relationship data from the context of off-target and drug repurposing candidate identification. The most similar, tool from which we take inspiration, is STITCH\textsuperscript{53} which enables the exploration of protein-chemical relationships with the intention to aid in the understanding of function. This project aims to explore relationships from a structural perspective, identifying similarities which may not be located by simply considering existing interactions alone. The main aims for achieving this objective are outlined below:

- Construct a database containing annotated protein, ligand and interaction data
- Develop similarity searching methods from both small molecule and protein input
- Develop methods to build a relationship network representation of data returned from similarity searches
- Annotate the constructed networks with valuable information to a user

The presented application should be considered a proof of concept, whereby the identified shortcomings motivate the research direction for the rest of this thesis. We expect that the work presented in the latter chapters (Chapters 2, 3 and 4) on binding site similarity could be integrated into the application for a more powerful interpretation of local structural relationships, especially from the context of shared ligand binding.
1.3 Methodology

To achieve the aims outlined (Section 1.2), a basic infrastructure and pipeline were created (Figure 1.2). The pipeline shows the control flow of the backend similarity searching program for both a ligand-based query and a protein-based query. Fundamentally, the structure of both pipelines is identical, although different methods, i.e. similarity searching, clustering and network representations, are utilised for small molecule and protein inputs. The basic program structure is relatively simplistic, starting with the input of a small molecule or protein sequence/structure, the program performs a similarity search of entities within the protein data bank (PDB). Similar entities are annotated with protein-ligand interaction data. The resulting data is clustered and used to construct a dynamic/interactive network representation, where nodes represent either a cluster of ligands or proteins and edges represent an interaction between protein and ligand. The clustering scheme used must be simple and facilitate the construction of a simple visualisation. This process will be wrapped with a user-friendly web-based interface, enabling ease of use, presenting data useful for multiple tasks within the drug discovery pipeline, integrating information from multiple sources clearly and concisely. In this section, the techniques used are presented and discussed, while the results and discussion section (Section 1.5) explains current implementation, the problems they have and how these may inevitably be improved.
Figure 1. 2 An illustration of the master pipeline for the LigNFam project. Some modules differ based on the type of input received, including the similarity search, clustering and network building.
1.3.1 Database Construction

The similarity searching program requires three databases: a ligand database, a protein database, and a protein-ligand interaction database (Figure 1.2). The contents of these three databases and the data sources and tools used for construction are presented in this section.

1.3.1.1 Ligand database

A database was constructed to enable similarity searching of ligands within the PDB, containing all ‘drug-like’ ligands represented by their three-letter/digit HET ID. The term drug-likeness refers to a qualitative concept defining how ‘drug-like’ a molecule is in terms of factors such as bioavailability. Historically, drug-likeness was determined using Lipinski’s rule of five, developed to optimise compounds for oral bioavailability. Since these rules are very restrictive by today’s standards, an extended set of drug-likeness rules were used for compound filtering (Table 1.2). In addition to basic ligand information, ligand descriptors are also included in the database consisting of: SMILES, molecular weight, number of atoms, number of nitrogen and oxygen atoms, number of hydrogen bond donors, number of hydrogen bond acceptors, component elements and a Morgan (ECFP) topological fingerprint with a radius of 2 (Section 1.4.1.1.1). The data used to construct the database was obtained from the RCSB PDB Ligand Expo with the construction performed using Python 3.6 utilising the library Pandas. Molecular descriptors were calculated using the cheminformatics library RDKit. Additionally, structure data files (SDFs) are extracted for all ligands to enable 3D searching and 2D and allow user downloads of structural data.
### Table 1

<table>
<thead>
<tr>
<th>Molecular Property</th>
<th>Lipinski</th>
<th>Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Molecular weight</td>
<td>&lt; 500 Da</td>
<td>100 &lt; Da &lt; 800</td>
</tr>
<tr>
<td>Hydrogen bond acceptors</td>
<td>&lt; 5</td>
<td>&lt; 16</td>
</tr>
<tr>
<td>Hydrogen bond donors</td>
<td>&lt; 5</td>
<td>&lt; 8</td>
</tr>
<tr>
<td>Rotatable bonds</td>
<td>&lt; 10</td>
<td>&lt; 16</td>
</tr>
</tbody>
</table>

Table 1. 2 Lipinski\(^{62}\) and extended compound filtering rules\(^{125}\) applied to retain only ‘drug like’ molecules in the constructed database.

### 1.3.1.2 BLAST database

As a proof-of-principle, a BLAST\(^{91}\) based sequence search is implemented to enable protein searching. While a sequence-based search method will allow one to probe the relationship between sequence and ligand similarity, we acknowledge that due to the effects of convergent evolution, often similar binding sites can exist in globally unrelated protein structures\(^{129}\), i.e. low protein fold and sequence homology. The quantification of local similarities is thus preferred for objectives including off-target prediction and identifying repurposing candidates. However, the quantification of local similarity is much harder to compute due to inherent subjectivity. The global similarity is thus used as a proof-of-principle, and binding site comparison methods for local similarity quantification are presented later in this thesis (Section 2) The BLAST database was constructed from protein sequences within the PDB. A deeper explanation of BLAST as a search tool is presented
in Section 1.3.3.1. Protein sequence information was downloaded in the FASTA format from the RCSB PDB\textsuperscript{47}.

### 1.3.1.3 Protein-ligand interaction database

A protein-ligand interaction database containing information regarding protein-ligand complexes within the PDB was constructed using data obtained from PDBSUM\textsuperscript{130}. Interactions were annotated with data, including residues involved in the protein-ligand interaction, calculated using files produced by the program LIGPLOT\textsuperscript{131} and Pfam domain(s)\textsuperscript{103} assigned to each complex, aiding in the understanding and grouping of proteins (Section 1.3.3.2). Pfam domains represent functional regions of a protein, with the presence of various domains characterising a protein's function. The Pfam database stores these conserved units, represented by a set of aligned sequences with their probabilistic representation (a profile hidden Markov model). The functional classification makes understanding evolutionary and functional relationships between proteins significantly easier. The use of human-readable names also lends itself well to our purpose, as users must be able to interact with data intuitively. Construction of the database was achieved using Python 3.6 and the Pandas library.
1.3.2 Ligand-based search

The ligand-based search is conducted under the assumptions made by the SPP (Section 1.1.2) that molecules sharing similar structural features are more likely to share similar properties. In this case, we are interested in shared binding or similar activity profiles, where we can make an overarching statement: given two molecules $A, B$ and a protein target $P$, if $A$ interacts with $P$ and $B$ is similar to $A$, then one can formulate a hypothesis that $B$ may have the potential to also interact with $P$. Of course, this statement will not always be correct, considering that the structural activity landscapes are non-linear in nature. However, considering multiple molecules and proteins allows one to make judgments more confidently since now one considers a potentially complex set of relationships. The network visualisation approach allows a user to more easily locate such relationships within data that may otherwise be unnoticed. Identification and further analysis of relationships may allow a user to identify off-targets and repositioning opportunities. The interface accepts SMILES strings as input since it is a widely used, machine-readable and compact molecular representation. This makes it ideal for storing in structured databases, where a graph structure can be constructed on demand for any given molecule.

1.3.2.1 Similarity searching

Initially, the interface was designed for use with molecular fingerprints, an efficient and rapid method for encoding and comparing molecular structures \textit{in-silico} (Section 1.1.2).
Later the ability to perform 3D searches was added using alignments of pharmacophoric representations. Using pharmacophore-based representations is interesting for this task since scoring similarities has less dependence on similar molecular scaffolds due to a more abstract molecular representation. Instead of locating similar structural elements, the approach focuses on identifying common chemical features with the ability to facilitate molecular interaction. Both procedures are outlined in Section 1.3.1.1.1 and Section 1.3.1.1.2.

1.3.2.1.1 2D Similarity Searching

Molecular fingerprints, most commonly, encode a molecular structure into a sparse binary vector encoding the presence or absence of unique chemical substructures. These vectors can be compared rapidly using metrics such as the Tanimoto coefficient (Section 1.1.2.4). Due to the inherent subjectivity of the concept of similarity, it is unsurprising that multiple types of fingerprints have been proposed. In this project, the Morgan, or extended connectivity fingerprint (ECFP)\textsuperscript{74}, is used due to its popularity in the community and robust implementation in the RDKit package\textsuperscript{132}. This particular implementation uses atom invariants and connectivity information. A Morgan fingerprint is calculated directly from a molecular graph, systematically recording the neighbourhood of each heteroatom in circular layers up to a specified radius. The recorded neighbourhoods represent atom-centered sub-structural features, which are subsequently mapped to integer code identifiers using a hashing algorithm. Each identifier serves as an index of a binary integer set to one in a large virtual bit vector. These fingerprints may also be stored as a fixed-length bit-
string by ‘folding’ the fingerprint to a smaller density. A folded fingerprint requires less
storage space while simplifying similarity calculations. However, the folding process may
introduce bit collisions, where a certain substructure element is represented by the same
bit, resulting in a loss of information. Fingerprinting was implemented using the RDKit,
with pairwise similarities calculated using the Tanimoto coefficient. The use of Tanimoto
over a parametric coefficient such as Tversky (Section 1.1.2.4) simplifies the user interface
and software backend.

1.3.2.1.2 3D similarity searching

3D similarity searching is implemented using a pharmacophore alignment method.
Pharmacophores represent locations of a molecule that can facilitate molecular interactions
frequently observed in protein-ligand complexes. Interactions commonly considered
include hydrogen bonding, steric effects and lipophilic and electrostatic interactions\textsuperscript{133}. Once defined, pharmacophores may be superposed, constructing an optimal alignment
between corresponding features. Just as with fingerprints, there have been numerous ways
proposed for the definition of pharmacophore points and also for the alignment procedure.
We use the open-source software Align-it (formerly Pharao)\textsuperscript{70} for this project, which uses
a Gaussian volume-based representation of pharmacophore points, enabling a
differentiable overlap optimisation protocol using gradient ascent. Align-it computes nine
different pharmacophore types displayed in Table 1.3. The calculation process for each of
these types can be found in the associated publication\textsuperscript{70}. Some features are also associated
with a normal direction, which impacts the scoring of alignments.
<table>
<thead>
<tr>
<th>Pharmacophore type</th>
<th>Associated Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aromatic ring</td>
<td>Yes</td>
</tr>
<tr>
<td>Hydrogen bond donor</td>
<td>Yes</td>
</tr>
<tr>
<td>Hydrogen bond acceptor</td>
<td>Yes</td>
</tr>
<tr>
<td>Lipophilic region</td>
<td>No</td>
</tr>
<tr>
<td>Positive charge center</td>
<td>No</td>
</tr>
<tr>
<td>Negative charge center</td>
<td>No</td>
</tr>
<tr>
<td>Hybrid donor/acceptor</td>
<td>Yes</td>
</tr>
<tr>
<td>Hybrid aromatic/lipophilic</td>
<td>Yes</td>
</tr>
<tr>
<td>Exclusion sphere</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1. A list of the pharmacophore feature types computed by the Align-it (Pharao) software. Some are also associated with a normal direction that impacts the scoring of alignments, i.e. correspondences with a similar normal direction score higher than those with divergent normal directions.

The further benefit of the smooth Gaussian representation is that for a given alignment, the Tanimoto coefficient can be used as a similarity metric where pairwise and self-volume overlaps are used in the place of counting common and unique indexes. The method can thus be dropped into the original pipeline with minimal modification to the clustering and network-based representation. The final interesting feature is the use of exclusion spheres, which are extracted from a receptor rather than being extracted from the ligand. These points have a different function during alignment, where any pharmacophoric query point overlapping with an exclusion sphere is penalised, preventing an alignment in which a ligand cannot fit within the confines of a target receptor.
For implementation, the original fork of Align-it was modified to meet our requirements, being compiled against RDKit (C++) as opposed to the Open Babel toolkit\textsuperscript{134}, with the addition of Python wrappers to enable a simple drop into the existing backend. The wrappers also add a utility for automating exclusion sphere generation, which was previously done manually for each query, and a simple visualisation tool. The updated code is publicly available on GitHub\textsuperscript{135} along with simple usage guidelines.

Since the molecular input form accepts SMILES, a one-dimensional representation, a 3D conformation must be generated for alignment-based similarity searching. In practice, ten query conformations are computed using the ETKDG method\textsuperscript{136} implemented in RDKit, to account for ligand flexibility due to the rigid alignment technique. The number of conformations generated represents a trade-off between robustness and efficiency. The final similarity score is the maximum result calculated across the ten alignments.

1.3.2.2 Clustering

Following a similarity search, ligands above a similarity threshold ($T_c \geq 0.5$) are clustered to create ligand nodes within a network, while proteins are clustered to create protein nodes. We find that thresholds lower than 0.5 introduces too much noise and is not conducive to visualisation. While the general consensus is that Tanimoto scores $T_c > 0.5$ indicates a similar molecule and that $T_c > 0.8$ indicates high similarity\textsuperscript{137}, this is indeed a fallacy considering that the context-dependent nature of a similarity is realistically precludes the use of meaningful discrete thresholds. Therefore, we refrain from claiming that this
threshold is chosen as a biologically meaningful threshold and instead regard the threshold as arbitrary for the sake of clean visualisation. The clustering implemented is simply defined by a rudimentary bucketing of molecules into four groups depending on their similarity to the query ($Tc \geq 0.8, 0.8 > Tc \geq 0.7, 0.7 > Tc \geq 0.6, 0.6 > Tc \geq 0.5$) and proteins are grouped into clusters depending on the Pfam annotation of functional domain interacting with the ligand. Despite the rudimentary nature of the clustering, we find that the results are more visually appealing when compared to other clustering techniques such as sphere-exclusion (Butina$^{138}$), which create numerous singleton molecules. We also find that with unsupervised clustering techniques, users find it difficult to understand the relationship of each cluster with the query molecule. The simple bucketing scheme retains a somewhat meaningful relevance to the original query. Although we emphasise visual appeal, we acknowledge that this should not overshadow the fidelity of the data. Future work could focus on creating a method to visualise clusters in a user-friendly manner. A potential solution could involve clustering with sphere-exclusion followed by using a continuous colour scale to map each cluster to a colour that reflects the clusters’ average similarity with respect to the query molecule. In this way, a user could understand the relationship of each cluster to the query without sacrificing cluster quality.

1.3.3 Protein-Based Search

The principle behind the protein-based search is the reverse of the ligand search in that the molecular similarity principle can be used in much the same way. Highly similar proteins
are likely to descend from a common ancestor sharing the same or similar functions. In a scenario where two proteins are highly similar and a molecule interacts with one of these proteins, we can suggest that the same molecule may also interact with the other protein. If these proteins are involved in different disease indications, this could represent a reposition. Again, in reality, molecular recognition is complex, and the selectivity of molecules towards a particular target is also non-linear, i.e. a change in a molecules stereochemistry can significantly change its activity. However, this is not picked up in many similarity evaluation algorithms. These changes are dependent on the protein target shape and physiochemical properties.

1.3.3.1 Similarity Searching

There are many ways to measure the similarity between proteins, usually categorised into global and local variants. In this project, BLASTP\textsuperscript{91} is implemented as a way to find similar sequences. We chose BLAST in the proof-of-principle application due to its relative popularity among researchers, its ease of configuration and its lower computational expense compared with more rigorous pairwise approaches including FASTA\textsuperscript{92} and SSEARCH (Smith-Waterman\textsuperscript{90} algorithm implemented as part of FASTA). While BLAST and FASTA can confidently identify homologues with >40% sequence identity (in long alignments), they often struggle identifying remote homologies in the so-called “twilight-zone”\textsuperscript{94} (20-30% identity). A powerful alternative that could be considered is HMMER\textsuperscript{95}, which uses profile Hidden Markov Models (HMMs) to represent families of proteins and has the ability to detect distant homologues with higher sensitivity. Historically the
practical use of HMM-based methods has been limited due to the additional complexity, however, the newest iteration HMMER3\textsuperscript{139,140}, uses an acceleration heuristic along with single instruction, multiple data (SIMD) optimisations bringing performance almost in-line with BLAST. In a production environment it may be worth exposing multiple search methods, however, we expect that the local 3D-based methods developed later (Section 2), will be a powerful alternative to sequence-based approaches.

The BLAST algorithm is heuristic, rapidly producing statistically significant local alignments. Put briefly, the algorithm identifies near-perfect matches to words (k-mers) within a query sequence, extending these matches to determine whether words are part of a longer sequence, resulting in a higher statistical relevance. This algorithm approximates the much more rigorous Smith-Waterman algorithm\textsuperscript{90}. The efficiency of the approach makes it ideal for performing interactive searches.

1.3.3.2 Clustering

Following the identification of similar sequences to the query, proteins are clustered into groups for visualisation. Currently, this clustering procedure is a rudimentary grouping into four groups ($I > 70\%, 70\% > I \geq 50\%, 50\% > I \geq 30\%, I < 30\%$) based on sequence identity, where identity is calculated with respect to the shortest sequence. Sequence identity thresholds are particularly difficult to define, especially given that even minor deviations in sequence, can have a large impact on the geometry and properties of an active site\textsuperscript{119}. Despite this sequence identity is an easy metric to understand and valuable to users without a bioinformatics background. We determined that the best way to define groups
would be, based on the likelihood that proteins within the cluster will share a similar function to that of the query. Numerous studies have been conducted, attempting to identify thresholds at which function can be transferred from the perspective of both EC numbers and GO classifications. Wilson et al.\textsuperscript{141}, determined that all four digits in an EC number can be transferred based on a sequence identity of 40\%, by analysing sequences in pairs of structural domains from the SCOP database. Another study conducted by Todd et al.\textsuperscript{141} back up these results through analysis of functional variation within superfamilies in the CATH database, suggesting thresholds for full EC conservation of, 40\% for single domain and 60\% for multi-domain proteins. For conservation of the first three digits, thresholds of 30\% and 40\% identity were suggested respectively. In contrast to these results, it was argued that due to potential biases in the datasets used the defined thresholds could not be trusted\textsuperscript{142} however, using techniques to counter bias Tian et al.\textsuperscript{143} and later Addou et al.\textsuperscript{144} also report that sequence identity above 40\% can be used as a threshold to confidently transfer the first three digits of an EC number while 60\% can be used to transfer the full EC number with 90\% confidence. Interestingly Addou et al.\textsuperscript{144} also report that despite probabilistic measures of homology being more capable at identifying highly divergent sequences, sequence identity provides a better measure of function conservation a result similar to that reported by Wilson et al.\textsuperscript{141}. A further study\textsuperscript{145} analysed the relationship between sequence and functional similarity using three categories of the GO, identifying that at \textasciitilde70\% sequence identity proteins are 90\% likely to share the same molecular function and biological process. Given these reported values, the upper bin can be described as “highly likely to share function”, with the two intermediate bins representing “likely to share similar function up to full EC numbers” and “likely to share similar function up to
three digits of an EC number”. The final bin (the twilight-zone) is included for the remainder of sequences, however, the confidence of assigning similar function and structure within this bin is considerably lower.

Each bin represents protein nodes within the network. To create ligand nodes, clustering is more complex, based upon an all-by-all similarity. Currently, three methods have been implemented: sphere exclusion (Butina)\textsuperscript{138}, scaffold tree\textsuperscript{146} clustering and K-means\textsuperscript{147}. Note that currently, the small molecule clustering only considers 2D similarity computations.

1.3.3.2.1 Sphere Exclusion clustering

Butina is an unsupervised, sphere exclusion clustering algorithm explicitly developed for chemical clustering\textsuperscript{138}. The development focus was to create a methodology for identifying dense homogenous clusters in which intra-cluster similarity reflects a specified Tanimoto cutoff. The method starts by identifying potential cluster centroids, calculating the number of nearest neighbours at the specified cutoff for each molecule in a set. Molecules with more neighbours are prioritised for selection, hence the set is sorted in descending order. In a sphere exclusion process, starting with the first compound (centroid) in the set, molecules are assigned to the centroid if their pairwise similarity is greater than the cutoff. After all possible compounds have been assigned, the process is repeated with the next centroid in the set. Molecules that are not assigned to a cluster become singletons. The downside is that singletons often may be a neighbour of a compound within a cluster,
although they are excluded due to a ‘stronger’ centroid. The method prevents the creation of heterogeneous clusters with low intra-cluster similarity.

1.3.3.2.2 K-means

K-means is an unsupervised centroid-based clustering algorithm. It can be used to cluster molecules using a variety of descriptors. The algorithm is iterative and consists of two repeating stages: cluster assignment and centroid moving. A number of points are randomly initialised to begin the process, representing the desired number of clusters. In the cluster assignment step, the algorithm iterates through each data point and assigns it to the closest centroid. In the centroid moving step, the average of each cluster is calculated, becoming the new centroid. The process is then repeated using new centroids until there is no longer movement. The drawback of this algorithm is that it must be specified in a standard implementation. If no knowledge of the data is known a priori, selecting a value is difficult.

1.3.3.2.3 Scaffold Tree

Scaffold Tree\textsuperscript{146} clustering is a rule-based method that recursively removes peripheral rings from a molecule. A set of chemically intuitive rules determines the ‘most peripheral’ ring. The number of rings in a particular scaffold indicates the hierarchy at which it exists within the resulting tree structure. For forming clusters, molecules containing the same hierarchy two (two ring) scaffolds are grouped. If only one hierarchy exists for a molecule,
this structure is used to form the group. The algorithm seems to be a good fit for this purpose since the method is non-parametric, the number of clusters does not need to be known \textit{a priori}, and the number of singletons is much lower than for sphere exclusion-based methods. Clusters based on shared scaffolds may also benefit from being useful for identifying scaffold hops\textsuperscript{148}. The method is implemented using the ScaffoldGraph software, which is introduced in Section 5.

1.3.3.3 Web Interface

A basic interface was constructed, consisting of an input page, a loading page, a results page and an error page (Figures 1.7, 1.8, Section 1.4.1.4). The input page contains a form for submitting either SMILES or FASTA inputs. The backend of the web interface (Section 1.3.3.4) was written in Python 3.6 using the Django framework\textsuperscript{149}. The front-end was written using the standard web development languages: HTML5, CSS, and JavaScript. The network visualisation was created using the D3.js visualisation library\textsuperscript{150} and jQuery, while the styling of the website elements utilised elements from the bootstrap library. To render molecular structures graphically, a JavaScript library, smiles-drawer, was employed\textsuperscript{151}.

1.3.3.4 Backend

The interfaces backend is implemented with the Django web-framework\textsuperscript{149}. The current structure is relatively simple consisting of: a form handler for input handling, the database searching (similarity) functions and a session handler based around an SQL database
The form handler is capable of accepting SMILES strings and protein sequences, which it validates to prevent invalid program inputs. SMILES strings are validated by using the RDKit to construct molecular graph objects from the input. If the RDKit fails to parse the input, the form handler will catch the error and report this to the front-end interface. Additionally, any molecule with an atom count of less than 8 will be classed as invalid, since these lie outside of the specified ‘drug like’ domain (Table 1.2).

Protein sequences are harder to validate, though the validator will not accept sequences with less than 80 characters or any sequences containing characters not within the standard IUPAC amino-acid alphabet. The JSON output returned by the database, is passed to the session handler, which controls the output in the frontend interface. The session handler is based around the native session handler in Django which uses cookies containing a session ID to access data stored in an SQL database. When a search is successfully submitted, the user is directed to a ‘loading page’, where they will be held until the backend has finished processing the request. Result data is stored in the SQL database with a cookie session identifier as the key. To allow users to submit numerous searches with the same session identifier, each result is given a separate identifier, a randomly generated UUID. When the user is redirected to the results page, the data is retrieved from the database using the session key and the result UUID key which is read from the URL. In this way, separate results are mapped to separate URLs.
Figure 1. 3 The backend logic for the LigNFam web interface. The pipeline is explained in Section 1.3.3.4
1.4 Results and Discussion

The current state of the application is considered to be a proof of principle, whereby the identified shortcomings motivate the research direction for the rest of the thesis. Basic databases with fundamental information have been compiled (Section 1.4.1.1), three similarity searching methods have been developed (Sections 1.4.1.2 and 1.4.1.3), and basic backend and interface have been constructed (Section 1.4.1.4). The following sections describe the current implementation and potential future directions.

In terms of concept, a few problems have been identified with the current implementation. The PDB is the main resource for this project since it contains useful information regarding 3D structures of protein-ligand complexes. However, a fundamental issue that we have discovered is that compared to small molecule resources such as ChEMBL\textsuperscript{49}, the amount of 3D structural data involving complexes containing non-endogenous molecules is particularly limited. A large proportion of drug targets include membrane proteins such as GPCRs, and although crystallography has improved\textsuperscript{152}, there is still a limited amount of data regarding these structures. When attempting validation to recreate various repositioned drugs, we note that many of the queried drugs have no similar drugs within the PDB or only a few with only a single target annotation. The lack of drug-like molecules in the PDB is an inherently limiting factor of this approach. On the other hand, using the non-steroidal anti-inflammatory drug Celecoxib (CEL), we note both the intended target cyclooxygenase-2 and a discovered off-target carbonic-anhydrase II. However, such examples are likely to exist when an off-target example is discovered experimentally, and a subsequent crystal structure is generated to understand the mechanism of binding. Using kinase inhibitors as queries returns more useful information
considering the abundance of data for this target class. Given a kinase-like inhibitor, one may be able to deduce what kinases a molecule may bind. There is no predominant solution to this issue, and with the increase in structural data, such methods are expected to become more valuable. One possible solution to the lack of protein structural data in the PDB, could involve utilising protein structure prediction algorithms such as AlphaFold\textsuperscript{153}, which has been widely regarded as a “revolution” in the field\textsuperscript{154}. However, while AlphaFold has the potential to increase the amount of available structural data, it is not effective in addressing the lack of protein-ligand complexes. To tackle this issue, a model called AlphaFill\textsuperscript{155} has been developed to supplement AlphaFold’s predictions with information about small molecule data and ions found in experimentally defined structures. While this approach is useful in many applications, it has inherent limitations, the main being that it cannot include many drug-like molecules for which no existing 3D data exists.

A further issue is that PDB complexes are not annotated with affinity data. Hence, some complexes may indicate low-affinity interactions with limited drug discovery/repositioning use. PDB entries have relatively limited information regarding pharmacology and chemical information. A future direction may focus on the annotation of complexes and ligands with this additional information, including: is the molecule a drug? (approved, experimental), diseases associated, side-effects, UniProt classification\textsuperscript{156}, molecular function, pathway information, symptoms, and biological process information. This information may help with repositioning, although it will have to be intuitively displayed to retain comprehensibility. One could envisage a heterogeneous network with nodes representing different elements (Protein, Ligand, Disease etc.). We note that since we have worked on this project, heterogeneous network approaches have become a popular
method for discovering potential repositioning candidates systematically\textsuperscript{18}. However, not many tools have approached this from a visualisation perspective, likely due to the complexity associated with incorporating a vast amount of data.

A possible problem with the network construction itself is that the data returned is only a single layer deep (i.e. a ligand search only retrieves proteins that directly interact and ligands that are directly similar to the query). It may be interesting to explore deeper levels, where say, after a ligand search, extracted proteins are used as a seed to find similar proteins to themselves, hence a ‘deeper’ layer of information. Considering all of these points, a way to achieve this could be to create an extensive graph database compiled using information from various data sources. The resultant databases could be searched using different inputs, including ligand, protein, and other entities such as disease identifiers.

1.4.1.1 Database

For the data required to create networks based on similarity queries, three databases were compiled: a database containing all ‘drug like’ ligands within the PDB (Section 1.4.1.1.1), a database containing protein-ligand interaction data annotated with Pfam domains (Section 1.4.1.1.2) and a custom BLAST database for protein sequence similarity searching (Section 1.4.1.1.3). The decision to pre-compile databases rather than make API calls was efficiency and simplicity driven. Minimising API calls negates the reliance of the tool on external services and increases calculation speed. Databases are stored as Pandas DataFrames in pickled files. Pickling is a process in which a Python object hierarchy is
converted into a byte-stream. The benefit of this is that Python objects, such as the Pandas DataFrames and the Morgan Fingerprints within them, can be stored and read back into Python objects without reconstructing the objects from raw text data. For testing algorithms, this approach was ideal, although, in a production environment, these could be converted into a SQL database structure. Another potential approach to consider is utilising graph databases such as Neo4j\textsuperscript{157}, which uses a graph structure to store and retrieve data through semantic queries that involve nodes, edges and their associated properties. Because much of chemical and biological data is relational, a graph database may be more natural compared with a tabular structure, potentially simplifying the process of formulating complex queries. A current limitation of graph databases, however is that there is a distinct lack of efficient chemical search cartridges.

1.4.1.1.1 Ligand Database

A Ligand database comprising every ligand within the PDB was compiled using Python 3.6, and data was downloaded from the RCSB PDB Ligand Expo\textsuperscript{126}. In the downloaded data, each ligand is identified by an alphanumeric HET ID and, additionally, is associated with a SMILES string and a chemical or given name. The tab-separated file was parsed using the Pandas Python library into a DataFrame object, a 2-dimensional data structure with columns of varying data types. For each entry in this table, an RDKit molecule object was constructed using the libraries SMILES string parser, thus, enabling the calculation of a set of descriptors including: molecular weight, number of atoms, number of nitrogen and oxygen atoms, number of hydrogen bond donors, number of hydrogen bond acceptors,
component elements and a Morgan topological fingerprint with a radius of 2 (ECFP4). During compilation, ligands are filtered based on these descriptors, using the parameters defined in Table 1.2. Further database filtering removes molecules from a ‘blocklist’ containing non-desirable molecules such as stabilisers used in crystallography, polysaccharides or excipients.

1.4.1.1.2 Protein Ligand Interaction Database

The protein-ligand Interaction database was compiled using data from multiple sources. Protein-ligand binding information was downloaded from the PDBSUM downloads page in a text file format. This file maps all the ligands within the PDB, represented by their alphanumeric HET code, to the PDB ID(s) they interact with. Adding PFAM annotations for each entry was a more complex process since a protein or protein chain can contain more than one Pfam domain, of which the binding site may only occupy one of these domains. To ensure that the correct Pfam domain(s) was selected for each ligand-protein complex, GROW data, kindly provided by Roman Lakowski, was queried, used in the program LIGPLOT. This data contains information about the interactions between a ligand and each protein residue. Each residue is annotated with the protein chain identifier within which it resides and the residues ID. Using this data for each protein-ligand complex, it was possible to ascertain which protein chains and which specific residues were involved in the interaction. Pfam, data downloaded using the RCSB RESTful web service, was used to assign domains to each protein-ligand complex if the range of interacting residues fell within the range of residues within the given Pfam domain. Pfam
domain information includes the Pfam accession, Pfam description, e-value and the first and last residue of the domain. Additionally, using data obtained from the PISCES protein culling server\textsuperscript{159}, the non-redundant protein identity and chain for each ligand-protein complex was added to the table, where proteins with a sequence similarity of >70\% were collapsed into the entry with the best quality structure. PISCES calculates these identities by creating a hidden Markov model (HMM) for each unique PDB sequence and searching the resulting collection with each HMM using the program HHsearch\textsuperscript{160}. The >70\% cutoff was chosen to remove redundancy from the dataset, for ease of interpretability, while retaining representative sequences. Studies have indicated that at a value of 60\% identity, in 90\% of cases proteins share the same EC number, and at the domain level, a 70\% sequence identity is required for a 90\% confidence in full EC transfer\textsuperscript{144}. It is thus highly likely that at this threshold clustered proteins will share highly similar structure and function.

1.4.1.1.3 BLAST Database

Network construction from a protein sequence input relies on a BLAST search of the PDB, extracting similar proteins to the target sequence. The standalone command-line BLAST software was downloaded and installed, and a custom database was compiled using the FASTA sequences of all proteins within the PDB downloaded from the RSCB.
1.4.1.2 Ligand-Based Search

Using the compiled databases, similarity searches can be executed easily. The program logic is shown in Figure 1.4. An input SMILES is parsed using the RDKit, and a Morgan fingerprint is calculated with the same parameters as those populating the database. This fingerprint is used to calculate pairwise similarity with every ligand within the database using the Tanimoto coefficient, with any ligand with a similarity greater than 0.5 being kept for network construction. Binding information can then be merged into the resultant table using an SQL-like left merge with the HET ID as the common identifier. From the result, a network data structure (JSON) is constructed, consisting of an array of nodes and an array of edges. The ligands are rudimentarily clustered to construct the nodes based on their Tanimoto similarity as described in Section 1.3.3.2, with each node annotated with: HET ID, Tanimoto score, SMILES, PFAM accession and PFAM name. Protein nodes are constructed by grouping the protein targets into Pfam functional clusters. Finally, edges are constructed with the criteria that if a ligand node contains a ligand that interacts with a protein within a Pfam node, an edge is drawn. Only one edge may exist between a ligand node and a Pfam node. For performing 3D searches, the pipeline is practically identical. The only changes are that ten conformers are generated for the query molecule, and the search is performed using a pharmacophore alignment as described in Section 1.3.2.1.2

The main issue with the ligand search, as outlined previously, is that not enough ‘drug like’ molecules exist within the PDB. The other main issue resides within the clustering of molecules. In the bucketing approach, the intra-cluster similarity is not respected and thus, drawing links between ligand and protein clusters may not be entirely
relevant. Some ligands within a cluster may not interact with a protein cluster to which an edge is drawn. This issue is intrinsically complex, with a fine line between comprehensibility and accuracy. Future work may wish to put effort into discovering a novel visualisation scheme, which may leverage a wide variety of graph layout algorithms implemented in graph visualisation libraries such as Cytoscape\textsuperscript{161} and D3.js\textsuperscript{150}. The subjectivity of molecular similarity will always be problematic in this regard.

1.4.1.3 Protein-Based Search

The logical process for the protein search pipeline is shown in Figure 1.5. A BLAST search is executed with a FASTA input; the output is captured in a temporary CSV file before parsed. Binding information is merged, as before, with protein ID and chain as the common identifier. A JSON network representation is built from the search results, much in the same way as from the ligand perspective. In this case, protein nodes are constructed from grouping proteins into rudimentary clusters based on their sequence identity, as described previously (Section 1.3.2.2), whereas ligands are clustered using all-by-all molecular similarity. One issue identified relates to rudimentary clustering. In the same way, rudimentary ligand clustering may produce non-meaningful links; the rudimentary clustering of proteins is likely to succumb to the same problem. From the ligand point of view, scaffold tree-based clustering generates the most visually comprehensible clusters, while Butina produces too many singleton clusters for comprehensible visualisation, and K-means requires some form of knowledge \textit{a priori}. 
Figure 1. An illustration of the implemented ligand search pipeline. The shown pipeline is for fingerprint-based search queries.
Figure 1. 5 An illustration of the implemented ligand search pipeline. The shown pipeline is for fingerprint-based search queries.
However, the fundamental issue with using BLAST for this application is that comparing global homology is a poor way to identify off-targets and repurposing candidates. The effects of convergent and divergent evolution lead to the existence of proteins sharing similar function with minimal global fold or sequence homology. Ideally, a quantification of local similarity should be used, i.e. binding site comparison. We believe that utilising methods for binding site comparison is one of the most promising research directions, especially due to the abundance of protein structures available. Using binding site prediction technology also allows one to search using targets for which binding data is unknown, precluding the strict requirement for ligand data. In chapters 2 and 3 we propose novel methods for binding site comparison using machine learning-based approaches. The proposed methods compete with the state-of-the-art, and judgments from these may be integrated into the visualisation. Due to the inevitable noise generated by local similarity methods, a metamethod combining the results of multiple tools may be employed to improve confidence.

1.4.1.4 Interface

A basic interface has been implemented, enabling interaction with the constructed visualisation. The interface consists of: a home page which also serves as a page for users to submit inputs (Figure 1.6A), a loading page (Figure 1.6B), a results page (Figure 1.7), an error page (Figure 1.8) and an about page. In the example, a walkthrough of a ligand search is displayed for the molecule Triclosan. After input (Figure 1.6A), the user is directed to a loading page (Figure 1.6B) showing the input rendered as an image and a
session key which is used in the results page URL. When the results have finished calculating, the user is redirected to a results page (Figure 1.7), where they are presented with a network rendered within a white box, with a blank box on the right-hand side. In the top right corner, the input molecule is depicted as an image, while below the network box are four buttons labelled: legend, statistics, download and reset. The legend button is selected by default, displaying a legend and description of the graph beneath.

Selecting the statistics box brings up a panel below, replacing the legend. The statistics include: relationship statistics (number of ligand clusters, number of protein clusters, number of unique ligands, number of unique proteins including chain and number of unique proteins not including chains), ligands cluster statistics providing further details for each ligand cluster (number of ligands, number of Pfam (protein cluster) associations and number of PDBs) and protein cluster (Pfam) cluster statistics providing further detail for each protein cluster (number of ligands, number of PDBs). The download button brings up a panel with a link to download all the SMILES representations presented in the graph. The reset button resets the positions of graph nodes.
Figure 1. 6 Screenshots of the LigNFam interface. (a) shows a screenshot of the home page where a SMILES input box and a FASTA input box are displayed for user input. (b) is a screenshot of the loading page after a ligand is input. The structure is rendered in the center of the page.
Figure 1. 7 Screenshots of the LigNFam results interface. Top: The center panel contains the network representation of protein-ligand relationships and the bottom panel displays information depending on the tab selected. The network is interactive and can be manipulated and expanded. Bottom: Selecting a node reveals the contents of that node in a tabular format in the right-hand pane. For ligand nodes this includes a depiction of each molecular structure in the cluster and the name and Pfam annotation of the protein it is bound to. In the case of protein nodes, depictions of the molecular structures of all associated ligands are displayed. All annotations can be selected, generating a popup window that contains links to various external resources where further information can be found (RCSB PDB\textsuperscript{47}, PDBe\textsuperscript{162}, PDBSum\textsuperscript{130} and Pfam\textsuperscript{103}). The network uses a force-directed, physics-based layout, where nodes can be manipulated to aid visual comprehension. Manipulated nodes become frozen, and can be reset to their initial state using the “reset” button in the lower right.
Figure 1. 8 Screenshot of the LigNFam error page. The error is presented when a sessions data is no longer available, or the request fails.

1.5 Conclusions

The developed interface, dubbed LigNFam, provides a user interactive representation of protein-ligand relationships through the construction of similarity-based networks. The interface, facilitates the answering of common questions in the drug discovery pipeline including: what ligands are similar? What proteins and protein-families do they bind to? And what are the functional relationships between these proteins? This will hopefully help identify: off-target side effects, distantly related proteins exhibiting similar binding characteristics and aid in the understanding of structure-activity relationship. Inevitably, the main objective is to aid with the process of drug repurposing. The current implementation includes, both fingerprint-based and 3D ligand pharmacophore searching abilities. Protein searches are currently implemented using BLAST sequence similarity although we acknowledge that local based similarities are more important for this purpose.
In the next sections, machine learning based methods for binding site comparison are proposed which are expected to aid provide significant improvements for understanding relationships between proteins and the ligands they bind.
II Machine Learning Based Approaches for Binding Site Comparison

2 Chapter 2: Introduction and Motivation

2.1 The Anatomy of a Binding Site Comparison

Modern methods for binding site comparison frequently use simplified representations of structure and physiochemical properties\textsuperscript{163}, often inspired by developments in molecular docking, 3D ligand alignments, and general computer vision. The general assumption is that ligand binding requires a certain degree of geometrical and physiochemical complementarity as afforded by the lock and key model\textsuperscript{164}. The algorithm space can be broadly classified into alignment-based and alignment-free methods. By way of comparison, alignment-based methods tend to provide a more rigorous evaluation of similarity, using structural superposition, providing a visual output that can be beneficial when interpreting results\textsuperscript{163}. Alignment methods are particularly time-consuming despite rigour since multiple evaluations must be made per pairwise comparison, substantially limiting their high-throughput application. In contrast, alignment-free methods are more approximate, some employing vector/histogram-based descriptions\textsuperscript{165,166} for ultra-fast nearest neighbour retrieval. The speed benefit is significant for high-throughput studies and thus can be used as a pre-screening step where a more rigorous approach is applied to rank retrieved ‘hits’.
Irrespective of whether an algorithm for binding site comparison is alignment-based or alignment-free, the first stage must involve computing some form of representation\textsuperscript{163}. Due to its subjectivity and complexity, it is not surprising that a diverse array of methods for representing a binding site exist. High-resolution methods such as all-atom models may be used, but models tend to utilize coarser representations such as physiochemically annotated feature point\textsuperscript{167} or Cα/Cβ positions\textsuperscript{168}. Other representations include graphs\textsuperscript{167,169}, molecular surfaces\textsuperscript{170–172}, molecular interaction fields (MIFs)\textsuperscript{129} or geometric projections\textsuperscript{168}. It is also possible for multiple methods to be combined, for example, SiteAlign\textsuperscript{168} computes a geometric projection given Cα/Cβ positions.

Regardless of the representation method, it must at least contain some description of both shape and physiochemical properties. The extent of approximation impacts the precision of the similarity evaluation and can be controlled through the resolution of the representation (sensitivity to atomic positions), matching tolerance (the distance/similarity at which two points are matched) and feature complexity\textsuperscript{163}. In the case of an all-atom representation, or a discretised grid, small fluctuations may have a large impact on similarity scores and thus also the balance between sensitivity and specificity. Decreasing resolution is achieved through spatial decomposition, such as representing residues as sets of physiochemical feature points (pharmacophoric points) as demonstrated by CavBase\textsuperscript{167}, SuMo\textsuperscript{173} and SiteEngine\textsuperscript{172}. Further reductions may consider one or two points for each residue such as the Cα or Cβ positions, as demonstrated by tools such as SiteAlign\textsuperscript{168}, PocketMatch\textsuperscript{166} and TM-Align\textsuperscript{174} (TM-Align is not explicitly a binding site comparison tool but is used for this task in multiple publications\textsuperscript{175,176}). Such approximations may be able to retrieve more distant similarities suitable for off-target prediction, however, may
also undervalue fine-grain features such as the importance of side-chain orientations or the solvent accessibility of locations where interactions may be mediated.

On the other hand, lower-resolution methods may be less sensitive to intrinsic flexibility, including small conformation changes such as side chain rotameric states up to larger-scale rearrangements. Despite less sensitivity to flexibility, larger motions are more difficult to account for, especially since all comparison methods model the binding site as a rigid entity. In principle, surface-based methods, such as Shaper\textsuperscript{171}, eF-Site\textsuperscript{177} and ProBiS\textsuperscript{178}, using point clouds or triangulated meshes should be the most sensitive since minor conformational changes can significantly impact the distribution of points on the molecular surface.

The matching of individual points is also critical, especially since similar residues may have the ability to mediate similar interactions. In the case of feature annotated points, the ability to mediate non-covalent interactions (hydrogen bond acceptors, donors, mixed donor/acceptors, aliphatic/hydrophobic and aromatic interactions) and the directionality of these features should be considered. Increasing the complexity of matching acceptance criteria can lead to more precise matches but must be designed carefully to minimise missed similarities. In interaction-based methods such as: TIFP\textsuperscript{169}, KRIPO\textsuperscript{179}, GRIM\textsuperscript{169} and IsoMIF\textsuperscript{129}, a different matching paradigm is used where instead of comparing features of binding sites directly, the interactions made between ligand and protein are considered.

Arguably, the most important part of a binding site comparison is processing a pair of representations to quantify the similarity. Ideally, this should correlate with biological similarity analogous with the similarity property principle (SPP). However, the SPP is more
difficult to apply in this context, and the significance of a match should be evaluated in a context-dependent manner. Quantification of similarity is highly dependent on the input representation and how it is processed. For example, an alignment-based method may consider the degree of overlap between points, whereas this is not possible in an alignment-free approach. Independent of the representation, however, scoring schemes can also weigh potential matches changing the degree to which local and global similarities impact the result. Descriptor based methods such as: RAPMAD$^{165}$, TIFP$^{169}$ and PocketMatch$^{166}$ are generally global descriptors and allow fast alignment-free comparison. They may be computed in diverse approaches, including the geometric hashing of pharmacophoric triplets$^{172}$ or the construction of histograms through binning the positions of pseudo-atoms based on their distance to a central point$^{165}$. SiteAlign represents a hybrid approach being both descriptor-based and alignment dependent where fingerprint similarities on a geometric projection are used to optimise a superposition. The metric used to measure the similarity, distance, or mutual information between descriptors is a vital choice considering it produces a quantification of the match and can have a sizable impact on the tool's effectiveness. Other alignment-free methods may use handcrafted scoring schemes based on the bipartite matching of local features paired with complex feature weighting schemes$^{124}$. Alignment-based methods are also very diverse, with some using graphs paired with clique-based matching schemes$^{167}$ and others using spherical gaussian overlap optimisation$^{170,171}$. CavBase$^{167}$ uses a clique-based scheme on product-graphs constructed from pseudo-atoms annotated with information about their physiochemical classification, local surface shape properties, and orientation to the molecular surface and cavity center. Building product graphs thus requires complex empirical matching criteria to accept
matching feature pairs. An interesting property of clique-based matching is that multiple alignments can be produced for any given binding site pair, although only the top-scoring alignment is considered in many cases. Instead of using cliques, VolSite/Shaper\textsuperscript{171} optimises the alignment of shape and physiochemically annotated grid-points using a differentiable gaussian overlap optimisation approach. The grid points represent a negative image of the binding site (pseudo-ligand), and therefore the alignment represents a superposition of the molecular surface. Surface-based approaches, including SiteEngine\textsuperscript{172}, eF-Site\textsuperscript{177} and to an extent CavBase (extended)\textsuperscript{180}, are interesting cases capturing important information about amino acid functional groups on molecular surface vertices/faces. This approach benefits from less dependence on the underlying amino acid sequence.

Most of the mentioned tools utilise normalised scoring schemes used for ranking pairs based on computed scores. Often these scores are designed to lie with a certain range, or are normalised based on the number of matching points, or the size difference between two binding sites. Often publications describing novel tools will also present ‘optimal’ thresholds for similarity searching, however, these should be taken with a pinch of salt as they are likely to reflect optimization bias. The SitesBase\textsuperscript{181} software takes an interesting approach for the determination of significance, using a non-redundant protein database to generate an extreme distribution of scores from which p-values can be calculated to analyse the change of achieving a given score by chance.

Table 2.1 presents a non-extensive summary of binding-site classification tools, their classification, alignment dependence, and availability. Many of the tools in this table will also be used for benchmarking the algorithms proposed herein. Further extensive
reviews of binding site comparison have been published by Kellenberger et al.\textsuperscript{163} and Ehrt et al.\textsuperscript{175,182}.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Classification</th>
<th>Computes Alignment</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavbase\textsuperscript{167}</td>
<td>Residue</td>
<td>Yes</td>
<td>Commercial (CCDC)*</td>
</tr>
<tr>
<td>RAPMAD\textsuperscript{165}</td>
<td>Residue</td>
<td>No</td>
<td>Commercial (CCDC)*</td>
</tr>
<tr>
<td>FuzCav\textsuperscript{183}</td>
<td>Residue</td>
<td>No</td>
<td>Academic</td>
</tr>
<tr>
<td>PocketMatch\textsuperscript{166}</td>
<td>Residue</td>
<td>No</td>
<td>Open Source</td>
</tr>
<tr>
<td>SiteAlign\textsuperscript{168}</td>
<td>Residue</td>
<td>Yes</td>
<td>Academic</td>
</tr>
<tr>
<td>SMAP\textsuperscript{184}</td>
<td>Residue</td>
<td>Yes</td>
<td>Academic</td>
</tr>
<tr>
<td>TM-Align\textsuperscript{174}</td>
<td>Residue</td>
<td>Yes</td>
<td>Open Source</td>
</tr>
<tr>
<td>G-LoSA\textsuperscript{185}</td>
<td>Residue</td>
<td>Yes</td>
<td>Open Source</td>
</tr>
<tr>
<td>ProBiS\textsuperscript{178}</td>
<td>Surface</td>
<td>Yes</td>
<td>Open Source</td>
</tr>
<tr>
<td>VolSite/Shaper\textsuperscript{171}</td>
<td>Surface</td>
<td>Yes</td>
<td>Academic</td>
</tr>
<tr>
<td>SiteEngine\textsuperscript{172}</td>
<td>Surface</td>
<td>Yes</td>
<td>Open Source</td>
</tr>
<tr>
<td>SiteHopper\textsuperscript{170}</td>
<td>Surface</td>
<td>Yes</td>
<td>Commercial (OpenEye)</td>
</tr>
<tr>
<td>IsoMIF\textsuperscript{129}</td>
<td>Interaction</td>
<td>Yes</td>
<td>Open Source</td>
</tr>
<tr>
<td>KRIPO\textsuperscript{179}</td>
<td>Interaction</td>
<td>No</td>
<td>Open Source</td>
</tr>
<tr>
<td>TIF\textsuperscript{169}</td>
<td>Interaction</td>
<td>No</td>
<td>Academic</td>
</tr>
<tr>
<td>GRIM\textsuperscript{169}</td>
<td>Interaction</td>
<td>Yes</td>
<td>Academic</td>
</tr>
</tbody>
</table>

Table 2. 1 Examples of binding site comparison tools using diverse approaches. Many of the tools mentioned are used for comparison in this work. The term ‘academic’ means that the software is free for academic research, but a license is required for other use cases. *CCDC is an acronym for the Cambridge Crystallographic Data Center.
2.2 Evaluation of Binding Site Comparison Algorithms

Due to similarity's complex and subjective nature, it is especially difficult to evaluate quantitively and highly dependent on context. It is thus not surprising that many binding site comparison methods employing different representations and scoring systems exist. Given these complexities, it is improbable that a method addressing all possible applications will ever be developed.

The most common evaluation method for binding site comparison is the ability to retrieve binding site pairs based on the binding of a common or similar ligand. For further robustness, this may also be guided by bioactivity annotations. Shared binding is a valuable metric that is biologically relevant and appropriate for evaluating applicability to contexts such as off-target interaction prediction, polypharmacology analysis, and drug repurposing.

Restating the context dependence of evaluations, thresholding scores, distances or similarities to calculate simple accuracy-based metrics is a bad idea. Thresholds must be determined separately for different use cases, and a hard dependence on thresholds will often lead to missed, potentially useful, similarities. Evaluations are thus performed for each query where the receiver operating characteristic (ROC) curve\textsuperscript{186} is used to measure performance. The ROC curve measures performance of a classification at all classification thresholds. The curve is plotted using the metrics: true positive rate (TPR or recall rate) on the $y$-axis and false positive rate (FPR) on the $x$-axis:
The area under the ROC curve (AUC) provides an aggregate measure of performance across all the potential decision boundaries and therefore represents the probability that a model ranks a positive example higher than a negative example. An AUC of 0.5 would thus indicate a random chance of ranking a random positive example higher than a negative, while an AUC of 1.0 indicates a perfect classification. The ROC curve and AUC are both scale-invariant and classification threshold invariant making them a popular choice for this task.

One issue with relying too heavily on the ROC curve and AUC is that for some tasks, one may want to consider the cost of false negatives versus false positives, i.e. when predicting a medical diagnosis minimizing false negatives even though this may come at the cost of increasing the number of false positives. In this case, predicted positives are likely to be subject to more rigorous evaluation before making a final diagnosis judgment.

For an individual evaluation, the optimal threshold can be determined using the Youden’s J statistic:

\[
J = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1
\]
The two right-hand quantities are TPR (sensitivity) and TNR (specificity). The J statistic is calculated for all points on the ROC curve, and the maximum value is taken as the optimal value for transforming a numerical output into a dichotomous result. In the binding site comparison context, the specificity measures a tool's ability to reject dissimilar binding site pairs within a given threshold. Therefore, the sensitivity measures a tool's ability to retrieve known binding site similarities correctly within the threshold. For specific tasks, the sensitivity and specificity metrics can be used to optimize the results obtained, selecting an appropriate application threshold. For example, when considering drug-repurposing, one may wish to optimize for specificity since only binding sites of high similarity are interesting, whereas, when predicting off-target interactions, one may wish to optimize for sensitivity where even minor similarities may be significant.

Another useful metric used by Ehrt et al.\textsuperscript{175} and commonly used in small molecule virtual screening is the enrichment factor (EF), describing the enrichment of similar (active) binding site pairs compared to the number of similar pairs that might be recovered during a random selection:

\[
EF_{x\%} = \frac{N_{\text{actives}_{x\%}}}{N_{\text{actives}_{100\%}}} / \frac{N_{x\%}}{N_{100\%}}
\]

The EF is a parametrised metric where for \(x\%\), the \(EF_{x\%}\) is calculated by the number of actives recovered within \(x\%\) of the ranked dataset: \(N_{\text{actives}_{x\%}}\), and the number of total
pairs at $x_{90\%}$ divided by the total number of actives in the dataset as a proportion of the total number of pairs: $N_{\text{actives}_{100\%}}/N_{100\%}$. Ideally one would observe a high early enrichment i.e. many actives at low values of $x_{90\%}$, where similar pairs are more often ranked highly in comparison to inactive pairs (dissimilar).

### 2.3 Binding Site Databases

Due to the rapid expansion in protein 3D structural data availability and the uptake of structure-based drug design strategies, the number of molecular modelling algorithms has also increased. For many of these molecular modelling procedures, including: docking and structure prediction, rigorous benchmarks and datasets exist for evaluating these procedures\textsuperscript{188,189}. While global structure comparison is reasonably simple to evaluate, local comparisons are much more challenging due to increased subjectivity. Most commonly-used binding site comparison tools are evaluated on datasets where proteins binding the same ligand are regarded as positive pairs and where a lack of shared binding is regarded as forming a negative pair. While this is adequate in some cases, the definition of negative pairs is problematic since the lack of structural data confirming the shared binding of a similar ligand does not preclude this from occurring in reality. Another issue within the domain is that most algorithms are optimised and evaluated on different datasets, making comparisons between methods difficult. Furthermore, these datasets often contain biases where proteins classified as similar in their binding site also come from the same protein fold or have a high sequence identity. Many evaluations thus overestimate the performance of the algorithm. Datasets containing binding site information, useful for evaluation, and
recent datasets specifically compiled for addressing the lack of standardised benchmarks are discussed below. Throughout this work, the datasets discussed are used to evaluate the tools proposed and developed herein.

2.3.1 ProSPECCTs

Considering that most binding site comparison tools take a wide variety of approaches and are optimised for specific datasets, it is tough to identify the strengths and weaknesses of existing approaches when considering a specific application. Until recently, no standardised benchmarking dataset existed, in contrast with other molecular modelling domains with much more rigorous evaluation procedures. Compiled by Ehrt et al., the Protein Site Pairs for the Evaluation of Cavity Comparison Tools (ProSPECCTs) dataset\textsuperscript{175} represents the most comprehensive dataset used to evaluate binding site comparison tools. The dataset was accompanied by a large-scale evaluation attempting to suggest the best tool(s) for use in different application areas. To this end, the datasets consist of 10 data subsets (P1-P7) containing positive (similar) and negative (dissimilar) binding site pairs, each designed to test binding-site comparison approaches within a different context. Each dataset is detailed further and summarised in Table 2.2.

The first dataset in the collection (P1/.2) contains structures with identical sequences to evaluate sensitivity to varying binding site definitions, which may be defined using a bound ligand or from a binding site prediction tool. Ligands commonly occupy different parts of the same binding site, including sub-pocket regions. Furthermore, diverse ligands
may bind in the same region, sometimes with different interaction patterns. Binding site similarity tools should have the ability to enrich similar binding sites regardless of how the binding site is defined. P1 contains proteins with identical sequences binding to diverse ligands, while P1.2 contains proteins with identical sequences binding to strictly similar ligands. Comparisons between both sets should reveal a tool's sensitivity to binding site definition, particularly if it heavily relies on the structure of the ligand itself.

P2 evaluates sensitivity to binding site flexibility, where positive pairs are formed from nuclear magnetic resonance (NMR) ensembles. Tools should be able to account for a certain degree of flexibility when searching for similar binding sites since proteins are flexible entities. Poor performance could indicate that a tool is too sensitive to the shape of a binding site.

Subsets P3 and P4 are handcrafted decoy datasets in which a series of up to five mutations have been artificially introduced. The benchmark evaluates a tool's ability to recognise small changes, enriching those with the original sequence over those with mutations. The P3 subset (decoy set 1) only considers mutations resulting in physiochemical property changes, whereas subset P4 (decoy set 2) also considers mutations that result in shape and physiochemical changes.

The following four subsets (P5/.2, P6/.2) were extracted from literature sources involving the comparison of binding site properties. The first set published by Kahraman et al.\textsuperscript{190} was designed to evaluate the relationship between ligand and binding site shape. Analysis revealed that binding site shape alone could not infer the shared binding of a common ligand, although no attempt was made to analyse if any local physiochemical
features were conserved between different proteins sharing a common ligand. P5.2 contains the originally published dataset, and P5.2 is the same dataset without considering phosphate (PO₄) binding sites. Removing phosphates allowed the fair comparison of tools that fail to process phosphate binding sites due to the small size. The P6/.2 subsets were published initially by Barelier et al.¹⁹¹, in which the authors identified three different types of molecular recognition events occurring in different proteins binding identical ligands. Interactions made between similar binding site residues, interactions made with dissimilar residues but from common functional groups, and interactions made with dissimilar residues at different functional groups of the same ligand. The dataset represents a challenging benchmark where similarities may not be as apparent as other datasets.

The final subset proposed, P7, represents a more realistic scenario where positives pairs are obtained from a previous analysis¹⁸². The true positives are combined with a selection of binding sites from the sc-PDB¹⁹², serving as negatives for the binding sites participating in positive pairs. The dataset evaluates the ability of tools to retrieve binding sites that have displayed applications in medicinal-chemistry projects.

The conclusion is that similarity is subjective and that tools should be selected with application in mind, and the use of multiple tools is encouraged. The second important observation is that selecting absolute thresholds for the binary classification of binding sites will often disregard interesting and potentially novel similarities. One should therefore be careful when evaluating hits from binding site comparison tools.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Objective</th>
<th>Active Pairs</th>
<th>Inactive Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Sensitivity with respect to binding site definition</td>
<td>13,430</td>
<td>92,846</td>
</tr>
<tr>
<td>P1.2</td>
<td>Impact of ligand diversity on binding site comparison</td>
<td>241</td>
<td>1,784</td>
</tr>
<tr>
<td>P2</td>
<td>Sensitivity with respect to binding site flexibility</td>
<td>7,729</td>
<td>100,512</td>
</tr>
<tr>
<td>P3*</td>
<td>Differentiation between binding sites with different physiochemical properties</td>
<td>13,430</td>
<td>13,430</td>
</tr>
<tr>
<td>P4*</td>
<td>Differentiation between binding sites with different physiochemical &amp; shape properties</td>
<td>13,430</td>
<td>13,430</td>
</tr>
<tr>
<td>P5</td>
<td>Classification of proteins binding to identical ligands and cofactors (without phosphate)</td>
<td>920</td>
<td>5,480</td>
</tr>
<tr>
<td>P5.2</td>
<td>Classification of proteins binding to identical ligands and cofactors</td>
<td>1,320</td>
<td>8,680</td>
</tr>
<tr>
<td>P6/P6.2</td>
<td>Identification of distant relationships between binding sites with identical ligands “observing” similar environments</td>
<td>19</td>
<td>43</td>
</tr>
<tr>
<td>P.7</td>
<td>Recovery of known binding site similarities within a diverse set of proteins</td>
<td>115</td>
<td>56,284</td>
</tr>
</tbody>
</table>

Table 2. Summary of datasets contained within the ProSPECCTs benchmark dataset collection\textsuperscript{175} to analyse the strengths and weaknesses of binding site comparison tools.

*P3/P4 shows the number of pairs with five residue variants. When considering all (1-5) mutations, inactive pairs are 67,150.

2.3.2 TOUGH-M1

The da\textsuperscript{T}aset t\textsuperscript{O}eval\textsuperscript{U}ate al\textsuperscript{G}orit\textsuperscript{H}ms for binding site Matching (TOUGH-M1) was proposed by Govindaraj and Brylinski\textsuperscript{123} to compare binding-site similarity evaluation
algorithms on pairs of proteins sharing a common ligand, yet sharing no global homology.

The authors claim that datasets historically used to evaluate the performance of binding site comparison tools are biased, often being build alongside the tool itself and optimized for highly specific use cases. Many of these datasets also contain binding-sites irrelevant for drug-discovery, including those bound to molecules such as solvents, precipitants and additives. The proposed dataset was thus introduced to provide a less biased evaluation set, using an automated approach to extract a large, balanced, diverse representative set of binding-site pairs annotated as positive or negative based on the binding of, or lack of, a shared ligand. The figure describes the process used to construct the dataset in further detail. 2.1.

The first step taken by Govindaraj and Brylinski\textsuperscript{123}, was the extraction of ligand-bound protein structures in which the bound ligand is similar to at least one approved drug within the Drugbank database\textsuperscript{48}. The similarity was assessed based on binary fingerprints using the Tanimoto coefficient, retaining molecules with a score greater than or equal to 0.5. Structures meeting these criteria were clustered with CD-HIT\textsuperscript{193} at 40\% sequence identity, presumably since proteins sharing >40\% identity are likely to share a similar function (up to three conserved digits of an EC number)\textsuperscript{144}, and representative structures binding dissimilar ligands were selected from each cluster. The authors further impose a constraint that evaluated tools should use predicted binding sites as input, hence, binding pockets were predicted using the Fpocket (2.0) software\textsuperscript{194}. Structures were only retained if the ligand-binding region was predicted adequately. Bound ligands were subsequently clustered with a Tanimoto threshold of 0.7, producing 1266 molecular clusters. Within clusters, proteins were selected pairwise with Fr-TM-Align\textsuperscript{195} alignment scores of 0.4.
These dissimilar structures binding similar ligands represent the positive set of TOUGH-M1 forming 505,116 binding site pairs. The negative set was constructed by comparing proteins inter-cluster and adding pairs with Fr-TM-Align alignment scores of 0.4 to the negative set. These dissimilar proteins binding dissimilar ligands total 556,810 binding site pairs.

Figure 2. An illustration of the protocol used by Govindaraj and Brylinski to construct the TOUGH-M1 dataset. The binding site pair dataset contains 7,524 protein structures forming 505,116 “similar” (positive) and 556,810 “dissimilar” (negative) binding site pairs.
2.3.3 TOUGH-C1

The daTaset тO evaluate algorithms for binding site Classification (TOUGH-C1) were compiled by Pu et al.\textsuperscript{196} for training machine learning models to predict binding site classifications based on bound ligands. The main subset of TOUGH-C1 contains nucleotide- and heme-binding proteins providing the training labels and data. These classifications are considered since they represent abundant ligands within the PDB and bind to proteins with diverse global structures. The final dataset contains 1,553 nucleotide- and 596 heme-binding structures representing cluster centers after clustering with CD-HIT\textsuperscript{193} with a threshold of 80%. The binding sites are defined through ligand-protein contacts. A further negative control dataset was compiled as a subset of TOUGH-M1\textsuperscript{123} where only proteins with a sequence identity 40% or TM-score 0.5 to any of the heme/nucleotide proteins were kept.

The collection was further filtered, discarding any protein binding a ligand similar to ATP, heme or 17β-estradiol (Tanimoto > 0.5), leaving 1,946 structures as the control set. The last remaining set of TOUGH-C1 is a small set of 69 steroid-binding proteins (Tanimoto > 0.7 to 17β-estradiol) acting as external negative control. A summary of the dataset is shown in Table 2.3.
<table>
<thead>
<tr>
<th>Classification</th>
<th>Number of Examples</th>
<th>Train/Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nucleotide</td>
<td>1,533</td>
<td>Train/Evaluation</td>
</tr>
<tr>
<td>Heme</td>
<td>596</td>
<td>Train/Evaluation</td>
</tr>
<tr>
<td>Control</td>
<td>1,946</td>
<td>Train/Evaluation</td>
</tr>
<tr>
<td>Steroid</td>
<td>69</td>
<td>Evaluation</td>
</tr>
</tbody>
</table>

Table 2. Summary of the TOUGH-C1 dataset for distinguishing protein binding sites by interacting ligand classification. The last column shows which subsets are used to train and evaluate machine learning models.

2.3.4  Vertex

The Vertex dataset is a benchmark proposed by Chen et al. containing protein pairs constructed from protein-level similarities. The construction of pairs is informed by bioactivity, where a positive pair is constructed if two proteins share similarities within the experimentally recorded bioactivities of their respective ligands. A negative pair thus must share common ligands with different measured bioactivities. The authors require that three common ligands are available with bioactivity ≤ 10µm for one of the proteins and >10µm for the other. A further drug-like filter was applied to all recovered molecules to filter out the impact of large un-specific molecules. The final dataset encompasses 6029 protein structures involving 6598 positive and 379 negative pairs. Since each protein structure may contain multiple binding sites/ligands, pairwise comparisons between two proteins must consider the result from multiple pairwise comparisons from all binding site combinations.
Therefore, the total pairwise comparisons are 1,461,668 for the positive pairs and 102,935 for the negative pairs.

Post-hoc analysis of the compiled dataset shows that the positive protein pairs have a median sequence identity of 17.5% and that 89.4% have a sequence identity in the range of 10-40%\textsuperscript{197}. Lack of sequence homology between pairs sharing common activities presents a more robust benchmark for binding site similarity since the data-id is not biased due to a sequence homology that could easily be detected with a simple global sequence alignment. Furthermore, a negative set based on activity difference makes more sense than defining a negative pair based on the lack of a shared co-crystallised ligand and low global homology. Indeed, these pairs may very well have the ability to share common active ligands. On the other hand, defining negatives in such a way requires confidence, and even with the wealth of data available, the dataset is still clearly biased towards positive pairs, with an \(~17\)-fold enrichment compared with negative pairs (6598 versus 379).

2.3.5 sc-PDB

The sc-PDB database\textsuperscript{192} is an extensive compilation of ‘ligandable’ binding sites from data extracted from the Protein Data Bank (PDB). Proteins are selected if they are in complex with a small synthetic or natural ligand (140 Da < MW < 800 Da). Since its conception in 2011, a machine learning model has also been introduced to filter proteins based on a predicted ‘ligandability’ metric. Before entry into the database, all entries are curated and checked manually. The curation process makes all entries suitable for docking studies.
through a pipeline consisting of: binding site identification, structure correction, including the addition of hydrogen atoms and optimisation of hydrogen-bonding networks. The final stage annotates the entry with protein function, ligand properties, and binding mode characterisation. The current database contains: 16034 entries, 4782 proteins and 6326 ligands (February 2022), representing a hugely beneficial resource for computational drug discovery. Given that the dataset inherently contains only “ligandable” sites screening the sc-PDB for related binding sites, a query could be used for drug-repurposing or scaffold-hopping.

### 2.3.6 Dataset Redundancy

All of the mentioned databases are drawn from the same underlying distribution, i.e. structural data from the PDB, hence, a certain degree of redundancy is expected. When evaluating or optimising algorithmic developments, it is critical to understand the degree of redundancy within evaluation data. This is especially true when developing methodologies utilising supervised machine learning algorithms. Machine learning uses subsets of data to train the weights within a statistical model and must be evaluated on data drawn from a different distribution to accurately indicate how a model will generalise to out-of-distribution (unseen) data. This process is more difficult with protein structural data since evolutionary relationships exist within the data. To assess the redundancy in the datasets presented, overlaps between each set were computed based on Uniprot accession codes. All chains within each protein were considered for the vertex dataset since that matches are computed in a bi-partite manner from all available pockets. The sc-PDB is not
included in overlap analysis because it is not explicitly a dataset designed for benchmarking binding site comparison algorithms. Overlaps between datasets are displayed in Figure 2.2 and Table 2.4.

Figure 2.2 A Venn diagram displaying the redundancy within binding site comparison benchmark datasets based on Uniprot accession codes. Diagram created using the DrawVenn software\textsuperscript{198}.\n
Table 2. 4 Table showing the number of entries and unique entries in terms of Uniprot accession codes, evaluating the redundancy within the datasets themselves. *The number of entries may be less than the reported number of structures since accession codes were unavailable for all protein entries in the databases. There are more Uniprot accession codes than protein entries for the Vertex dataset since each entry corresponds to multiple protein chains.

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of Entries*</th>
<th>Number of Unique Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProSPECCTs</td>
<td>2,776</td>
<td>1,274</td>
</tr>
<tr>
<td>TOUGH-M</td>
<td>7,332</td>
<td>7,125</td>
</tr>
<tr>
<td>TOUGH-C</td>
<td>4,086</td>
<td>3,959</td>
</tr>
<tr>
<td>Vertex</td>
<td>6,218</td>
<td>490</td>
</tr>
<tr>
<td><strong>Overall</strong></td>
<td></td>
<td><strong>8395</strong></td>
</tr>
</tbody>
</table>

2.4 The Application of Machine Learning to Binding Site Comparisons

The diversity of binding site comparison tools is unsurprising given the subjectivity of the domain and the fact that shared ligand binding is highly dependent on both the ligand and protein structure. It is often the case that proteins binding identical or similar ligands share no obvious structural similarities evaluating similarity as a complex problem. Binding site comparison algorithms quantify similarity using hand-crafted representations and complex
feature weighting schemes. These algorithms are also commonly optimized and validated on small sets of binding site pairs often containing sources of bias. We propose that learning from data rather than hand-crafting weighting schemes has the potential to reduce bias in binding site comparisons. Indeed, this opinion is also shared by others.

Before introducing our proposals, a brief introduction to machine learning and its application to three-dimensional and molecular data is presented.

### 2.4.1 Machine Learning

Machine learning is a domain of computer science focusing on algorithms that can ‘learn’ from data. A trained model can make predictions or decisions based on new observations. A typical machine learning model aims to learn a function $f: X \rightarrow Y$, or hypothesis, that maps an input space $X$ to an output space $Y$ representing possible predictions. These algorithms can broadly be categorized into: supervised, unsupervised and weakly-supervised approaches, describing how the training process is conducted.

#### 2.4.1.1 Supervised Learning

Supervised learning algorithms make use of labelled datasets, $S = (x_1, y_1), ..., (x_n, y_n) \in X \times Y$ where $x_i$ corresponds to a data point and $y_i$ to its corresponding target value. In a classification scenario the target space $Y$ is discrete, with $|Y|$ being the number of classes and in a regression scenario we may consider $Y = \mathbb{R}^m$. 
Formally put, labelled pairs \((x_i, y_i)\) are treated as values of random variables \((X_i, Y_i)\) that are identically and independently distributed (i.i.d) with respect to a probability distribution \(P(x, y)\). The assumption of a joint probability distribution enables the modelling of uncertainty in predictions, i.e. due to noise in data, since \(y\) is not a deterministic function of \(x\). The overall goal of a supervised learning objective is thus to find a good hypothesis with respect to a \emph{loss} function, \(L: Y \times Y \rightarrow \mathbb{R}_{\geq 0}\), which is a measurement of how far \(f(x_i)\) lies from \(y_i\) i.e. \(L(y_i, \hat{y}_i)\). The \emph{risk} associated with the hypothesis is defined as an expectation of the loss function:

\[
R(f) := \int_{X \times Y} L(y, f(x)) dP(x, y)
\]

The smaller the risk the better the hypothesis. The challenge, however, is that an algorithm must minimize the risk without being able to evaluate \(P\) directly since it is unknown. Therefore, through the evaluation of the average loss over dataset \(S\), the \emph{empirical risk}:

\[
\hat{R}(f) := \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))
\]
the risk can be minimized approximately. This process is called empirical risk minimization (ERM) and is generally considered a computationally difficult task even for simple linear classifier functions. Machine learning algorithms deal with the associated computational complexity by applying convex optimization strategies or imposing assumptions about the distribution $P(x, y)$. Popular supervised machine learning algorithms include support vector machines, decision trees, random forests, naïve Bayes and artificial neural networks.

2.4.1.2 Unsupervised Learning

In direct contrast to supervised learning, unsupervised learning algorithms do not require labelled data sets, deriving patterns and structure from unlabeled data. The assumption made is that through mimicry, the algorithm is forced to generate a compact internal, latent representation of the input space $X$. Unsupervised algorithms are generally grouped into clustering and association problems. Clustering approaches deal with finding patterns and structure within unlabeled data to find natural groupings (clusters) of data points. Some clustering algorithms require the user to define the number of clusters to locate, while others can automatically make this decision based on a specified level of granularity. Association rule learning algorithms establish associations or rules describing large input space portions. For example, users of an online retailer who buy product $X$ are more likely to also buy product $Y$. These algorithms are thus of use when building recommendation systems or discovering anomalies in data (i.e. fraudulent transactions in banking). Dimensionality reduction is another example of unsupervised learning to reduce a high
dimensional input space into a lower-dimensional space while also minimizing information loss. Working with high dimensional data is difficult in many cases due to increased sparsity due to the curse of dimensionality and often computational intractability. Generally, dimensionality reduction is thus often used to facilitate visualisation of complex data, through reduction into two or three dimensions, but may also be a precursor to input into supervised algorithms.

2.4.1.3 Weakly-Supervised Learning

Supervised learning requires access to high-quality labelled datasets, often limiting application in real-world settings. The associated cost of generating data tends to manifest in a few ways: insufficient quantity of labelled data, insufficient subject expertise for labelling data, or insufficient time or resources to hand-label data. Weakly supervised learning approaches thus have the ability to learn from weaker or imprecise data labels without sacrificing predictive capability. Supervision comes in three forms: incomplete, inexact and inaccurate.

Incomplete supervision is used when a subset of the data is labelled correctly, yet there is not enough data to train a fully supervised model. Active learning is a form of incomplete supervision in which a human expert is used to annotate potentially informative or representative subsets of the data to minimize the cost associated with annotation while significantly improving model performance. Semi-supervised learning is another type of incomplete supervision that utilizes the unlabelled data to build a model more powerful than a supervised model using only the labelled portion. This type of learning may refer to
either transductive or inductive learning\textsuperscript{202}, where the goal of transductive learning is to infer the correct labels for unlabelled data, whereas the goal of inductive learning is to learn the optimal mapping from $X$ to $Y$.

Imprecise supervision involves using data with ‘weak’ or imprecise labels. Weak labels may take noisy measurements, heuristically determined classifications, pre-trained model classifications or judgements of similarity between pairs or within tuples of data. Multiple instance learning (MIL)\textsuperscript{203} is one form of imprecise supervision, in which bags (groups) of data are labelled according to the classification of a single instance (key instance) or the majority label within the bag. The task is to either infer the labels of single instances or the bags themselves. Metric learning (distance metric learning, similarity learning) may also be classed under the imprecise supervision umbrella and concerns constructing a task-specific distance metric from weakly labelled data. Learnt distance metrics may be used for downstream tasks such as classification, clustering and information retrieval.

The final form of weakly supervised learning, inaccurate supervision, concerns the use of low-quality labels where many examples may be incorrectly annotated. Data-editing approaches are widely used in this field, where models identify potentially mislabelled instances and either correct or remove the label. Identification of mislabeled nodes may be achieved through the use of relative neighbourhood graphs, however this restricts the approach to low-dimensional data due to sparsity\textsuperscript{201}. 
2.4.2 Deep Learning

Machine learning has seen an explosion of interest within the past decade, especially in computer vision and natural language processing (NLP), where machine learning-based approaches have set new gold standards for specific applications, such as image classification. Growing computational power is partially responsible for recent popularity, allowing the construction of models with increased degrees of freedom. The other significant driver of progression in this field has been the compilation of various large, annotated datasets containing domain-relevant information. The availability of large datasets has permitted the training of deep models, which constitute a separate distinction of machine learning known as deep learning. Deep learning models can learn complicated patterns through the hierarchical addition of simpler, modular building blocks. In theory, deeper models have a higher capacity for learning complex and more abstract patterns within data. The modular building blocks used are commonly realized by artificial neural networks (ANNs) or variations of this concept. In addition to supervised learning, recent research has focused on using deep learning for both unsupervised and weakly-supervised tasks.

2.4.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are a subset of machine learning inspired by the biological neural networks that constitute animal brains (Figure. 2.3). ANNs consist of an
assembly of interconnected units called neurons (perceptrons), where connections (synapses) exist to transmit data signals. Typically, neurons are organized in layers, with the output of the first layer of neurons being transmitted downstream (postsynaptic) into the next layer of neurons. Connections between neurons are associated with a weight multiplied by the input from the previous layer. Weights \( w_{i,j} \) are updated throughout the training procedure, which can either increase or decrease the strength of the signal sent downstream. All connections into a neuron are summed, and an activation function generates the neurons output. An optional bias term may also be associated with the neuron and is applied before the activation function:

\[
y_j = \varphi \left( \sum_i w_{i,j} \cdot x_i + b \right)
\]

The activation function is an important part of the function since it determines the strength of the signal passed to the next layer of neurons. Popular activation functions include: sigmoid, step functions and rectilinear functions such as the rectified linear unit (ReLU)\(^{204}\). In a classification context, a SoftMax activation function (logistic) is applied to the last layer constraining the outputs within the range \([0,1]\), which may also be interpreted as class probabilities. Using this simple scheme, ANNs can represent any Boolean function (AND, OR, XOR, etc.) and are considered universal approximators. Typical neural networks may have thousands to millions of neurons and connections, yet despite still containing several
of orders of magnitude less capacity than a human brain, ANNs have been able to outperform humans in many tasks.

Figure 2. 3 Illustrates a three-layer artificial neural network (ANN) with one hidden layer. The weights are defined for each connection between the neurons.
2.4.2.2 Gradient Decent

Gradient descent is the most popular method for tuning the weights in ANNs. It is a first-order, iterative optimization algorithm to find a local minimum for a differentiable function. Within each iteration, the algorithm takes steps in the opposite direction of the gradient at the current instance since this corresponds to the area of steepest descent. If we consider that the risk function has to be optimized as a function of the parameters of the neural network (weights, biases), and is an average of a given loss function over training data points, then the function that must be minimized is:

\[
f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i (w)
\]

Gradient descent works by selecting an initial input point, \( w_0 \in \mathbb{R}^N \) and a step size \( \gamma \) (learning rate) and moving along the sequence denoted by:

\[
w_{t+1} := w_t - \gamma \nabla f(w_t)
\]

Where \( \nabla f(w_t) \) is the gradient of the loss at \( w_t \). Note that the step size is allowed to change at each iteration to control convergence. A full iteration of training data is referred to as an
epoch and instead of updating parameters based on the full training data, updates are often performed on mini-batches consisting of \( k \) data points with the gradient averaged over the \( k \) points within the mini-batch. Using mini-batches during training allows an increase in learning rate since the gradient becomes less ‘noisy’, leading to faster convergence. A numerical computation of the gradient requires estimation of all partial derivatives and hence evaluation of the loss function \( N \) times making small changes to each parameter individually. Thus, before a single forward step can be taken in the direction of the negative gradient \( n \cdot N \) function evaluations must be performed which is not particularly efficient. To combat inefficiency the gradient is usually calculated analytically using the backpropagation method, requiring a single forward and backward pass.

2.4.3 Geometric Deep Learning

Geometric deep learning (GDL), or non-Euclidean deep learning\(^{205}\), is a nascent field that stems from the desire to apply deep neural network architectures, which have been hugely successful for image-based tasks, to geometric data such as: grids, groups, graphs and surfaces. Non-Euclidean specifically refers to the case where the shortest path between two points is not equivalent to the Euclidean distance between them. Note that while a triangle mesh representation would be considered non-Euclidean, a volumetric grid (voxels) is typically a Euclidean projection of a non-Euclidean domain. In a general sense GDL refers any approach that incorporates a geometric prior to improve the quality of a predictive model\(^{206}\).
Traditionally predictive tasks involving geometric data make use of handcrafted features or some form of empirical model which assumes expert knowledge, often failing to capture the full spectrum of information available in available data. For these reasons deep learning is expected to offer substantial benefit in the field due to its ability to learn from the data itself. However, despite the successes of deep-learning applied to uniform data, a paradigm shift for geometry processing has not yet been fully realised, however, the growing number of methods indicates that this may soon change.

2.4.3.1 Symmetries, invariants and equivariant

Within a given domain, a symmetry is a manipulation (transformation) in which a given property is unaffected (invariant), therefore this concept is fundamental to GDL. Transformations may be discrete, smooth or continuous such as those defined within the Euclidean group $E(3)$. To put the importance of symmetries into perspective one may consider that the rotational orientation of a molecule, or the ordering of its atoms (in terms of representation, not structural/relational changes) should not have an impact on the prediction of certain properties i.e. the prediction task requires both rotation and permutation invariance respectively.

Sticking with the example of molecular systems, 3D representations can be considered as entities in Euclidean space within which certain transformations (symmetry operations) can be applied. These transformations can be applied with respect to three different symmetries: line, point and plane, and are isometric (rigid), meaning that application of the transformation will not change the Euclidean distances between points.
The Euclidean group of transformations, \( E(3) \) corresponds to three transformations, and their arbitrary combinations: rotation, translation and reflection. Rotation refers to the transformation in which an object is rotated around an axis, by a certain number of degrees. A 3D shape may have an infinite number of rotation axes. Translation refers to the movement of all points along an axis by a given distance and reflection is the mapping of the space to itself through an inversion or mirroring. The special Euclidean group, \( SE(3) \), is a special case of the Euclidean group, \( E(3) \), where only rotation and translation is considered and the special orthogonal group, \( SO(3) \), considers only rotation. Molecular systems are symmetric with respect to the group \( SE(3) \), since rotations and translations do not impact properties such as solubility and atomic charges, however, molecules are not necessarily symmetric with respect to the group \( E(3) \) since chiral properties depend on the configuration of stereogenic centers thus are not reflection invariant.

Put in a more formal context, given a function \( f \), a domain \( \Omega \) and a group \( \mathfrak{G} \) where \( p \) is an \( n \)-dimensional real \textit{representation} of \( \mathfrak{G} \): \( p: \mathfrak{G} \to \mathbb{R}^{n \times n} \), the function \( f \) \( (f: X(\Omega) \to Y) \) is \( \mathfrak{G} \)-invariant if \( f(p(\mathfrak{g})x) = f(x) \) for all transformations in the group \( \mathfrak{g} \in \mathfrak{G} \) and \( x \in X(\Omega) \). Put simply, the output of the function is not changed by the application of the group transformation on the input space. Invariance is in fact a special case of the equivariance property where \( f \) \( (f: X(\Omega) \to X(\Omega')) \) is \( \mathfrak{G} \)-equivariant if \( f(p(\mathfrak{g})x) = p'(\mathfrak{g})f(x) \) for all transformations in the group \( \mathfrak{g} \in \mathfrak{G} \). In layman’s terms, the application of the group transformation on the input space affects the output space in the same manner.
2.4.4 Application of Machine Learning to Biological/Chemical problems

Given the power of modern machine learning and its outstanding ability to extract patterns from data, it is not surprising that such techniques for bio- and cheminformatic problems have much promise. Much of the progress within these domains has been achieved by directly transferring models designed for computer vision and natural language processing (NLP) tasks.

The convolutional neural network (CNN) is one of the models designed to process an image or grid-like data. This model has seen much success within microscopy and pathology, where it has been used to identify lung squamous cell carcinoma\textsuperscript{207} from pathology images and predict cardiac failure from endomyocardial biopsy samples with greater accuracy than two trained pathologists\textsuperscript{208}. Furthermore, the utilisation of pre-trained models on out of domain tasks (i.e. large-scale image classification) enables the application of these models to smaller in-domain datasets since useful cross-domain features may exist and can be tuned toward a new objective\textsuperscript{209}. Extending the CNN for processing 3D images is a simple process since convolutions can just as easily be applied on 3D grids due to the inherent Euclidean structure. Volumetric CNNs have been applied successfully in image-based classification tasks, including: emphysema detection in pulmonary 3D-CT images\textsuperscript{210}.

Due to the translation equivariance of convolutional filters, CNNs are also used for semantic segmentation tasks where the objective is to label individual pixels within an image or volume. The DeepMedic model proposed by Kamnitsas et al.\textsuperscript{211} used 3D CNNs in a semantic segmentation setting for automatic brain-lesion detection on brain scans,
achieving state-of-the-art results. The main limiting factor of 3D CNNs is the availability of large sets of annotated data, which are required since the number of parameters increases cubically with grid size and resolution. Therefore, it is common that high-resolution scans are down sampled using interpolation techniques to reduce computational cost at the expense of significant information loss. With high parameter spaces, the risk of overfitting is also increased, posing a critical issue regarding medical imaging since data sources are often noisy and samples are small. For application in a clinical setting, it is imperative that a model can generalise to unseen data and provide some form of explanation for a given prediction. Undoubtedly this will be the focus of further research within the field.

In addition to image-related models, natural language processing (NLP) models used for machine translation, text-to-speech and data mining have seen a wide adoption within bioinformatics being applicable to sequence data in the fields of genomics and proteomics. NLP models have been used for the retrieval of similar sequences\textsuperscript{212}, protein function classification\textsuperscript{213} among many others. Prediction of protein 3D structure from amino-acid sequence data has also been an area of intense focus, with the recent breakthrough, AlphaFold\textsuperscript{153}, demonstrating the power of deep learning within this domain. Outside of sequence data, NLP models have shown some success in cheminformatics, within which molecules are commonly represented in the form of a line notation (typographical notation using alphanumeric characters), such as the popular simplified molecular-input line-entry system (SMILES). Utilising NLP with typographical representations of molecules has been used for molecular feature extraction for use in various tasks including retrosynthetic pathway prediction\textsuperscript{214}. 
The amplified complexity of 3D structural data due to additional spatial dimensions has slowed progress inevitably within this domain relative to simpler representations. However, substantial developments have been made to utilise both small-molecule and protein structural information with GDL. With regard to small molecules (druglike), graph neural network (GNN) based approaches have seen the most success, considering that graphs are a natural model for molecular data. GNN based approaches have seen relative success in toxicity prediction\textsuperscript{215} and de-novo generation\textsuperscript{216}. Recent approaches, such as SchNet\textsuperscript{217}, have demonstrated the ability to predict quantum chemical properties with the same or similar accuracy to density functional theory (DFT) based approaches (the current gold standard).

Although GNNs have been leveraged for protein structure applications\textsuperscript{218} volumetric voxel grids in tandem with 3D CNNs remain the most commonly applied technique. Grids are generally constructed through sampling properties, such as atom type occupancies, within an atomic system at discrete intervals. Methods using 3D-CNNs have been applied to numerous tasks, including binding site prediction/inpainting\textsuperscript{219} and binding site classification\textsuperscript{196}. Despite the achievements made in this domain, the same limitations apply when applying 3D-CNNs to medical image analysis. In the field of computer vision, alternatives to 3D-CNNs are being actively researched, with the majority opting for point cloud\textsuperscript{220} based approaches.

In addition to graph and grid-based approaches, models utilising molecular surface representations have recently been proposed\textsuperscript{221}. Molecular surfaces model proteins as a continuous shape onto which geometric and physiochemical properties (e.g. electrostatics, hydrophobicity, polarizability, local curvature) can be projected. These properties are
critical in determining the interaction strength between macromolecules and have played a large role in the study of molecular recognition. Many have proposed the use of various handcrafted descriptors for capturing information about surface patterns, including: 3D Zernike descriptors (3DZD)\textsuperscript{124} and geometric fingerprints\textsuperscript{222}. Theoretically, an approach that can learn from data has a higher potential to capture domain-relevant patterns. Despite potential, the application of deep learning to surfaces has been relatively limited due to the additional complexity of working with surface representations. Surfaces are usually modelled either as triangulated mesh structures, point clouds or are discretised onto a uniform grid. A recent approach, MaSIF kickstarted the use of GDL on molecular surfaces, applying 2D geodesic convolutions to patches extracted from protein surface meshes to predict protein-protein interactions and binding site-ligand prediction\textsuperscript{221}. With the increase in attention on GDL, it is expected that many more applications involving molecular surfaces will be proposed in the near future.

2.4.5 Application of Machine Learning to Binding Site Comparison

Although machine learning has been applied to binding site related tasks, the application of machine learning to binding site comparison directly is relatively limited despite the potential benefits. While the supervised binding site classification methods proposed by Pu et al.\textsuperscript{196} and Gainza et al.\textsuperscript{221} implicitly learn binding site similarity to some extent, this similarity is within the context of a few ligand classes. It thus is not clear how these approaches will generalise to data not represented by any of the included classes. In the case of DeepDrug3D\textsuperscript{196}, the model learns to distinguish between heme, nucleotide and
control (diverse proteins) classes within the TOUGH-C1 dataset. Since the control class includes many diverse protein structures, there is no guarantee that relationships within the learnt feature space are meaningful in terms of functional similarity.

2.4.5.1 DeeplyTough

To our knowledge the only approach trained specifically for binding site comparison is the DeeplyTough model\textsuperscript{176}. The authors propose the problem as a metric learning objective where ground-truth similarity labels are provided by the TOUGH-M1\textsuperscript{123} dataset due to its size and balanced classes (positive, negative). Binding sites are modelled as voxel grids (4D tensors) with eight atomic feature channels $f: \mathbb{R}^3 \rightarrow \mathbb{R}^8$: occupancy (presence/absence), hydrophobic, aromatic, hydrogen bond donor, hydrogen bond acceptor, positive ionizable, negative ionizable and metallic, based on AutoDock4\textsuperscript{223} atom types. The occupancy of an atom is defined using a smooth indication of the respective van der Waals radius. The grid $p \in \mathbb{R}^{8\times24\times24\times24}$ is sampled from the protein structure with a step size 1.0Å and centered at point $\mu \in \mathbb{R}^3$ defined either by the centroid of a convex hull constructed from alpha spheres defined by Fpocket 2.0\textsuperscript{194} or from the convex hull centroid of residues within 8Å of ligand heavy atoms. The metric learning objective aims to learn a latent feature (embedding) such that similar inputs (binding site structures) correspond to latent features that lie close to each other in the learnt metric space while dissimilar inputs correspond to latent features that lie further apart by an arbitrarily defined margin $m > 0$.

Multiple loss functions exist for the metric learning objective, with most modern approaches using input triplets mined within a batch, consisting of a positive and negative
sample for a given input (anchor). This triplet generation process is typically informed by class labels, where an anchor is paired with an instance of the same class (positive) and an instance from another (negative). The problem with this paradigm in binding site datasets is that the ground-truth labels are primarily unknown. Simonovosky and Myers\textsuperscript{176} note that only 3,991 positive pairs within TOUGH-M1 can be used to build such triplets. The DeeplyTough model is thus trained using the pairwise margin loss that has been successfully used for image retrieval\textsuperscript{224}: 

\[
L(p_i, p_j) = \begin{cases} 
\|d_\theta(p_i) - d_\theta(p_j)\|_2^2, & Q \in \mathcal{P}, \\
\max (0, m - \|d_\theta(p_i) - d_\theta(p_j)\|_2^2), & Q \in \mathcal{N} 
\end{cases}
\]

Where in this case the function \(d_\theta\) is a 3D-CNN with \(\theta\) representing the models’ learnable parameters and \(m\) is an arbitrarily defined margin \(m > 0\) defining the distance at which a pair of inputs is considered dissimilar. The metric learnt is reflected by the Euclidean distance between elements within the latent space (i.e. metric space).

Since no canonical orientation of proteins, or indeed binding sites, exists, the learnt latent space should be invariant to rotations and maintain a certain degree of ‘fuzziness’ to account for discretisation artefacts and minor atom movements (the inclusion/exclusion of atoms towards the edge of the grid). To address nuisance transformation (any transformation, or series of transformations, that does not change the ground-truth label), the authors use recently proposed rotation equivariant, 3D steerable CNNs\textsuperscript{225}, where 3D
convolutional filters are parameterised as a linear combination of a complete steerable kernel basis. The other factors are addressed with data augmentations (random rotations and transformations) in tandem with a novel ‘stability loss’ since augmentation alone was not sufficient to achieve practical invariance.

The final model was evaluated on both the ProSPECTs\textsuperscript{175} and Vertex\textsuperscript{197} datasets displaying encouraging performance across a significant portion of the benchmarks. The authors suggest that the DeeplyTough\textsuperscript{176} model should be used as a universal tool for binding site comparison rather than a specialist one since it does not excel at any particular benchmark in comparison with other available tools and rather maintains consistency.

2.4.6 Proposal

As discussed in detail, binding site comparison is a remarkably complex task, requiring both a structured representation and a means to score the pairwise similarity between representations. We and others\textsuperscript{199,176} propose that the use of machine learning rather than handcrafted schemes has the potential to reduce bias associated with optimisation while also leveraging its pattern recognition abilities. GDL seems to be the perfect fit for this task since molecular structures can be represented as both non-Euclidean (topological graph structures) and Euclidean structures (arrangements of atoms in 3D space). However, the best representation for its application to protein structures is still unclear. While volumetric approaches are currently commonplace, they suffer from many limitations, often failing to capture important contextual information such as intramolecular contacts and important physiochemical information. Furthermore, 3D-CNNs are disadvantaged from a technical
standpoint, requiring fixed input sizes (cubic) which is inappropriate for binding site modelling since attention may be focused on inaccessible portions of the binding site (buried). They also suffer from sensitivity to nuisance transformations and are computationally inefficient due to high data sparsity (convolving over empty space) and high memory burden. This, in particular, also makes them prone to overfitting. While there are various solutions to some of these problems no one method has addressed all issues in one approach.

Given the inherent limitations with current approaches and the vast array of data representations compatible with GDL, we suggest that the exploration of other binding site representations, algorithms and supervision approaches will be highly beneficial to the field. In this work, two alternative methods are proposed focusing on different aspects of the problem. The first of these methods utilises recent advances in graph neural networks combined with a pharmacophoric representation of protein binding sites. Two paradigms are suggested within this approach: graph embedding and graph matching. The graph embedding approach takes the same metric learning approach as DeeplyTough\textsuperscript{176}, while the graph matching approach considers cross-graph relationships within the neural network architecture for joint reasoning. The second proposed method presents an unsupervised approach using molecular surface data in tandem with a computer vision technique, Bag-of-Features. The idea behind using an unsupervised approach is that removing the dependency for labelled ground-truths should aid in removing bias further while also learning useful representations that may be used for further downstream tasks. We expect that the proposed methods are not likely to be better for all conceived applications due to the context-dependent nature of binding-site comparison but should be considered
complementary approaches. One could envision an approach utilising multiple techniques to draw a final hypothesis.
3 Chapter 3: ToughGraph: Weakly Supervised Learning of Binding Site Representations Using Graph Neural Networks

3.1 Introduction

The use of graph structures and graph theoretic concepts has a rich history in chemical and biological informatics, although the application of graph neural networks (GNNs) to molecular data is relatively recent. While small molecules have received much attention in this regard, application to protein structures has seen a slower uptake. Despite this, it is unfair to say that GNNs have had no success in this domain. The PROTEINS and D&D\textsuperscript{226} datasets have become standardised benchmarks for novel GNN algorithms concerning the distinction between enzymes and non-enzymes. Further applications have been discussed in Section 2.4.4. The breadth of applications demonstrates the power of models based on graph-based representations for molecular tasks.

In light of recent advances in protein structural learning, we present a binding site similarity method using GNNs with the ability to identify binding-site similarities across protein families. We present models based on two paradigms: graph embedding and graph matching (Figure. 3.1). The prior follows a similar formalism to that of Simonovsky & Meyers\textsuperscript{176}, using a GNN to encode protein binding sites into fixed-length descriptor vectors, which can subsequently be used to search large sets of precomputed binding site vectors in an efficient manner using pairwise Euclidean distance. This approach is motivated by the need for fast comparisons across large datasets and takes inspiration from
binding site fingerprinting methods, such as RAPMAD\textsuperscript{165} and PocketMatch\textsuperscript{166}. The graph
matching paradigm extends this idea based on recent works attempting to learn a similarity
metric between pairs of graphs through joint reasoning, using GNNs that also consider
cross-graph relationships. One particular method presented by Li et al.\textsuperscript{227} realises this using
a cross-graph attention-based mechanism while demonstrating its effectiveness across
various graph-matching benchmarks. Attention is a commonly used technique in neural
network architectures, where a model can learn to attend more to important parts of input
data while reducing the influence of less important data. The relative importance of
different sections of the data is entirely context dependent and is learnt during model
optimisation. The cross-graph attention mechanism uses this concept to increase the
influence of messages propagated between pairs of nodes from each graph which are
deemed to be more important in the context of the global graph similarity. A formal
derivation is included in Section 3.2.2.2.

Since a mapping from a three-dimensional structure to a one-dimensional vector
will inevitably require an inevitable loss of information, it is expected that learning a metric
conditioned on a pair of graphs will be able to capture information relevant to the similarity
computation of the pair in question rather than focus on information that represents the
global metric space. This method represents a trade-off with the embedding model between
efficiency and accuracy, and we expect that each model will be useful for different
applications or complementary.
Figure 3. An illustration of the proposed graph models, adapted from Li et al. The graph embedding model computes a graph embedding without considering cross-graph messages (dotted lines), while the graph matching model (GMN) can jointly reason about similarity. In both examples, distances between the learnt latent feature vectors are considered to measure similarity.
3.1.1 Graph Data

A graph is formally defined as \( G = (V, E, f) \) where \( V \) corresponds to a set of vertices (nodes), \( E \) is a set of edges connecting nodes and \( f: E \to V^{<\omega} \) is a function assigning each edge to its set of vertices. Let \( v_i \in V \) denote a given vertex and \( e_{ij} = (v_i, v_j) \in E \) denote an edge connecting \( v_i \) to \( v_j \). The neighbourhood of a node is thus defined as \( N(v) = \{u \in V | (v, u) \in E\} \). A common representation of a graph is the adjacency matrix \( A \), a square \( n \times n \) matrix such that an element \( A_{ij} \) is equal to one if an edge exists between and \( v_i \) and \( v_j \) else \( A_{ij} = 0 \). In many applications once may consider an attributed graph, with properties annotated on either nodes or edges. Let \( X \in \mathbb{R}^{n \times d} \) be a vertex property (feature) matrix where \( x_v \in \mathbb{R}^d \) is the property vector of a given vertex. Thusly, edge properties may be defined by an edge property matrix \( X^e \in \mathbb{R}^{m \times c} \) where \( x^e_{vu} \in \mathbb{R}^d \) is the property vector of the edge \((v,u)\). Directed graphs occur where all edges are also associated with a direction. Undirected graphs are a special case of a directed graph where a pair of edges with inverse directions exist between any connected vertices. The undirected case is thus only true if the adjacency matrix is strictly symmetric. Another commonly used graph matrix is the graph Laplacian matrix defined as \( L = D - A \in \mathbb{R}^{n \times n} \) where \( D \) is the degree matrix. The symmetric normalized \( \tilde{L} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \) Laplacian is usually used instead of the Laplacian directly due to improved generalizability when used in machine learning.
3.1.2 Graph Similarity

Evaluating the similarity between pairs of graph-structured data has been researched extensively and is used in a wide range of applications, including: bioinformatics, cheminformatics, recommender systems and social network analysis. A common task is retrieving similar graphs from databases given a query. This similarity is typically defined as exact matches (full or sub-graph isomorphism) or a measure of structural similarity through the use of metrics like graph edit distance (GED) or maximum common subgraph (MCS). While effective structural similarity measures, these approaches are known to be NP-complete\textsuperscript{65}. Graph kernels are another method for computing graph similarity and are predominantly used in kernelized machine learning approaches for graph classification or regression. Graph kernels typically encode similarity through measuring structural relationships between graph elements such as: walks or paths, substructures or subtrees\textsuperscript{228}. While graph kernels are used within machine learning models, the kernel itself is usually ‘hand crafted’ motivating a wave of approaches leveraging the power of GNNs to learn a similarity metric from raw graph data.

3.1.3 Graph Neural Networks

Graph neural networks (GNNs) are a subset of GDL deep learning since the graph domain is inherently non-Euclidean. Graphs represent one of the most popular branches of GDL since many forms of data can be represented by graphs with complex relationships and interdependencies. The complexity in graph data comes from the fact that graphs are
characteristically irregular, with a variable number of vertices, edges, no vertex ordering and variable neighbourhood sizes. With respect to machine learning, further complexity exists since most algorithms assume that all instances are independent, while this assumption does not hold for graph-structured data. GNNs is thus an attempt to generalise machine learning to graph structures circumventing the aforementioned issues. GNNs can be separated into two broad streams: spectral and spatial.

Spectral approaches borrow from the field of graph signal processing\textsuperscript{229}, performing convolution in the graph Fourier domain. The name “spectral” refers to the fact that convolutional filters are defined with respect to the eigenvalues and eigenvectors of the graph Laplacian matrix. Due to the use of the Laplacian, spectral GNNs learn localized filters which are particularly sensitive to graph structure.

Spatial methods operate directly on the graph domain defining convolutions locally where updated features are computed based on a weighted aggregation of a features within a vertexes local neighbourhood. Spatial methods are often preferred to spectral methods since they are more efficient, generalizable and flexible. The lack of efficiency of spectral methods can be explained by the requirement to perform eigenvector computation or operate over the whole graph at once. This requirement limits the spectral methods to smaller graphs while spatial graphs can operate directly in the graph domain through information propagation schemes. Furthermore, spectral based methods often generalize poorly to unseen data given that any graph perturbations result in a change of eigenbasis. Spatial methods do not suffer in this respect operating locally where weight sharing is possible across different locations and graph structures. More formally a spatial GNN integrates the graph connectivity and vertex features through information propagation.
where updated vertex features are defined as $H = f(G)X$, where $G$ is normally represented using the adjacency matrix. The goal therefore is to effectively find an optimal vertex aggregation function $f(\cdot)$ that learns to aggregate a vertexes local neighbourhood in order to compute an updated vertex representation $H$. Note that the same principle maybe applied to update global graph properties or edge attributes and that edge/global attributes may also be used to inform aggregation functions.

Graph pooling is another important aspect in GNNs and is analogous to that used within CNNs. Pooling may refer to graph *coarsening* where the number of parameters is reduced through a down-sampling of nodes, thus reducing the potential for overfitting or to graph *readout* where a graph-level representation is computed based on the updated vertex representations and/or edges and global features. This final readout may be further processed and passed through a SoftMax function for classification tasks. For a comprehensive review of spectral, spatial and pooling operations, the reader is referred to\textsuperscript{230}.

### 3.1.4 Graph Matching Neural Networks

Using GNNs for the fast evaluation of graph similarity has recently received attention. The idea is that given a pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we wish to learn a function that can produce a similarity score $s(G_1, G_2)$ between them. The model may make use of both edge, node and global graph attributes when computing a pairwise score. For an effective comparison the network must be carefully designed to satisfy three fundamental properties: permutation invariance, generalizable and learnable meaning that any similarity metric can be learnt through parameter update during optimization.
The first approach to consider this paradigm to the best of our knowledge is SimGNN\textsuperscript{231}, where two strategies are proposed. The first strategy applies a Siamese style GNN to the input graphs, followed by a graph level pooling with a context-aware attention pooling operation which the authors suggest will learn representation with respect to the global context of the respective graph. After pooling, the graph-level features are processed using a neural-tensor network where interactions between the pair of features is considered. The second approach adds a pairwise vertex comparison layer where pairwise vertex feature similarities are aggregated into a histogram and appended to the pooled global feature. Irrespective of the applied strategy, a small MLP further processes the output, and a similarity score is computed. In their approach, the authors choose to learn GED as the similarity metric, demonstrating state-of-the-art results with respect to other GED approximation methods.

In another approach, Li et al.\textsuperscript{227} propose the graph matching network (GMN), where the similarity computation is based on a metric learning approach with weak labels based on whether a graph is considered similar or not. The authors compare a standard metric learning approach and an approach utilising a cross-graph attention module, adding a model's ability to reason about similarity jointly. The cross-graph layer measures how well a vertex in one graph can be matched to one or more nodes in the other. The authors demonstrate better accuracy when using the cross-graph messaging at the cost of efficiency.
3.1.5 Approach

Since, in the binding-site similarity task, we have no quantitative definition of similarity \textit{a priori}, the GMN model seems like a better fit for the problem since similarity is still defined in metric space. Correspondences may be interpreted as atomic or physiochemical similarities based on local structural regions learnt during information propagation. Of course, the assumption made is that these correspondences exist between binding sites binding to similar ligands. This general assumption is taken by most binding site comparison tools, especially those based on structural alignment. In our case, the learning process allows the model to determine what should define correspondence and how to utilise this information for defining a similarity metric.

3.2 Methodology

3.2.1 Modelling Protein Binding Sites

Similarly, to small molecules, proteins can be naturally represented as graph structures at various spatial resolutions. At the finest resolution, atomic-level representations represent the protein structure in a way analogous to the treatment of small molecules, where vertices represent atoms and the edges represent the relationships between these, either by chemical bonds or interactions based on radial distance cut-offs. Residue-level graphs can be constructed at a coarser level where vertices consist of individual amino acid residues and edges can be based on intramolecular interactions, chemical bonding, or radial distance
cut-offs. In some applications, a state between these levels is generated where nodes represent positions indicating the presence of certain physiochemical properties, dubbed pseudocenters\textsuperscript{167} (analogous to pharmacophores on small molecules). After constructing the graph structure, additional information can be incorporated by assigning numerical features to nodes and edges, representing chemical properties, atom/residue type, solvent accessibility values, secondary structure assignments, atomic charges, and geometric relationships (distances, angles).

After experimenting with residue-based graphs with limited success, we decided to use the pseudocenter approach as defined by CavBase\textsuperscript{167}. In this way, the set of atomic input positions is reduced while retaining more information than a residue-level structure, which neglects the type of interactions formed with a particular ligand. We reason that this scheme may also reduce the dependence on direct residue-residue correspondences due to a discrete encoding of residue type in vertex features. Pseudocenters are constructed from surface exposed residues and correspond to a set of five physiochemical properties essential for the formation of molecular interactions: hydrogen-bond donor (DON), acceptor (ACC), ambivalent donor/acceptor (HAC), hydrophobic/aliphatic (HYD) and aromatic (ARO).

3.2.1.1 Pseudocenter Construction

Pseudocenters are constructed from surface exposed residues and correspond to a set of five physiochemical properties essential for the formation of molecular interactions: hydrogen-bond donor (DON), acceptor (ACC), ambivalent donor/acceptor (HAC, e.g. side-chain nitrogen atoms in histidine ND1/NE2), hydrophobic/aliphatic (HYD) and
aromatic (ARO). For the full definition of each pseudocenter the reader is directed to the CavBase publication\textsuperscript{167} from which the definitions are used. The pseudocenters are constructed at positions within binding site residues defining locations representing 3D coordinates and a physicochemical feature. Binding site residues may be defined by a bound ligand or from a binding site prediction method. The concept is analogous to the pharmacophore concept. Each pseudocenter is also assigned two vector features, $v$ and $r$. The vector $v$ represents the average direction along which a putative interaction could be formed. The second vector $r$ represents the orientation of the interaction with respect to the cavity of the binding site region and is computed from the normalised summation vector, aggregated from all vectors originating from a pseudocenters 3D location towards all points on the binding sites molecular surface with a 3.0Å spherical region. In the original CavBase binding site comparison approach, binding sites are extracted using the LIGSITE\textsuperscript{232} prediction algorithm, a purely geometric approach for extracting depressed regions in molecular surfaces. The method discretises the protein into a 0.5Å resolution voxel grid performing scans from each axis of the cube and each cubic diagonal (seven scans in total) recording protein-solvent-protein events and incrementing the ranks of associated voxels. The higher the final rank the more buried the voxel. Binding sites may be extracted from continuous high-ranking regions. Since in our application pockets may be defined in multiple ways including from a bound ligand or from a prediction method, we skip the ranking step and take solvent grid points adjacent to protein occupied grid points as the definition of the molecular surface. Cavbase filters pseudocenters based on their exposure properties however, we retain the information and allow the network to make an informed decision. Pseudocenters are saved for further processing.
3.2.1.2 Feature Computation

Each pseudocenter is enriched with further information to aid with learning. Since vector and positional features are sensitive to nuisance transformation, features are processed in an invariant manner. With more recent graph neural network proposals future work may focus in using $SE(3)$ equivariant networks where the raw geometric features can be retained. To each pseudocenter a value describing accessible surface area defined by the number of adjacent molecular surface points and a description of the concavity/concavity of the associated surface region using the weighted PCA approach as described in\textsuperscript{180}. To represent each physiochemical classification in a machine learning appropriate manner each pseudocenter is transformed into a one-hot vector. For the DAC feature both donor and acceptor bits are set to one rather than being considered independently resulting in a for dimensional binary vector. All vertex features are subject to normalisation before training.

3.2.1.3 Graph Construction

Given a pseudocenter model we require a graph representation $G = (V, E)$ where the node set $V = \{v_1, v_2, ..., v_n\}$ represents each pseudocenter within the binding site definition. Each node is attributed with the aforementioned features, a unit vector describing a pseudocenters relative orientation to the binding site surface $r$, a vector $v$ representing the direction of a features projection and a vector describing its orientation to the geometric
center of the binding site definition. The geometric center is defined as the center of the convex-hull constructed from defined binding site atoms/residues. The edge set is defined as \( E = \{e_{i,j}, ..., e_{n(n-1)}\} \), where vertices are connected to their eight nearest neighbours. Initially we experimented with connecting vertices within a specified radius, however vertices were often left isolated due to a non-uniform sampling of centers, and at higher radial thresholds the large number of edges dramatically increased computation time and memory usage due to the convolution operator chosen. Future work may experiment with more efficient operators to enable higher edge densities. Constructed edges are annotated with the unit vector representing the direction between two edges \( u \) and the distance normalised using the exponential function \( f(x) = e^{-x^2/\sigma^2} \) where \( \sigma \) is optimized as a hyperparameter. In this manner higher values are assigned to vertices which are closer together. Once defined vector features are transformed into angular features to retain transformation invariance: \( \angle(u,v), \angle(u,r), \angle(v,r) \). The final graph is a vertex and edge attributed graph capturing information relevant for binding site comparison in terms of both shape and chemical features.

3.2.2 Model

3.2.2.1 Graph Embedding

The objective of the graph embedding model is to learn a latent representation that can capture the structural and physiochemical similarity of binding site pairs. This is achieved by mapping each graph to a latent representation independently, where the vector space of
structurally similar sites is closer than that of structurally dissimilar sites. One key advantage of this approach is that graph embeddings can pre-computed and indexed, ahead of time, enabling fast nearest-neighbour retrieval. Our GNN model consists of three main parts: (1) an encoder, (2) a number of propagation layers, and (3) an aggregation function.

The encoder network computes a mapping from vertex and edge features, assigning each node and edge an initial hidden feature vector. We experimented with using learnable embeddings for each pseudocenter type and concatenating this to the rest of the features, yet simply applying separate multilayer perceptrons (MLPs) to the raw vertex, and edge features performed equally well. The two encoder MLPs can formally be defined following the notation used by Li et al.:

\[
h_{i}^{(0)} = MLP_{vertex}(x_{i}), \quad \forall i \in V \\
e_{ij} = MLP_{edge}(x_{i,j}), \quad \forall (i, j) \in E
\]

The hidden features are subsequently propagated using a spatial message passing function. The propagation updates vertex hidden representations \(\{h_{i}^{(t)}\}_{i \in V} \rightarrow \{h_{i}^{(t+1)}\}_{i \in V}\) using the functions:

\[
m_{j \rightarrow i} = f_{message}(h_{i}^{(t)}, h_{j}^{(t)}, e_{i,j})
\]
The messaging and node are typically represented by one of the many graph neural network layers available. We use the continuous kernel-based convolution\textsuperscript{233}, originally used for predicting the quantum properties of small molecules. The convolution is defined as:

\[
h_i^{(t+1)} = f_{\text{vertex}} \left( h_i^{(t)}, \sum_{j: (j,i) \in E} m_{j \rightarrow i} \right)
\]

Where \( \Theta \) is a weight matrix and \( f_\Theta \) is a small MLP mapping the edge vector \( e_{i,j} \) to a \( d \times d \) matrix. The messages are aggregated over a vertexes local neighbourhood using a simple average across the updated features. The application of multiple \( (t) \) graph convolution layers enables the propagation of features from more distant vertices. A global aggregation function is required to compute a feature vector representation of the entire graph. The aggregation is performed over node representations \( \{ h_i^{(T)} \} \). Simple functions such as the sum, average or max can be used although we found a minor, but not insignificant, performance benefit from using an attention-based pooling mechanism\textsuperscript{234}:
The function maps node representations to a global feature vector using a weighted sum of
node features transformed with an MLP $f_\theta$. Weights are computed from the same hidden
node features using a small neural network $f_{\text{gate}}$. The act of the weighting process allows
the network to learn which features are the most important for the similarity task. The final
feature vector $h_G$ is L2 normalised and can subsequently be compared to other graph
feature vectors using the Euclidean distance to quantify similarity.

3.2.2.2 Graph Matching

The graph messaging function is implemented analogously in the GMN network, computing a matching vector that measures how well a vertex in one graph can be matched with the other. The process is formally defined by the following:

$$m_{j \rightarrow i} = f_{\text{message}}(h_i^{(t)}, h_j^{(t)}, e_{ij}), \forall (i, j) \in E_1 \cup E_2$$

$$\mu_{j \rightarrow i} = f_{\text{match}}(h_i^{(t)}, h_j^{(t)}), \forall i \in V_1, j \in V_2, \text{or } i \in V_2, j \in V_1$$
\[ h_i^{(t+1)} = f_{\text{node}} \left( h_i^{(t)}, \sum_j m_{j\rightarrow i}, \sum_{j'} \mu_{j'\rightarrow i} \right) \]

Where \( f_{\text{match}} \) is a function that communicates cross-graph information, using the attention-based module:

\[ a_{j\rightarrow i} = \frac{\exp \left( s_h(h_i^{(t)}, h_j^{(t)}) \right)}{\sum_{j'} \exp \left( s_h(h_i^{(t)}, h_{j'}^{(t)}) \right)} \]

\[ \mu_{j\rightarrow i} = a_{j\rightarrow i}(h_i^{(t)} - h_j^{(t)}) \]

Where \( a_{j\rightarrow i} \) is the attention value between node \( j \) in the second graph and node \( i \) in the first. Such attention weights are calculated across every pair of nodes across the pair. \( s_h \) represents a similarity function, where we use the pairwise dot-product of the current node embeddings due to its simplicity. The interaction of all the nodes \( j \in V_2 \) with the node \( i \) in \( V_1 \) is then given by:

\[ \sum_j \mu_{j\rightarrow i} = \sum_j a_{j\rightarrow i} (h_i^{(t)} - h_j^{(t)}) = h_i^{(t)} - \sum_j a_{j\rightarrow i} h_j^{(t)} \]
Intuitively $\sum_{j} \mu_{j \rightarrow i}$ measures the difference between $h^{(t)}_{i}$ and its closest neighbour in the other graph. The pairwise attention mechanism increases computational complexity but results in stronger structural bonds between the two graphs. Note that the function $f_{\text{match}}$ depends on the whole set $\{h^{(t)}_{j}\}$ this was omitted for the sake of brevity. Following the propagation layers the global graph descriptor is computed using the same aggregation module as the graph embedding module.

### 3.2.3 Training Dataset

Both models are trained on the TOUGH-M1 dataset with the same data splitting strategy proposed by Simonovosky and Meyers\textsuperscript{176}. In brief, to prevent data leakage between training and validation folds caused by protein sequence or fold homology, protein entries are assigned to sequence clusters at 30%, where entries within the same cluster may only be allocated to either the training or validation set. Any protein which could not be assigned a sequence cluster was removed. The splitting method was evaluated using global structural alignments, which showed that only a limited amount of structurally similar proteins were distributed between the folds\textsuperscript{176}. Before training, all features are precomputed for faster training.
3.2.4 Training Procedure and Parameters

For a fair comparison with DeeplyTough\textsuperscript{176} we follow a similar training procedure, utilising the aforementioned splitting strategy in tandem with a ten-fold cross-validation protocol using a group-shuffle splitting strategy where train and test splits are generated randomly according to the pre-computed protein sequence clusters. Using this technique means that proteins within the same sequence cluster can never be a member of both the train and test set, allowing a better informed evaluation of model performance. Validation metrics are averaged over the ten repeats and the standard error is recorded. We follow both DeeplyTough and Govindaraj and Brylinski\textsuperscript{123} reporting ROC metrics including the AUC. Precision recall analysis is also provided. Hyperparameters including the number of graph convolutions, the dimension of hidden features, the output dimension, batch size and learning rate were manually optimized. The same pairwise contrastive loss function is used for both models with a margin $m = 1.0$ and is optimised using a variant of stochastic gradient descent, Adam\textsuperscript{235} with the batch size learning rate, weight decay, and $\beta1$ and $\beta2$ hyperparameters set to 32, 0.0005, 0.0, 0.9 and 0.999 respectively. We find that a low learning rate is required for stability of the learning. Each epoch is limited to 25,000 pairs to further reduce overfitting. To aid with convergence a learning rate decay scheme is used where the learning rate is decayed by $\lambda=0.1$ on plateau to a minimum learning rate of 0.00001. The network takes approximately one hour to train on an NVIDIA\textsuperscript{®} GeForce\textsuperscript{®} RTX 2080 Ti GPU.

To further aid with network generalisability and stability, a series of augmentations are applied to input training graphs. The first augmentation jitters the vertex positions using
random noise sampled from a gaussian distribution. The center of the distribution is set to zero- with a width of 0.2Å to mimic minor atom movements. The binding site center is also transformed by a random value between [-2Å, 2Å], altering distance and angle features which are constructed on-the-fly. The final augmentation removes up to 10% of vertices in the graph randomly. We found that 10% was the highest value that could be used before training destabilised, likely due to the removal of too much important contextual information. Augmentations force the model to learn a more robust metric space where minor variations have less impact on the similarity between sites in proteins with identical sequence.

3.3 Results and Discussion

3.3.1 Training Results

We compare both the graph embedding model and the graph matching model with previously reported results published by the TOUGH-M1 authors\textsuperscript{123} using the tools: APoC\textsuperscript{236}, G-LoSA\textsuperscript{185} and SiteEngine\textsuperscript{172}. All three methods are alignment-dependent. We also compare with the results obtained by the DeeplyTough 3D-CNN model. To keep consistency with previously reported results, we follow the evaluation procedure outlined by the authors of DeeplyTough, following the sequence-based splitting protocol described in Section 3.2.3. for all algorithms ROC and PR curves are shown in Figure 3.2. Note that our results differ slightly to the original publication, since the random number seed used for splitting in DeeplyTough’s evaluation protocol is unknown.
Figure 3. 2 ROC (a) and PR (b) plots with associated AUC and AP values for the evaluation of the performance of binding site comparison algorithms on validation folds of TOUGH-M1. The standard error is computed from ten random folds.

All machine learning based approaches outperform the more traditional based approaches, ToughGraph (embedding) performs slightly worse in terms of AUC (0.85) in comparison to the other approaches, DeeplyTough (0.91) and ToughGraph (matching) (0.93). The matching model significantly outperforms the embedding model, in our tests, potentially suggesting that the ability to jointly reason about similarity strengthens the model’s ability to make judgments based on the attending to potential correspondences. The embedding approach suffers in terms of precision-recall with an average precision (0.71) similar to SiteEngine (0.71). Although, we suggest that this could be due to the stronger reasoning ability of the graph matching networks, there is also the possibility that an extensive hyper-parameter search could reveal a better-quality embedding, closing the existing performance gap.
The lower average precision relative to DeeplyTough could possibly be attributed to the amount of information lost when constructing a graph with a lower than atomic resolution. It is also likely that the nature of convolution over voxel grids, facilitate a stronger ability to capture shape-related features compared to a graph where geometric relationships are difficult to capture. A future approach may decide to make use of the recent wave of $SE(3)$ equivariant graph neural networks for a stronger ability to reason geometrically. The matching approach does not suffer from the loss of resolution, outperforming every other approach tested. The extra performance may therefore be a direct consequence of the cross-graph messaging. Furthermore, ToughGraph (matching) is more stable across folds compared with the embedding model, performing almost identically to DeeplyTough which is the most stable model across the testing folds.

Simonovsky and Myers$^{176}$ identified that false negatives may highlight questionable ground truths in the data, including biologically endogenous models which bind promiscuously to many heterogenous biological targets. These molecules include nucleotides, amino acid monomers, sugars, buffers and stabilisers used in crystallography. We also found this to be the case with our models. This highlights a fundamental problem with a mass collection of data based on simple rules (in this case, shared binding of similar molecules). Promiscuous ligands such as nucleotides and non-specific drug molecules such as the broad-spectrum kinase inhibitor, quercetin, which has also demonstrated binding to SARS-Cov 3CL protease$^{237}$, present an issue with this form of data collection where shared binding may be due to the chemical properties of the ligand rather than a similarity between the two binding sites. Indeed, many have proposed that conformational flexibility is a possible mediator of promiscuity with the ability to adopt diverse 3D positions within
different binding sites. Labelling two binding sites on the basis of the shared binding of a promiscuous ligand is likely to generate noise due to limited structural homology within these examples. We further identified further edge cases where ligands were only partially bound within a binding site region or problems concerning binding sites located between two protein chains, while only one is retained by the definition. No clear pattern was identifiable from the inspection of false positive binding site pairs. The matter is particularly complex, since a lack of evidence of binding does not rule out potential shared binding. False positives should ideally be verified experimentally, although this would inevitably be a costly endeavour.

The predominance of nucleotides (ATP, ADP, FAD, NAD) within TOUGH-M1 (33%) is another factor to be aware of, reflecting the saturation of biologically endogenous molecules with the PDB. The clear bias within the data is a factor that should be addressed in the future. The model we propose in Chapter 3, aims to rectify this problem by using an unsupervised approach removing any dependence on label and thus the associated biases. Despite the abundance of nucleotides in the dataset, they are unlikely to bias performance estimations due to the heterogeneous nature of nucleotide binding. Kahraman et al.\textsuperscript{190} observed that the shape of nucleotides is significantly less conserved compared to more rigid molecules such as heme and steroids. It is therefore more likely that these ‘difficult’ examples could destabilise training, leading to issues with generalisability outside of the datasets distribution.

While we observe a strong performance of the proposed models, particularly the matching model, it is necessary to validate with out of distribution datasets to verify that the model can generalize to unseen data. This is especially true considering the uncovered
biases and questionable labels within the dataset. In the case of the matching model, generalizability is a slightly larger concern, since the cross-graph attention mechanism could learn to reflect biases existing in the underlying data distribution. The models were further validated on two independent datasets: ProSPECCTs\textsuperscript{175} and Vertex\textsuperscript{197} described in Sections 2.3.1 and 2.3.4

### 3.3.2 ProSPECCTs Benchmark

The ProSPECCTs benchmark dataset, as described in Section 2.3.1, is a collection of ten independent subsets designed to test different aspects of binding site similarity, identifying strengths and weaknesses. In contrast with the TOUGH-M1 dataset, binding sites are defined by a bound ligand rather than a prediction algorithm and hence we construct ToughGraph inputs based of binding sites extracted from all residues within radius $R = 8\text{\AA}$ from any ligand heavy atom. This difference offers an interesting opportunity to examine how the model responds to different binding site representations. To prevent the overestimation of performance due to leakage between the TOUGH-M1 and ProSPECCTs sets we remain consistent with the suggested splitting strategy\textsuperscript{176} removing any structure from the training set if it shares 30\% sequence identity with any protein within ProSPECCTs. Simonovosky and Myer\textsuperscript{176}, reduced the parameter space of DeeplyTough in response to the reduction in training data (4862 proteins, 401,266 binding site pairs remaining) however we found no negative impact to retaining the same architecture.
Results are compared with both DeeplyTough and those published alongside the ProSPECCTs database\textsuperscript{175}, with benchmarking data available for 21 different binding site comparison algorithms. AUC is used as a performance metric (Table 3.1) in-line with both publications. Since precision-recall is a better way to analyse the results of a retrieval task in the case of imbalanced data the average precision (AP) scores are also provided and compared with those reported by DeeplyTough (Table 3.1).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1</th>
<th>P1.2</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P5.2</th>
<th>P6</th>
<th>P6.2</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeeplyTough</strong></td>
<td>0.95</td>
<td>0.98</td>
<td>0.90</td>
<td>0.76</td>
<td>0.75</td>
<td>0.67</td>
<td>0.63</td>
<td>0.54</td>
<td>0.54</td>
<td>0.83</td>
</tr>
<tr>
<td><strong>ToughGraph (embed)</strong></td>
<td>0.95</td>
<td>0.98</td>
<td>0.96</td>
<td>0.64</td>
<td>0.54</td>
<td>0.61</td>
<td>0.59</td>
<td>0.56</td>
<td>0.56</td>
<td>0.84</td>
</tr>
<tr>
<td><strong>ToughGraph (match)</strong></td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
<td>0.76</td>
<td>0.70</td>
<td>0.67</td>
<td>0.65</td>
<td>0.46</td>
<td>0.46</td>
<td>0.81</td>
</tr>
<tr>
<td>Rank / 24</td>
<td>4</td>
<td>1-7</td>
<td>7</td>
<td>3-5</td>
<td>9</td>
<td>6-7</td>
<td>4-5</td>
<td>9-11</td>
<td>15</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 3.1 AUC values for the ToughGraph models on each of the 10 ProSPECCTs data subsets. The rank is computed based on results presented in two studies\textsuperscript{175,176}, evaluating 24 tools across the ProSPECCTs dataset. The rank is computed from the best result achieved by either ToughGraph model.
Table 3. 2 Average precision values for the ToughGraph models on each of the 10 ProSPECCTs data subsets. The results for DeeplyTough are used directly from the original publication\textsuperscript{176}.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1</th>
<th>P1.2</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P5.2</th>
<th>P6</th>
<th>P6.2</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeeplyTough</strong></td>
<td>0.78</td>
<td>0.90</td>
<td>0.57</td>
<td>0.76</td>
<td>0.77</td>
<td>0.33</td>
<td>0.28</td>
<td>0.37</td>
<td>0.37</td>
<td>0.45</td>
</tr>
<tr>
<td><strong>ToughGraph (embed)</strong></td>
<td>0.84</td>
<td>0.92</td>
<td>0.80</td>
<td>0.75</td>
<td>0.54</td>
<td>0.29</td>
<td>0.25</td>
<td>0.44</td>
<td>0.44</td>
<td>0.46</td>
</tr>
<tr>
<td><strong>ToughGraph (match)</strong></td>
<td>1.00</td>
<td>0.95</td>
<td>0.93</td>
<td>0.80</td>
<td>0.76</td>
<td>0.35</td>
<td>0.29</td>
<td>0.27</td>
<td>0.27</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Across the first two datasets (P1/.2), designed to test sensitivity to binding site definition both DeeplyTough and ToughGraph (embedding) performed comparably, with ToughGraph displaying slightly higher average precision. When compared with other methods both DeeplyTough and the embedding model rank 10\textsuperscript{th} indicating a susceptibility to the binding site definition. In an ideal situation an algorithm should have no trouble identifying sites within proteins with identical sequence. ToughGraph (matching) outperforms both models by a large margin ranking within the top five algorithms for both subsets. The approach is much less sensitive to binding site definition, presumably due to its ability to locate simple correspondences between the overlapping regions between two representations. The performance gain is in-line with the intuition that the model should have a stronger reasoning ability when allowed to access both members of a pair as opposed to reasoning in terms of a learnt global metric space. The intuition extends to dataset P2 with the matching model again outperforming both the embedding and DeeplyTough
models when it comes to distinguishing between members within an NMR ensemble from members within another ensemble. The result also demonstrates tolerance with respect to some level of flexibility. Intriguingly the ToughGraph (embedding) model is also more robust to flexibility than DeeplyTough. Perhaps this is due to small movements altering the center from which the binding site is sampled altering occupancies in constructed the voxel grid. In terms of ToughGraph, a graph structure should be less susceptible to this type of artifact with the only change being minor variations to the lengths associated with edge relationships.

The decoy datasets P3 and P4, measure the discrimination ability of algorithms by introducing five artificial mutations into identical binding sites. ToughGraph (matching) performs well on dataset P3 matching the performance of DeeplyTough, however the performance decreases for dataset P4 where shape-based mutations are also introduced. This performance drop is likely due to a graph neural networks lesser ability to reason about geometric relationships, whereas a voxel grid implicitly represents the surface shape of a molecular structure. The matching is likely less affected, again, due to its stronger matching ability, identifying the mutated positions.

The next four subsets (P5/.2, P6/.2) were compiled from two independent publications\textsuperscript{190,191}. The first two contains proteins binding to similar ligands and cofactors (excluding and including PO$_4$). Results are fairly comparable across the board with a slight preference for the matching algorithm, in terms of rank all methods rank within the top 25% with mediocre scores being in line with the observation made by the author limited shape homology exists between sites of flexible molecules. Methods that rank in the top ranks all include some form of size dependent scoring function penalizing sites with
different sizes. This is consistent with the observation that size alone is able to differentiate binding sites in this set given the large size differences ATP and FAD for example. Performance across the P6 subsets is essentially random. For the matching model we predicted this outcome considering the set was developed based on three observations of interaction patterns. The first describes the binding of the same ligand by different residues with similar interactions while the second describes binding of the same ligand with different interactions and the last describes the interaction of the same ligand with different interactions and the engagement of different chemotypes\textsuperscript{191}. Sites with no obvious correspondences does not fit well within the matching models paradigm.

The last set (P7) is a more realistic retrieval experiment in which the task is to retrieve identified similarities extracted from various literature sources. All methods displayed similar performance ranking in the top half of all tools benchmarked by Ehrt et al.\textsuperscript{175} Initial benchmarking identified ToughGraph (matching) as the most performant algorithm for binding site matching however it is important to acknowledge that this comes at a performance cost from a computation standpoint. The method requires inference for every pairwise comparison, whereas embedding based methods can precompute latent descriptors for rapid nearest neighbour retrievals. One could perhaps envision a pipeline in which a fast nearest neighbour approach is used to calculate an initial ranking with a re-ranking computed with the more robust model. The results also suggest that a matching based paradigm may not be the best choice for discovering remote similarity or sites with no obvious correspondences, although such examples are hard for any method to retrieve without sacrificing precision/specificity.
3.3.3 Vertex Benchmark

ToughGraph is finally benchmarked using the Vertex dataset (Section 2.3.4) The benchmarks objective is to identify similar proteins from pocket-based similarities. For each pairwise protein comparison, an all-by-all comparison of all identified binding sites is computed, and the minimum distance (or maximum similarity) used to compute a ranking. As with ProSPECCTs binding sites are defined using bound ligands. The model was retrained with the same procedure outlined in Section 3.3.2 removing homologous proteins between training and validation sets. The AUC and AP are shown in Table 3.3.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AUC</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeeplyTough</td>
<td>0.830</td>
<td>0.986</td>
</tr>
<tr>
<td>SiteHopper</td>
<td>0.887</td>
<td>0.992</td>
</tr>
<tr>
<td>TM-align</td>
<td>0.767</td>
<td>0.983</td>
</tr>
<tr>
<td>Pocket-Match</td>
<td>0.604</td>
<td>0.961</td>
</tr>
<tr>
<td>ToughGraph (embed)</td>
<td>0.756</td>
<td>0.980</td>
</tr>
<tr>
<td>ToughGraph (match)</td>
<td>0.860</td>
<td>0.989</td>
</tr>
</tbody>
</table>

Table 3. 3 AUC and AP values for evaluating ToughGraph and other binding site comparison algorithms on the vertex dataset. Data for other tools is used directly from\(^\text{176}\). ToughGraph (matching) achieved a performance (AUC 0.860) between that of DeeplyTough (AUC 0.830) and SiteHopper (AUC 0.887) while ToughGraph (embedding) (AUC 0.756) trails TM-Align (AUC 0.767) only out performing Pocket-Match (0.604)
which performs the worst out of the tools tested. The results again highlight the benefit of introducing the matching function, enabling the model to jointly reason over a pair of inputs. The method performs only second to SiteHopper\textsuperscript{170}, a state-of-the-art commercial tool which performs alignments based on protein feature points represented by spherical gaussians. Despite performance close to that of SiteHopper, ToughGraph (matching) runs orders of magnitude faster, although is still much slower that the embedding based model and DeeplyTough, since latent features can be precomputed prior to matching. All methods achieve a high average precision, however, since the Vertex has a large class imbalance, in favour of positive pairs (1,461,668 positive and 102,935 negative pairs), the average precision metric overestimates performance since it ignores true negative labels.

3.3.4 Case Study – Estrogen Receptors

Drug repurposing involves using approved drugs for an indication other than its original intended purpose\textsuperscript{238}. Target-based repurposing achieves this through target-hopping, where a molecule displays affinity towards an off-target. These off-target interactions are possible since molecules usually interact with many targets within a biological system\textsuperscript{34}. This phenomenon is known as polypharmacology, or drug promiscuity, i.e. the more targets a molecule interacts with, the more promiscuous it is considered to be. The driving factor behind promiscuity is the topic of a long-standing debate, with many contradictory results reported within the literature. A systematic study of drug promiscuity based on structural data in the PDB discovered weak correlations between chemical properties of small molecules and promiscuity, yet identified that over 71\% of the analysed drugs share at least
two targets with similar binding sites\textsuperscript{34}. The results suggest that local structural similarities between related and un-related proteins may be partially responsible for promiscuity. Our binding site similarity model presents a platform with which this assertion may be verified.

In this case study we investigate the drug molecule 4-hydroxy-tamoxifen (OHT) which is known to interact with ten pharmacologically relevant targets (information from Drugbank, accessed 01/01/22) including the estrogen receptors (ER), alpha and beta (ER$\alpha$, ER$\beta$) as well as the estrogen-related receptor gamma (ERR$\gamma$). Estrogen receptors play a significant role in various cancers, most notably in breast cancers with 70\% classed as ER-positive\textsuperscript{239}. Estrogen receptor inhibition has therefore been a key breast-cancer treatment strategy. All three targets have representative entries in the sc-PDB database\textsuperscript{192} (Table 3.4) and with at least one of those in complex with OHT. We used one entry for each protein: ER$\alpha$ (3ERT), ER$\beta$ (1X7B) and ERR$\gamma$ (1SQ9) each in complex with OHT as a query, performing a virtual screen against the sc-PDB. The ToughGraph (matching) model was used to perform the virtual screen using the Euclidean distance between binding site pairs for constructing a ranking (Figure 3.3). For each query we also computed an iterative ROC-AUC score at three distance thresholds ($D \leq 0.03$, $D \leq 0.035$, $D \leq 0.04$) where distances above the threshold were arbitrarily set to $D = 1.0$ (Tables 3.6-3.7). Only proteins with an AUC above 0.75 at each threshold were considered for further analysis. We acknowledge a similar experiment performed by Schalon et al.\textsuperscript{168} with which we qualitatively compare our results.
Table 3. 4 The number of estrogen and estrogen-related receptors included in the sc-PDB database.

<table>
<thead>
<tr>
<th>Protein name</th>
<th>Number of entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estrogen receptor alpha</td>
<td>60</td>
</tr>
<tr>
<td>Estrogen receptor beta</td>
<td>26</td>
</tr>
<tr>
<td>Estrogen-related receptor gamma</td>
<td>6</td>
</tr>
</tbody>
</table>

Using ERα as a reference, 101 hits were recovered below a distance threshold of 0.04 of which 31 are unique proteins and the majority are singletons. 40 of the 60 ERα structures, 14 of the ERβ structures and 5 of the 6 ERRγ structures are recovered. Despite the large distance distinction between the first rank and the remaining ranks we do not observe a large distinction between estrogen receptor entries and other non-estrogen methods as was observed by Schalon et al.\textsuperscript{168} using an alignment-based approach. We attribute this to the relative loss of information that occurs when compressing the information about a complex 3D structure into a vector of 256 elements. Despite this the method is confident in the retrieval of all estrogen receptors with iterative AUCs at $D \leq 0.04$ all $\geq 0.80$. It is also worth noting that while our method was able to retrieve the majority of ERRγ structures in the top ranks, the method used by Schalon failed to retrieve any at all. This is however in contrast with the screening from the ERβ in which we failed to retrieve any of the ERRγ structures while their method retrieves all entries. This odd observation highlights the general sensitivity of binding site comparisons and ideally one
would use the judgments made by numerous methods to make definitive conclusions. Our results for ERβ are also strange in that the model is more confident of similarities between ERβ, and two other nuclear hormone receptors: thyroid hormone receptors (THR) and retinoic acid receptors (RAR) than with itself. In the top 166 hits 16 of the 26 ERβ structures and 19 of the 60 ERα structures were retrieved. When changing the query structure to another ERβ entry (2FSZ) we found a better enrichment of ERβ, demonstrating that our approach has some inherent sensitivity to binding site definition. The final screen with ERRγ recovered 4 of the 6 ERRγ structures, 45 of the 60 ERα and 12 of the 26 ERβ structures. Despite not always achieving the highest AUCs all screens recover mainly estrogen receptors in the top ranks $D \leq 0.03$.

A closer look at significantly enriched sites among all the screens (Tables 3.5-3.7) revealed that the majority of identifications correspond to members of the nuclear hormone receptor superfamily and in all cases both ERα and ERβ were identified at or below a distance threshold of 0.04. A particularly interesting result is the recovery of the androgen receptor (AR) with the ERβ query, since this receptor is also a known target of OHT$^{240}$. It is also worth noting that ARs were also recovered in some of the highest ranks for the other screens despite a lower confidence (AUC). Schalon et al.$^{168}$ also identify similarities with HIV-1 and -2 protease and p38 MAP kinase 14 (MAPK14) and while we did not identify this with confidence HIV-1 and -2 protease did appear in the top ranks $D \leq 0.45$ for both ERα and ERRγ. MAPK14 was also identified at rank 141 in the ERβ suggesting a potential similarity between these sites although to our knowledge no experimental evidence exists to confirm this relationship.
Figure 3. 3 Virtual screening results for three estrogen receptors (ER\(\alpha\), ER\(\beta\) and ERR\(\gamma\)) bound to 4-hydroxy-tamoxifen. The top hits are ranked according to the distance between their binding site feature vectors computed with a graph embedding model. For plot (a) the query is a ER\(\alpha\) (3ERT), (b) is ER\(\beta\) (1X7B) and (c) is ERR\(\gamma\) (1S9Q). Only entries with a distance to the query below 0.04 are displayed.
### Table 3. 5 Virtual screening results using an estrogen receptor alpha binding site (3ERT) against the sc-PDB database.

<table>
<thead>
<tr>
<th>Protein name</th>
<th>$D&lt;=0.030$</th>
<th>$D&lt;=0.035$</th>
<th>$D&lt;=0.040$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estrogen-related receptor gamma</td>
<td>0.83</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>Estrogen receptor alpha</td>
<td>-</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>Peroxisome proliferator-activated receptor gamma</td>
<td>-</td>
<td>-</td>
<td>0.81</td>
</tr>
<tr>
<td>Bromodomain-containing protein 4</td>
<td>-</td>
<td>-</td>
<td>0.79</td>
</tr>
<tr>
<td>Estrogen receptor beta</td>
<td>-</td>
<td>-</td>
<td>0.80</td>
</tr>
<tr>
<td>Geranyl geranyl pyrophosphate synthase</td>
<td>-</td>
<td>-</td>
<td>0.77</td>
</tr>
<tr>
<td>Pteridine reductase</td>
<td>-</td>
<td>-</td>
<td>0.77</td>
</tr>
<tr>
<td>NH(3)-dependent NAD(+) synthetase</td>
<td>-</td>
<td>-</td>
<td>0.77</td>
</tr>
</tbody>
</table>

### Table 3. 6 Virtual screening results using an estrogen receptor beta binding site (1X7B) against the sc-PDB database.

<table>
<thead>
<tr>
<th>Protein name</th>
<th>$D&lt;=0.030$</th>
<th>$D&lt;=0.035$</th>
<th>$D&lt;=0.040$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thyroid hormone receptor beta</td>
<td>0.77</td>
<td>0.77</td>
<td>0.94</td>
</tr>
<tr>
<td>Retinoic acid receptor gamma</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Androgen receptor</td>
<td>-</td>
<td>0.77</td>
<td>0.91</td>
</tr>
<tr>
<td>Estrogen receptor beta</td>
<td>-</td>
<td>0.82</td>
<td>0.87</td>
</tr>
<tr>
<td>Glucocorticoid receptor</td>
<td>-</td>
<td>0.77</td>
<td>0.85</td>
</tr>
<tr>
<td>Steroid 17-alpha-hydroxylase/17,20 lyase</td>
<td>-</td>
<td>-</td>
<td>0.90</td>
</tr>
<tr>
<td>Retinoic acid receptor RXR-alpha</td>
<td>-</td>
<td>-</td>
<td>0.84</td>
</tr>
<tr>
<td>Estrogen receptor alpha</td>
<td>-</td>
<td>-</td>
<td>0.79</td>
</tr>
<tr>
<td>Mycocyclosin synthase</td>
<td>-</td>
<td>-</td>
<td>0.79</td>
</tr>
<tr>
<td>Aristochene synthase</td>
<td>-</td>
<td>-</td>
<td>0.76</td>
</tr>
<tr>
<td>Nuclear receptor ROR-gamma</td>
<td>-</td>
<td>-</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Table 3. 7 Virtual screening results using an estrogen-related receptor gamma binding site (1S9Q) against the sc-PDB database.

<table>
<thead>
<tr>
<th>Protein name</th>
<th>D&lt;=0.030</th>
<th>D&lt;=0.035</th>
<th>D&lt;=0.040</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estrogen-related receptor gamma</td>
<td></td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>Estrogen receptor alpha</td>
<td></td>
<td>0.76</td>
<td>0.86</td>
</tr>
<tr>
<td>Peroxisome proliferator-activated receptor gamma</td>
<td></td>
<td></td>
<td>0.89</td>
</tr>
<tr>
<td>Peroxisome proliferator-activated receptor alpha</td>
<td></td>
<td></td>
<td>0.82</td>
</tr>
<tr>
<td>Estrogen receptor beta</td>
<td></td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td>Bromodomain-containing protein 4</td>
<td></td>
<td></td>
<td>0.79</td>
</tr>
<tr>
<td>4,4’-diapophytoene synthase</td>
<td></td>
<td></td>
<td>0.76</td>
</tr>
</tbody>
</table>

The most interesting result we discovered is the recovery of both peroxisome proliferator-activated receptor gamma (PPARγ) and bromodomain-containing protein 4 (BRD4) for both the ERα and ERRγ screens. PPARγ is also recovered in the top rankings (rank 88) for the is ERβ screen although not with confidence. The result is particularly noteworthy for two reasons: (1) it was not discovered by Schalon et al.\textsuperscript{168} and (2) a recent publication reported an ‘unexpected’ similarity between PPARγ, BRD4 ERs and cyclin-dependent protein kinase 2 (CDK2)\textsuperscript{241}. The similarity was uncovered using an approach similar to the similarity ensemble approach (SEA)\textsuperscript{242}, although using scaffolds instead of Tanimoto similarities. The authors identified a flavone-like privileged scaffold (Figure 3.4) which is part of the drug molecules flavopiridol and genistein (Figure. 3.4) of which the former is a known inhibitor of CDK2 and the latter ERβ and PPARγ. Flavopiridol (Figure. 3.4) is also found bound to BRD4 in the PDB entry 4O71. Furthermore, BRD4 is known
to interact with kinase inhibitors of which luteolin and quercetin share a similar flavone-like scaffold and are also partial agonists of PPARγ. The findings in the report suggest that some sort of putative similarity in these sites is likely to exist. To validate their findings the authors performed docking experiments and \textit{in vitro} testing, using molecules similar to known ligands for each target against BRD4. The highest hit rate was obtained for PPARγ ligands followed by CDK2 and ERβ. The observation validates the predictions made by our matching model and its ability to find similarities between binding sites of unrelated proteins, an essential quality required for the prediction of off-targets and for the identification of drug-repurposing candidates.

![Figure 3. 4 A flavone-like scaffold, and two drug molecules: Flavopiridol and Genistein containing it. The former is a known inhibitor of CDK2 and the latter ERβ and PPARγ.](image)

Since ligands of PPARγ have the highest hit rate versus BRD4 a sanity check was performed using PPARγ entries to perform a virtual screen of the sc-PDB. For four out of
the five entries tested (3H0A, 3V9V, 3VJH, 1I7I) and retrieved BRD4 within the top ranks with iterative AUC scores $> 0.8$ at $D \leq 0.04$.

### 3.4 Conclusions

In this study, we proposed the ToughGraph GNN models for quantifying the similarity between pairs of binding sites. The method utilises a graph representation of binding sites inspired by recent advances in the application of GNNs to small molecules, protein structural data, and more traditional graph-based approaches. The proposed model is significantly faster traditional graph-based approaches while also demonstrating competitive performance and outperforming other recently proposed deep-learning based approaches. Our main contribution however is the use of a graph-matching neural network paradigm incorporating a cross-graph messaging function for joint-reasoning ability. The addition of the matching function significantly increases performance over the baseline embedding model, almost matching the performance of an alignment-based state-of-the-art approach, SiteHopper$^{170}$, when evaluating on a benchmark dataset informed by activity labels. To demonstrate the utility of the model a case study was performed in which the model was able to retrieve a recently discovered, and validated, relationship between estrogen receptor, peroxisome proliferator receptor $\gamma$ and bromodomain containing protein 4$^{241}$. We expect that the matching based model may be utilised alongside other binding site comparison methods for the prediction of off-target interactions and the identification of target based repositioning candidates.
It is important to note however that training on the TOUGH-M1\textsuperscript{243} dataset may potentially be problematic due to a significant bias toward nucleotide binding sites and questionable ground truth labels. This leads us to believe that the next logical step is to explore approaches, utilising different forms of supervision. In the next chapter (Section 4) we develop an unsupervised approach utilising learnt rotation-invariant local descriptors, coupled with a computer vision inspired approach for 3D shape retrieval.

Further research may focus on improving the binding site representation, using additional features or different levels of resolution. More recent advances in equivariant graph convolutions\textsuperscript{244} may also be worth exploring, operating on raw geometric relationships as opposed to engineered features such as angles. Another idea may also be to focus on explainability, creating mechanisms to understand why a model has made a particular judgement, an increasingly important field of research within the field of machine learning.
Chapter 4: SurfsUp: Unsupervised Learning of Rotation Invariant Local Surface Descriptors for Binding Site Comparison

4.1 Introduction

Predicting protein-ligand interactions from structural data is a fundamental problem in biology. A straightforward approach is to investigate evolutionary similarities between proteins utilising global sequence and structural comparisons. However, the major issue is that global homology does not indicate shared binding since interactions are mediated in smaller local depressions in the molecular surface known as binding sites. The molecular surface is simply a high-level representation of protein structure, commonly used for visualisation, modelling the structure as a continuous shape. In addition to visualisation, the molecular surface may be annotated with physiochemical properties and has long been the preferred method for describing electrostatic and hydrophobic interactions\textsuperscript{245}. Representing and comparing binding sites as surfaces is beneficial since explicit amino acid correspondences are not required to formulate a similarity score. Several proposals have been made utilising molecular surface data for binding site comparison, including: spherical harmonic decompositions (SHD)\textsuperscript{246}, 3D Zernike descriptors (3DZD)\textsuperscript{124} and fingerprint-based methods\textsuperscript{222}. As previously mentioned, such ‘handcrafted’ descriptors introduce sources of bias and fail to capture the full complexity of available data.

Based on our previous results and observations (Section 3) and the difficulty in determining ground-truth similarity relationships between protein binding sites, we
propose an unsupervised approach, learning powerful rotation invariant local surface features using a state-of-the-art autoencoder ANN, called PPF-FoldNet\textsuperscript{247}. The hypothesis is that the molecular surface contains both shape and chemical information important for understanding molecular recognition and that similar binding sites will share similar surface patterns. The presented algorithm, SurfsUp, is paired with a bag-of-features (BoF) approach for fast similarity-based virtual screening or paired with a 3D registration approach (RANSAC) to compute binding site superpositions.

### 4.1.1 Autoencoders

An autoencoder is a special type of ANN that can learn efficient encodings of unlabelled data (unsupervised). A model is supervised by its ability to reconstruct an input from the encoding. Resultant encodings are often referred to as latent features.

Autoencoders consist of two main parts: an encoder that maps the input to a latent feature and a decoder that reconstructs the input from the latent feature. The smaller the dimension of the latent feature, the more the autoencoder is forced to make approximations maintaining only the most relevant patterns in the training data. Formally the encoder and decoder can be defined as:
\[ \phi: X \rightarrow \mathcal{F} \]
\[ \psi: \mathcal{F} \rightarrow X \]
\[ \phi, \psi = \arg\min_{\phi, \psi} \| X - (\psi \circ \phi)X \|^2 \]

The encoder processes the input \( x \in \mathbb{R}^d = X \) and produces the latent feature \( h \in \mathbb{R}^p = \mathcal{F} \).

Autoencoders are trained through minimisation of a reconstruction error acting as a form of ‘loss’ function. As with standard ANNs the model is optimised using gradient descent and backpropagation. At inference time, the encoder is decoupled from the decoder for downstream utilisation of the generated latent features.

### 4.1.2 Machine Learning on 3D Surfaces

Algorithms for the registration or retrieval of similar 3D objects, represented as triangular meshes or point clouds, have traditionally involved hand-crafted local descriptors, lacking in discriminative ability and generalization\(^{205}\). With the current wave GDL and the successes of learned descriptors in the 2D domain, it is a natural succession to apply deep learning methods to 3D data. The application of deep learning to 3D data is still an ongoing field of research and has not been solved by any means, however, pioneering works have been able to adapt ANN for triangulated mesh and point cloud inputs. Point-cloud based approaches are the most popular, with the seminal work PointNet\(^{220}\) demonstrating on-par or state-of-the-art performance on various benchmarks. Since publication, many works
have built upon the backbone proposed in this work. A recent publication applied local geodesic convolutions to extracted, triangulated molecular surface patches for supervised protein-protein interaction prediction and binding-site classification\cite{221}. To the best of our knowledge, the model, dubbed MaSIF, was the first attempt to apply deep learning to molecular surfaces, demonstrating impressive performance and inspiring our approach. Since the publication of MaSIF, limited progress has been made in this domain.

The application of deep learning for learning efficient local features from surfaces has suffered from a collection of issues: (1) the requirement for supervision and a large amount of labelled data, (2) sensitivity to nuisance transformations in or, (3) the requirement for significant input preparation, and (4) lack of generalization ability to unseen data. A recent approach, PPF-FoldNet\cite{247}, an unsupervised, invariant, sparse and efficient autoencoder, attempts to address these issues. The model operates directly on point clouds and accounts for point sparsity, out-performing many other models on registration tasks. The model maintains transformation invariance through point pair features (PPF) attributed to local surfaces ‘patches’. The model's objective is to reconstruct PPFs, learning an efficient and feature-rich latent feature. Multiple approaches have been proposed in subsequent articles, but all require at least a weak form of supervision.

4.1.3 Approach

Since ground truths are so difficult to establish and the observation that TOUGH-M1 may contain questionable ground truths, we hypothesize that an unsupervised method should be
a better approach to leverage the power of machine learning for extracting discriminative features from molecular surface data. We extend the PPF-FoldNet approach to include the chemical features, hydrophobicity and electrostatic potential. The novelty of this approach is the ability, when paired with a bag-of-features approach, for rapid screening of binding site similarities while also being able to use learnt features to construct superpositions. To our knowledge, no method works using unsupervised feature learning, with the ability to run in both alignment-free and dependent manners.

4.2 Methodology

4.2.1 Surface Preparation

For this work, we calculate the solvent excluded surface (SES), which represents the boundary of the molecular volume with respect to a solvent probe. The software MSMS (Maximal Speed Molecular Surface)\textsuperscript{248} is used for SES calculation, with a probe radius of 1.4Å and a triangulation density of 3 vertices per Å\textsuperscript{2}. The software outputs the surface as a triangulated mesh structure represented by arrays of vertices, faces, vertex normals and face normals. A basic clean-up process removes degenerate faces, duplicate faces, infinite values, and unreferenced vertices. Four iterations of Laplacian smoothing are also applied to remove noise from the surface generation process. Four iterations are enough for vertex positions to converge without sacrificing performance by iterating further.
4.2.2 Surface Patch Sampling

Local surface regions, or patches, are extracted from the triangulated SES at specified reference points $x_r$. Around selected reference points, a sphere with radius $R$ is placed, and subsequently, all surface vertices within the sphere are considered members of a local region. Since local regions may contain disconnected regions of molecular surface, a clustering procedure, clusters regions with connected triangles and only the largest connected region is retained. The resulting region is considered a patch, and triangle information is discarded, rendering the patch a 3D point cloud with vertices and vertex normals. For input into a neural network, the patches need to be of a consistent size. Vertices from the patch are randomly sampled if the patch contains more points than the input size. If a patch contains fewer points than the input size, all points are sampled, and random points are duplicated until the size is reached. The PPF-FoldNet architecture displays robustness to differences in sampling density\textsuperscript{247}, but since the surface is constructed using a specific density, the number of points within a patch does not differ by large amounts. In the following experiments a radius, $R$, of 5Å is chosen for patch creation, and 324 points are sampled per patch based on the average amount of points within constructed patches.

4.2.3 Local Patch Representation

The PPF-FoldNet\textsuperscript{247} deep neural autoencoder architecture handles invariance to SO(3) (six degrees of freedom) transformations through encoding the local geometry of surface
patches into point-pair features (PPFs) (Figure 4.1). Since the PPFs are intrinsically rotation invariant, the learned embedding space also inherits this desirable property. This method is fundamentally different from other patch-based approaches, such as MaSIF\textsuperscript{221}, which requires the network to be trained with multiple patch rotations applying a permutation invariant function over the outputs. The latter approach is not only an approximation but is also inefficient and destabilises the training process. PPF-FoldNet instead learns an auto-encoded representation, or codeword, to describe a local surface patch. The learnt representation may subsequently be used in downstream applications.

Given an oriented point cloud representation: $X = \{x_i \in \mathbb{R}^6\}$, where oriented means that each point is also adorned with a normal direction (e.g. tangent space), a local patch on the surface may be defined as: $\Omega_{x_r} \subset X$ where $x_r$ represents a central reference point. Central reference points defining a patch center may be defined in multiple different ways depending on application. Each sampled patch can subsequently be encoded into PPFs where features describing local geometry are computed between the reference point and all other points within the patch (i.e. within a radial threshold):

$$F_\Omega = \{f(x_r, x_1) ... f(x_r, x_i) ... f(x_r, x_N)\} \in \mathbb{R}^{4 \times (N-1)}, \ i \neq r$$

PPFs are then defined for all pairs, $f(x_r, x_i)$, where the mapping $f: \mathbb{R}^{12} \rightarrow \mathbb{R}^4$ decomposes a pair of oriented points into three angles and a distance:
\[ f: (x^r_I, x^f_I)^T \rightarrow (\angle(n_I, d), \angle(n_l, d), \angle(n_I, n_l), \|d\|_2)^T \]

where \( d \) is the vector between the reference point and the paired point and is simply computed as \( d = p_i - p_j \). \( \|\cdot\| \) refers to the Euclidean distance and an angle between two vectors is always computed as:

\[
\angle(v_i, v_j) = \tan^{-1}\left(\frac{\|v_i \times v_j\|}{v_i \cdot v_j}\right)
\]

This approach to computing the angles is used since it does not suffer from numerical accuracy when calculating small angles, and the result is guaranteed to fall within the range \([0, \pi]\). The PPF descriptor does not consider the raw Cartesian coordinates or vertex normals since these properties are not rotation invariant, i.e. application of a rotation would alter the descriptors values. To consider these raw features one would have to compute a canonical orientation of the patch from which the descriptor could then be computed, adding significant cost to the calculation. A visual representation of PPF computation is shown in Figure 4.1. One can imagine that PPF encodes a local patches geometry in a way that can distinguish concave and convex patches.
PPF features encode geometric properties of a local surface patch, yet as noted by Kahraman et al.\textsuperscript{190}, shape alone is not enough to facilitate ligand binding at the surface of a protein. Other factors including: electrostatics and hydrophobic effects are essential for molecular interaction. To this end, we propose an extension of the PPF features to include descriptions of both the aforementioned. The extended feature set is denoted PPF(EH) with $E$, $H$ denoting the electrostatic and hydrophobic contributions respectively. The electrostatic feature contribution consists of annotating each point, $x_i$ within a local patch with an electrostatic potential value (ESP) calculated as described in Section 4.2.4. The hydrophobic contribution is defined as the hydrophobicity of the closest amino acid residue.
according to a simple hydrophobicity scale. Further information is given in Section 4.2.5.

Both properties are precomputed over the whole surface and hence the extended features can be computed given a physiochemically annotated oriented point set: \( X = \{ x_i \in \mathbb{R}^8 \} \), and a patch region, \( \Omega_{x_r} \subset X \), as PPF(EH)s between point pairs defined using the central patch point \( x_r \) and its neighbours within a radial threshold \( R \) as:

\[
f: (x_r^T, x_i^T)^T \rightarrow (\angle(n_r, d), \angle(n_i, d), \angle(n_r, n_i), \|d\|_2, ESP(x_i), HYD(x_i))^T
\]

where \( ESP \) and \( HYD \) correspond to the electrostatic and hydrophobic terms respectively. The final feature sets for a given a local patch represent the input to the autoencoder which is described in section 4.2.6. The learnt embedding is expected to be informative of the local geometry and chemistry of a surface patch, capturing the most distinctive information from the high-dimensional input space \( N \times 6 \), where \( N \) is the number of points considered within a patch region. It is worth mentioning here that compared to MaSIF\(^{221}\), the PPF(EH) input representation is not only more compact but is orders of magnitude more efficient to compute since a polar coordinate system for each patch is not required.

### 4.2.4 Electrostatic Potential Calculation

Electrostatic potential (ESP) values were calculated at each vertex on the triangulated solvent excluded molecular surface (SES) using the following procedure. Firstly proteins
were stripped of heterogen atoms (HET) and protonated with the REDUCE software\textsuperscript{249}, which adds hydrogen atoms in a standardised geometry optimising the orientation of OH, SH, NH\textsuperscript{3+}, methionine methyl groups, asparagine and glutamine sidechain amide groups and the histidine imidazole. Once protonated, the PDB files were prepared for electrostatics calculation using PDB2PQR\textsuperscript{250}, which adds missing atoms, radii and partial charges to a protein structure generating a PQR file. We use the PARSE forcefield for partial charge calculation as selected by default. APBS (v3.0)\textsuperscript{251} was used to compute the Poisson-Boltzmann electrostatics from the PQR file using the auto-generated input file prepared by PDB2PQR. Finally, ESP values were interpolated at each vertex using Multivalue, provided within the APBS software suite. To reduce the effect of very large charge values, the ESP values were capped to ±30 and normalised within the range $[−1, 1]$.

### 4.2.5 Hydrophobicity Calculation

Each vertex on the triangulated solvent excluded molecular surface (SES) is assigned a scalar hydrophobicity value simply by assigning the corresponding Kyte-Doolittle hydrophobicity scale\textsuperscript{252}, shown in table 4.1, of the nearest amino acid residue. The Kyte-Doolittle scale is commonly used for the prediction of solvent exposed and transmembrane regions, hydrophobic regions are assigned a positive value whereas hydrophilic regions are assigned a negative value. The Kyte-Doolittle scale is commonly used in machine learning applications since values lie with a standardised range [-4.5,4.5], which lends itself well to normalising within the range $[−1, 1]$. After assignment of all vertices the values are
smoothed by averaging values within a two-ring neighbourhood, Figure. 4.2, to smoothen
the sharp boundaries created during nearest neighbour assignment.

<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>Kyte-Doolittle Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alanine</td>
<td>1.80</td>
</tr>
<tr>
<td>Cysteine</td>
<td>2.50</td>
</tr>
<tr>
<td>Aspartic Acid</td>
<td>-3.50</td>
</tr>
<tr>
<td>Glutamic Acid</td>
<td>-3.50</td>
</tr>
<tr>
<td>Phenylalanine</td>
<td>2.80</td>
</tr>
<tr>
<td>Glycine</td>
<td>-0.40</td>
</tr>
<tr>
<td>Histidine</td>
<td>-3.20</td>
</tr>
<tr>
<td>Isoleucine</td>
<td>4.50</td>
</tr>
<tr>
<td>Lysine</td>
<td>-3.90</td>
</tr>
<tr>
<td>Leucine</td>
<td>3.80</td>
</tr>
<tr>
<td>Methionine</td>
<td>1.90</td>
</tr>
<tr>
<td>Asparagine</td>
<td>-3.50</td>
</tr>
<tr>
<td>Proline</td>
<td>-1.60</td>
</tr>
<tr>
<td>Glutamine</td>
<td>-3.50</td>
</tr>
<tr>
<td>Arginine</td>
<td>-4.50</td>
</tr>
<tr>
<td>Serine</td>
<td>-0.80</td>
</tr>
<tr>
<td>Threonine</td>
<td>-0.70</td>
</tr>
<tr>
<td>Valine</td>
<td>4.20</td>
</tr>
<tr>
<td>Tryptophan</td>
<td>-0.90</td>
</tr>
<tr>
<td>Tyrosine</td>
<td>-1.30</td>
</tr>
</tbody>
</table>

Table 4. 1 The Kyte-Doolittle scale\textsuperscript{252} for defining amino acid hydrophobicity, originally intended for identifying both surface-exposed and transmembrane regions from a protein sequence. Positive values are associated with hydrophobic residues while negative values are associated with hydrophilic residues.
Figure 4. 2 Illustration depicting the two-ring neighbourhood of vertex $v$. The coloured triangles and their corresponding vertices represent the one-ring neighbourhood while the outer, non-coloured, triangles and their corresponding vertices represent the two-ring neighbourhood.

4.2.6 Autoencoder Architecture

The auto-encoder architecture used to learn a low-dimensional embedding for a given surface patch annotated with PPF(EH) features is based on the PPF-FoldNet model\textsuperscript{247}, comprising an encoder and a decoder module. PPF-FoldNet itself is inspired by PointNet\textsuperscript{220}, FoldingNet\textsuperscript{253} and PPFFNet\textsuperscript{254} and hence the model architecture has many similarities to all of these models. The major difference being that all of the other
mentioned models act on a point cloud as a whole in a supervised context. PPF-FoldNet however, operates on local regions in an unsupervised manner, generating latent low-dimensional descriptors which may be utilised for downstream tasks through decoupling the decoder network. The full architecture is visualised in Figure 4.3.

4.2.6.1 Encoder

The encoder module comprises PointNet\textsuperscript{220} like encoder with a three-layer point-wise multi-layer perceptron (MLP) followed by a max-pooling operation to form a global feature vector. The low-level features from each MLP layer are then concatenated with the max pooled global feature vector using skip-connections. The resultant features are passed to further two-layer MLP followed by another max-pooling operations resulting in the final latent representation, or codeword. All the MLP layers contain: a densely connected layer, followed by a batch normalisation\textsuperscript{255} and a ReLU\textsuperscript{204} activation function. All layers within the encoder are either unaffected by the input ordering or proven to be invariant to order permutations. This means that for any given patch and a random shuffling of the corresponding PPF(EH) features ($F_{\Omega}$) will result in an equivalent output. This property is particularly important since we want to be able to compare codewords meaningfully.
4.2.6.2 Decoder

The objective of the decoder is to reconstruct the input PPF(EH)s, $F_\Omega$, using the latent codeword representation, forcing the latent representation to retain the most distinctive information from a high-dimensional input space, for the purpose of computing the reconstruction. Inspired by the approach used in FoldingNet\textsuperscript{253} the decoder deforms an $M \times 2$ (2D) uniform grid informed by the codeword rather than using an approach based on interpolation or up-sampling. In this process each 2D coordinate of the uniform grid is concatenated with a copy of the latent codeword, which is subsequently fed into two consecutive five-layer MLPs, performing the “folding” operation. The first MLP begins the folding, deforming the grid which is again concatenated to the output of the first MLP and the second MLP completes the reconstruction of the six-dimensional feature space. Since the feature space is larger than the three dimensions considered by FoldingNet, a five-layer MLP is used instead of a three-layer for increased learning potential.

4.2.6.3 Loss Function

A loss function based on the distance between two extended point-pair feature sets of unequal cardinality is required for optimisation since the size of the grid $M$ is not guaranteed to be the same size as the input $N$, and feature correspondences are lost during the training process. The Chamfer metric is commonly used for this purpose:
\[ d(F, \hat{F}) = \phi \left( \frac{1}{|F|} \sum_{f \in F} \min_{\hat{f} \in \hat{F}} \| f - \hat{f} \|_2, \frac{1}{|\hat{F}|} \sum_{\hat{f} \in \hat{F}} \min_{f \in F} \| f - \hat{f} \|_2 \right) \]

where \( \phi \) is a permutation invariant function such as summation, maximum or average and \( \hat{f} \) is the reconstructed input. The Chamfer metric has become a de-facto metric for point cloud reconstruction objectives since it is efficient and simple to implement. Other potential metrics such as the Earth Movers Distance are difficult to implement (especially on graphics processing units), with iterative approaches for calculation being very approximate with no guarantee of convergence. In this work we use the Chamfer metric taking \( \phi \) as the average since it was the most stable in our experiments.

Since the Chamfer metric is difficult to interpret and sensitive to outliers\(^{256} \), the F-score is also reported to monitor the training progress. The F-score is a well-established metric in the point cloud reconstruction community evaluating the distance between an input and reconstructed point cloud as a harmonic mean between precision and recall. Precision represents the accuracy of a reconstruction from an input (ground-truth) through counting the percentage of points within a specified distance threshold to the input. Recall measures the reverse counting the percentage of points on the input within a distance threshold from the reconstruction. The percentage threshold may be controlled in order to control the strictness of the measure, thus the F-score measures the percentage of points that were reconstructed correctly within a specified tolerance. During training the F-score with thresholds ranging from 0.05% to 20% are continuously monitored along with the Chamfer metric to check for convergence of the model.
Figure 4. 3 Auto-encoder architecture for the unsupervised learning of transformation invariant local protein surface descriptors. The architecture resembles that of PPF-FoldNet\textsuperscript{247}, used for local point cloud feature learning which can be used in tasks such as point cloud alignment.
4.2.7 Training dataset

Constructing a training set for the auto-encoder requires the extraction of a large set of protein surface patches. The set must be large for learning a good latent representation and represent the whole “surface-space”, meaning the patches encompass the majority of possible geometries and physiochemical surface property (ESP, HYD) distributions. Given a balanced, representative set of patches, the model should learn to interpolate patches with property variations.

For construction of the training set we decide to use the TOUGH-M1 dataset (7,524 experimental structures)\textsuperscript{123}, representing a non-redundant, diverse and representative set of protein structures. The dataset is openly available and also contains pre-processed structures appropriate for use in molecular docking studies, hence no complex pre-processing needs to be done. For generating surface patches for training, 1,024 patches are sampled per-structure from pre-calculated, feature-annotated solvent excluded surfaces resulting in $1024 \times 7524 = 7,704,576$ surface patches with computed PPF(EH) features per patch vertex with respect to the sampled central vertex $x_r$. Surface patches are sampled uniformly from each protein structure using the Poisson disk sample set construction algorithm proposed by Yuskel\textsuperscript{257} and implemented in the geometry processing package Open3D\textsuperscript{258}. The algorithm ensures a uniform sampling of patches across the protein surface maximizing the chances of constructing a representative sampling of the surface-space. The value of 1,024 was arbitrarily selected based off visualising sample distributions across a small set of diverse proteins within the TOUGH-M1 dataset. The value of 1,024 was deemed to generate patches covering the majority of the protein surface without
considerable redundancy (overlap). We assume the resultant set of almost 8 million surface patches represents a representative set with enough information for the model to learn a powerful latent representation without a considerably increased memory burden. In principle with access to high-performance-computing (HPC) facilities, unrestricted by memory limitations one could sample patches at every single surface vertex for a more powerful latent representation.

In this work, patches were precomputed before training, however, future work may consider sampling patches and computing PPF(EH) on-the-fly, given only the precomputed SES. This would considerably reduce the memory burden of our approach and potentially enable a more high-throughput training protocol. Realising such an approach may require low-level programming and is thus out of the scope of this work which provides a proof-of-principle application rather than a highly optimised protocol.

4.2.8 Training Procedure and Parameters

For training, a sequence-based splitting approach proposed in\textsuperscript{176}, whereby protein structures sharing greater than 30% sequence identity are allocated to the same cluster, is used to minimise data leakage between training and testing folds. We acknowledge that this will not have as much of an effect on molecular surface data since overlap is inevitable due to potential surface similarities between unrelated proteins. During one epoch, protein structures within the training fold are iterated over, and a random patch is selected from each for training. The original architecture of PPF-FoldNet\textsuperscript{247} is maintained other than the
input, and output layers are modified to accommodate the extended features. The latent feature dimension is 512 elements. The network is trained for 2000 epochs with a stochastic gradient descent variant, Adam\textsuperscript{235}, with the batch size learning rate, weight decay, and $\beta_1$ and $\beta_2$ hyperparameters set to 32, 0.001, 0.0, 0.9 and 0.999 respectively. A learning rate decay scheme is used to aid with convergence where the learning rate is decayed by $\lambda=0.1$ on plateau to a minimum learning rate of 0.00001. The network takes approximately four hours to train on an NVIDIA\textsuperscript{®} GeForce\textsuperscript{®} RTX 2080 Ti GPU.

4.2.9 Global Descriptor Computation

While learnt local features defined at reference points across the molecular surface are useful in their own regard, it is much more efficient for many applications to consider some form of aggregation of the local features into a global feature representation. In the domain of binding site comparisons, collections of local descriptors are aggregated into binding site “fingerprints” or global descriptor vectors, i.e. both RAPMAD\textsuperscript{165} and KRIPO\textsuperscript{179} construct global descriptors as histograms computed from pairs of local feature points. The benefit of methods following this paradigm is that features may be precomputed and stored in large databases with a low memory burden. Databases can then be queried rapidly, computing hundreds or thousands of comparisons per second. A further benefit of global representations is that they add a certain level of 'fuzziness' to comparisons enabling the retrieval of potential off-targets missed by more rigorous alignment-based methods, often requiring bijective-correspondences to be computed. This increased 'fuzziness' often comes at the cost of retrieving a greater number of false positives. Although truly suggesting that
a target is a false positive requires experimental evidence, which is notably lacking within the public domain.

Motivated by fast, approximate binding site comparison techniques and advances within computer vision, namely non-rigid shape retrieval, we suggest computing a global descriptor based on the learnt local surface features. Local surface features are aggregated into a global histogram using a Bag of Features (BoF) approach commonly used in the computer vision community for both image (2D) and shape (3D) retrieval. The BoF descriptor itself is inspired by the Bag-of-Words approach, used in natural language processing as a feature for machine learning. The BoF process consists of three steps: local feature computation, 'codebook' computation and global feature computation. The steps taken in our method are further described in detail.

4.2.9.1 Selecting Surface Patches

Before the global feature computation, a subset of patch centers are selected to speed up computation. Ideally, these patch centers should be evenly distributed across the binding site surface and represent the most distinctive features. To realise this, we propose a simple algorithm starting with the extraction of the binding site. Where ligand information is available, this process is achieved by selecting all protein atoms within a given radius Å from all ligand heavy atoms. When ligand information is not available, atoms can be selected using a binding site prediction tool such as FPocket$^{194}$ or LIGSITE$^{232}$. The closest surface vertex is selected starting with either the closest or a randomly selected protein atom. If the distance between the atom and surface is greater than 4.0Å, the point is
discarded, and the algorithm moves on to the next protein atom. A patch is constructed from a selected surface vertex as described in Section 4.2.3, as long as the patch center does not lie within a geodesic distance of 3.0Å from any existing patch centers. This filter is lenient enough to allow some redundancy since patches are constructed with the confines of a 5.0Å sphere, yet prevents oversampling.

### 4.2.9.2 Bag-of-Features

Given a set of descriptor attributed patches, a global binding site descriptor can be computed through a process involving the quantization of each descriptor using a precomputed set of representative feature vectors, hereby referred to as a 'codebook'. A codebook of size is computed offline using an unsupervised learning approach such as k-means. In this work, a Mini-batch k-means algorithm is appointed to avoid convergence issues occurring when using traditional k-means with a large quantity of data (>2,000,000 sampled surface patches). The final codebook is formed through the selection of the computed cluster centroids. Given the computed codebook, computing a global descriptor involves comparing each descriptor attributed surface patch with the codebook and aggregating overall resultant encodings to form a feature histogram. The approach is illustrated in Figure 4.4.
Figure 4. Illustration of the Bag-of-Features (BoF) approach for the computation of a global descriptor from a collection of features sampled at local regions on a shape or molecular surface. (a) Sampled surface patches are clustered with in an offline unsupervised learning approach to compute a codebook: $v = \{v_1, v_2, v_3\}$. (b) Two different quantization schemes are used to map a sampled patch to the computed codebook.
Given the codebook $\nu$ and a set of descriptor attributed surface patches, for each patch center $x \in X$ and its associated descriptor $p(x)$, a feature distribution $\theta(x) = (\theta_1(x), ..., \theta_V(x))^T$ is defined with size $1 \times V$ using a SoftMax encoding scheme:

$$\theta_i(x) = c(x) \exp \left( \frac{\|p(x) - \nu_i\|^2}{2\sigma^2} \right)$$

where $c(x)$ is a normalisation constant ensuring that $\|\theta(x)\|_1 = 1$. The $i$th element $\theta_i$, can be interpreted as the probability of the patch $x$, being close in distance to a patch $\nu_i$ in the codebook $\nu$. The equation represents a “soft” quantization approach where a “hard” quantization can be achieved simply by setting $\sigma \approx 0$, resulting in $\theta_i(x) = 1$ and zero otherwise ($i$ is the index of the nearest codebook element $\nu_i$ in descriptor space) (Figure 4.4b). Integrating the feature distribution over all patch centers results in a single $V \times 1$ feature vector:

$$f(X) = \int_X \theta(x) d\mu(x)$$

Commonly referred to as a Bag of Features (BoF) descriptor\textsuperscript{259}. The full process is illustrated in Figure 4.4, and a visual example is shown in Figure 4.5. BoF descriptors can be compared using simple distance, similarity, or statistical metrics. After extensive
experimentation, the exponential chi-squared similarity\textsuperscript{260} was chosen due to efficiency and the best results on initial validation data. The exponential chi-squared similarity is defined as:

\[
X^2_{\text{BoF}}(X, Y) = \exp \left( -\frac{1}{\lambda} \sum_{n=1}^{N} \frac{(X_i - Y_i)^2}{(X_i + Y_i)} \right)
\]

Since the BoF features are essentially histograms chi-squared is a natural metric in this descriptor space. The scaling factor $\lambda$ is ignored ($\lambda = 1.0$) since we found that it made no difference to results. In this work we consider both hard and soft quantisation to evaluate performance, since one may perform better than the other for different datasets/applications. For soft quantisation $\sigma$ is set as a quarter of the mean Euclidean distance ($\text{pdist}$) between all codewords in the computed codebook.
Figure 4. 5 Illustration of Bag of Features (BoF) histograms for three protein-ligand binding sites. The BoF descriptor is constructed with a soft quantisation and a codebook consisting of 64 codewords for ease of visualisation. The first two binding sites (blue, 1e6wA and orange, 1xelA) bind the coenzyme nicotinamide adenine dinucleotide (NAD) and are assigned a chi-squared similarity of 0.86 despite a low sequence identity of 14.1%. Both sites display local structural similarities reflected by overlapping regions in the BoF descriptor. The remaining binding site (green, 2ohtA) binds an experimental aminopyridine-based inhibitor, IP6, at the amyloid precursor protein cleavage (APP) site. The binding site shows no obvious homology to those in either of the previous proteins, reflected by a low chi-squared similarity of 0.1 for both pairwise comparisons (6.1% and 10.1% sequence identity, respectively). When considering 1e6wA, one may consider the potential for off-target interactions with 1xelA, yet the probability of off-target interactions with 2ohtA is substantially lower.
4.2.9.3 Spatially Sensitive Bag-of-Features

Global descriptor computation, using the Bag of Features approach, provides an elegant solution for fast pairwise comparison and has displayed excellent results in 3D shape retrieval, especially in non-rigid shapes, where shapes under isometric deformations must be retrieved. Seddon et al.\textsuperscript{64} successfully applied the BoF approach using the per-point spectral wave kernel signature (WKS\textsuperscript{261}) to the alignment-free retrieval of molecular shapes. However, the issue with the BoF paradigm is that only the distribution of representative features is considered and not the relationship between these features. Considering this problem from a natural language perspective, it is quite clear that the context of a word within a sentence often depends on other words in close proximity; this is especially true in the case of a contranym. Sentences may, therefore, often contain similar words yet should not be considered similar due to their global context. Textual search engines will therefore also consider individual words and combinations or \textit{expressions}. Extending this concept to local patches on the molecular surface, one expects that, given two different proteins, multiple similar regions will exist. The differentiation between these two proteins must thus be attributed to the relationships between spatially close regions.

Bronstein and Bronstein\textsuperscript{259} proposed an extension of the BoF approach by considering not only the frequency of representative features but also the spatial relationships between them. The approach coined, \textit{Spatially Sensitive Bags of Features} (SS-BoF) follows the same initial procedure as BoF, differing in the integration approach, in which pairwise relationships are considered:
\[ F(x) = \int_{\mathbb{R}^2} \theta(x)\theta^T(y)K_t(x,y)d\mu(x)d\mu(y) \]

The integration results in the representation \( F \), a \( V \times V \) matrix which can be considered as a bag of features in a codebook of size \( V^2 \) comprising of pairs of representative features. In the hard quantization approach an entry \( F_{ij} \) would be large if instances of a pair of representative features \( i, j \) are spatially close to each other. The representation therefore represents both the proximity between representative features in descriptor space and on the surface \( X \). In the original proposal the authors define \( K_t \) as either the diffusion distance or the heat kernel of the Laplace-Beltrami operator (Laplacian). Since in this application we deal with local molecular surface regions defining a binding site, creating boundaries, a SoftMax transformation of the approximate geodesic distance between patch centers is used for \( K_t \) where we set, \( \sigma = 2.0 \text{Å} \). The approximate geodesic distance is calculated using the shortest path approach with the Dijkstra algorithm. It is important to use the geodesic distance for this purpose since a Euclidean distance measurement is not able to differentiate between a point on the opposite side of the binding site (i.e. across the cavities volume) with a point on the same side as the reference point. The SS-BoF descriptor is an extension of the original BoF approach and hence comparisons can be conducted with equivalent metrics.
Due to the nature of the BoF descriptors, it is likely that local features will often be shared between small and large binding sites. For example, ligands with similar pharmacophoric regions might bind to similar regions on the molecular surface. When looking for off-targets of a large ligand, retrieving small binding sites is not particularly useful since the smaller volume is unlikely to accommodate the larger ligand. Motivated by this observation, an optional size-dependent scoring factor is proposed in which the similarity between two descriptors is further penalized by the difference in the number of surface patches sampled on each binding site. The penalty is defined as:

\[ P(f(X_1), f(X_2)) = \frac{\min(n_{X_1}, n_{X_2})}{\max(n_{X_1}, n_{X_2})} \]

where \( f(X_1) \) and \( f(X_2) \) are feature vectors computed from sampled patch centers: \( X_1 \) and \( X_2 \) from two distinct protein surfaces. Two scalar values \( n_{X_1} \) and \( n_{X_2} \) represent the number of sampled patches on each protein surface respectively. The penalty is simply the percentage of patches from the smaller binding site with respect to the larger and is applied through multiplication with the similarity score. The larger the difference between the number of sampled patches on each surface the lower the percentage and hence the greater the penalty. For example, if both samples contained an equal number of patches the penalty would be \( P = 1.0 \), so the penalty has no effect. In the case \( n_{X_1} = 20 \) and \( n_{X_2} = 40 \) then...
\[ P = 0.5 \] and the similarity score will be halved. Note that this penalty is essentially one-sided in that the impact of the penalty is symmetric whether either descriptor is used as the query or reference. We did not consider an asymmetric version in this work but since asymmetric metrics such as Tversky are useful for the virtual screening of small molecules, a penalty considering asymmetry could be analysed in future work.

### 4.2.10 Binding Site Alignment

Until now, algorithms have been proposed for computing global descriptors of protein binding sites, which can be utilised for fast, approximate nearest neighbour searching. The problem with this type of approach is the lack of interpretability compared to an alignment-dependent approach, where the result may display key correspondences, be they residues directly or between physiochemical feature points. Furthermore, after a binding site alignment, the overlap of ligands in each site can also give insight into whether a scaffold-hop or drug repurposing may be possible. Indeed, this view is shared by Ehrt et al.\textsuperscript{175}, who states that: binding site similarity is only a reasonable explanation for shared ligand binding if corresponding ligands overlap in binding site superposition, suggesting a visualisation step may be crucial for many applications, including repurposing.

Within computer vision, the alignment of surfaces represented as point clouds is a common operation. The PPF-FoldNet architecture used in this work was initially built to generate rotation-invariant local features and construct subsequent alignments from various shapes. Validation of the PPF-FoldNet model demonstrated a strong ability to recover
ground-truth alignments, surpassing many other state-of-the-art approaches, especially under conditions in which the query shapes were subject to nuisance transformations\textsuperscript{247}. While unsupervised latent features are a relatively new extension for surface alignment, feature-based approaches are not new. The use of local descriptors including FPFH\textsuperscript{262} is widespread within the literature. More recently, this approach has been applied to molecular surface data with the approach coined SENSAAS\textsuperscript{263}, constructing small molecule surface alignments using coloured point cloud representations and FPFH descriptors. The colours used represent the physiochemical feature class of the nearest atom to each surface vertex.

Based on the presented knowledge we propose that the power of learnt latent descriptors may have the ability to produce binding site alignments with respect to the input features, i.e. geometry, electrostatic potential and hydrophobicity. Given the two input binding site surfaces represented as oriented, feature annotated point clouds: $P = \{p_i \in \mathbb{R}^8\}$ and $Q = \{q_i \in \mathbb{R}^8\}$ we require the discovery of a rigid transformation $T \in SE(3)$, resulting in a certain overlap. A non-linear function $g(\cdot)$ maps the input points into a latent feature space, where in this case the non-linear function is the trained autoencoder model. The latent feature for point $p_i$ is $g(p_i)$, and $g(P)$ calculates the complete set of latent features for the points in $P$. Given oriented points and their associated latent features, or codewords the rigid transformation is computed with the Random Sample Consensus (RANSAC) algorithm implemented in Open3D\textsuperscript{258}. The RANSAC algorithm is conceptually simple and robust on noisy data, not requiring a large percentage of ground-truth overlap between two, point sets which may often be the case when constructing binding site surface alignments. The algorithm begins by selecting three random points
from the source point cloud \( P \) and constructs a set of nearest neighbour (NN) correspondences, \( M \), in feature space (Euclidean distance between latent features) from point cloud \( Q \):

\[
M = \{ (p_i, q_i), g(p_i) = NN(g(q_i), g(P)), g(q_i) = NN(g(p_i), g(Q)) \}
\]

From point correspondences, \( M \), the transformation, \( T \), is found using the Kabsch algorithm. Over a set number of iterations the transformation resulting in the highest number of points within a distance threshold, between \( P \) and \( Q \) represents the optimal alignment. Points within this threshold are referred to as inliers, and the root-mean-square-error (RMSE) is measured for these points. We choose a threshold of 1.2Å, based on initial experiments with a small population of binding site pairs, although this parameter can be easily changed to tune the result obtained, a larger threshold will be less sensitive to small surface deformations. We found that thresholds lower than 0.8Å were often too sensitive to generate a reasonable alignment between similar binding site pairs which had relatively minor deviations in surface shape. Further conditions may also be applied to prune the search space, rejecting correspondence sets if aligned point clouds are further apart than a specified distance threshold or if their vertex normals are divergent by a specified threshold (radians). Following the RANSAC procedure the iterative closest point algorithm (ICP) is applied to optimize the final alignment. The ICP alignment does not require explicit correspondences, estimating the transformation parameters from each points nearest-neighbour. The alignment optimisation simply ‘tightens’ the final result.
Once the transformation, $T$, has been estimated, it can be applied to the protein structure and associated ligand(s). Validation of alignments is usually achieved through visualisation of the resultant superposition. Although, in the case where similar ligands are bound, the alignment could be validated through recovery of the ligand binding pose by measuring the root-mean-square-deviation. However, this form of validation may not always be applicable; for example, in the case that hydrophobic interactions predominate to facilitate molecular recognition, many acceptable binding site alignments may exist due to local areas of similarity and ligand overlap may not be a good indication of an optimal alignment.

A small set of experiments discovered that using this alignment procedure with a small set of sampled surface patches could not recover a good alignment due to point sparsity. Therefore, when computing a binding site alignment, latent features are computed using $g(\cdot)$ for patch regions sampled at every surface vertex within a binding site region. These surface vertices may be defined through distance to a bound ligand or, in the case of an Apo protein, from a predicted binding site, i.e. surface vertices falling within the predicted region.
4.3 Results and Discussion

4.3.1 Training results

The results of the training process are displayed in Figure 4.6. The loss curves demonstrate a stable learning process with a very good generalisation on the ‘unseen’ validation fold. The loss plateaus after approximately 50,000 optimisation steps, although the model continues learning up until approximately 180,000 steps. The loss of the validation data appears smoother in this figure since the loss is averaged over the entire dataset while the training loss is logged per mini-batch. The F-score of model, checkpointed at the point with the lowest validation loss, is also displayed, measuring the harmonic mean between precision and recall. High reconstruction rates at relatively low thresholds indicate a strong reconstruction ability of the trained model.
Figure 4. Training results for the PPF-FoldNet model trained on local regions of molecular surfaces. The top panel displays the loss (Chamfer distance) for both the training and validation folds over 200,000 optimization steps. The bottom panel displays the F-score at various thresholds computed over the validation fold with the trained model.
A visualisation of the models’ latent space is performed to test whether the learnt space is meaningful. The PaCMAP\textsuperscript{264} dimensionality reduction algorithm is applied to a subset of latent features (‘codewords’) extracted from PPF(EH) patches forming a two-dimensional embedding space. To each 2D coordinate, we annotate a colour based on the average chemical property value of the patch, as shown in Figure 4.7. Although a rough approximation, this simple annotation reveals insight into the learnt feature space. The figure coloured by the hydrophobicity feature displays a clear pattern with a left to right gradient from more hydrophilic (pink) to more hydrophobic (green). In line with expectations, a larger proportion of the surface consists of hydrophilic residues. A clear pattern is also displayed in the figure coloured by electrostatic potential, almost forming a bimodal-like distribution with positively charged (red) regions at the top left and right of the embedding space and negatively charged (blue) occupying the center. Although difficult to make concrete conclusions, it is clear the model has learnt some form of meaningful latent space.
Figure 4. 7 A PaCMAP\textsuperscript{264} dimensionality reduction calculated from encoded local regions (patches) on protein molecular surfaces. The left plot is coloured by the hydrophobicity property while the right is coloured by electrostatic potential.

In another experiment we calculate latent features for each vertex within a binding site region of the enzyme, carbonic anhydrase II, reduce the dimension to three using PCA, and map this to an RGB colour. The same process is applied to an identical protein from a different PDB structure (1A42, 1ZE8). The result is a qualitative justification of replicability. The coloured binding sites are displayed in Figure 4.8, displaying similar colours in similar regions of the two binding sites, despite small differences in the surface shape and definition due to different ligands (BZU, PIU). The subpocket coloured purple is occupied by a sulphonamide in both instances.
Figure 4.8 Visualisation of latent features upon the molecular surfaces of two protein-ligand binding sites of the enzyme carbonic anhydrase II: (a) 1A42, (b) 1ZE8. To compute colours, PCA is applied to the latent features to reduce the dimension to three, associating each vertex with an RGB colour. Similar regions are associated with similar colours.

### 4.3.3 Codebook Construction

For the following benchmarks a set of codebooks were pre-computed as per Section 4.2.9.2, with varying sizes ($V$): 64, 128, 256, 512, 1024, 2048 and 4096. All codebooks were constructed using L2 normalized latent features computed from surface patches extracted from a set of 1024 protein structures in the TOUGH-M1 dataset. In an attempt to remove potential sources of bias, the selected structures are not closer than 30% sequence identity to any structure within the ProSPECTs dataset. The inertia for each cluster size
in $V$ is shown in Figure 4.9 in the form of an elbow plot. The inertia falls rapidly until around $k=512$ where it begins to plateau. The result suggests that a codebook size of around this size is likely to be optimal.

Figure 4. 9 Elbow plot displaying the inertia for the various cluster sizes used to compute a BoF codebook. The inertia is simply the sum of squared distances from each sample within a cluster to its centroid, repeated for all clusters: $\sum_{i=1}^{N}(x_i - C_k)^2$.

4.3.4 Benchmarks

The SurfsUp method was benchmarked using the ProSPECCTs$^{175}$, TOUGH-C1$^{196}$ and Vertex$^{197}$ benchmarks. We chose to omit TOUGH-M1 here due to the questionable ground-
truth labelling. The optimal parameters for BoF generation were determined by measuring performance over the ProSPECCtS dataset with the best parameters selected for the remaining benchmarks. Both BoF and ss-BoF features are tested.

### 4.3.4.1 ProSPECCtS Benchmark

The SurfsUp method was validated on the ProSPECCtS dataset since it is currently the most robust benchmark set available. Each of the 10 subsets tests a different aspect of binding site similarity (Table 2.2). We attempted all SurfsUp parameter combinations including, the size of the BoF/ss-BoF descriptor vector ($V$) (determined by codebook size), the type of quantization used (hard or HQ, soft or SQ), whether the size dependent scaling factor is included in the scoring function and the distance threshold ($R$) determining the surface patches to consider. For the value of $V$ we tried: 64, 124, 248, 512, 1024 and 2048, reducing the range for the ss-BoF to 512 to prevent a large memory burden, and for $R$ we tried: 3.0, 4.0, 5.0 and 6.0Å . A grid-search was performed over all parameter combinations for each algorithm variant (SQ/HQ, BoF/ss-BoF) determining the best parameters using AUC as the determining performance metric. AUC was used to remain consistent with the benchmarking procedure used by Ehrt et al.\textsuperscript{175}.

The grid-search results for SurfsUp using a soft-quantization approach and no size dependent scaling is presented in Figure 4.10. From the results it appears that a radius of $R = 6$Å and a dimension of $V = 512$ is the optimal parameter set, that has (1) the most balanced performance across all the data subsets and (2) the best performance to size ratio. We repeated this procedure for all of the variations mentioned although the remaining
seven grid-search results are not shown for the sake of brevity. We observed across the results that a radial sampling of $R = 6.0 \, \text{Å}$ performs best in all cases, despite performance generally being higher with lower radial thresholds for P5/.2, P6/.2 and P7. In these 5 cases it is important to acknowledge that these particular sets are highly unbalanced. The P7 subset contains 115 positive pairs and 56,284 negative pairs, a 363x difference. For unbalanced subsets a better metric is the average precision (AP), focusing on the quantity of positive retrievals. The AP score for these 5 sets indicates that the best radius lies realistically between $R = 5.0 \, \text{Å}$ and $R = 6.0 \, \text{Å}$ (Figure 4.11). Concerning dimension $V$ the optimal performance was achieved at $V = 512$ SQ BoF, ss-BoF and HQ ss-BoF. HQ BoFs seemed to benefit from a higher dimension $V = 2048$. The best parameters for all variants were collated and applied to each data subset and compared with other machine learning base approaches: DeeplyTough$^{176}$ and ToughGraph (Tables 4.2, 4.3).
Figure 4. 10 Grid search results (AUC) for SurfsUp using a soft-quantization approach and no size dependent scaling, on each ProSPECTs subset. The y-axis is the radius $R_{Å}$ and the x-axis is the size of the codebook $V$. From left to right, top to bottom the dataset subsets are: P1, P1.2, P2, P3, P4, P5, P5.2, P6, P6.2 and P7.
Figure 4. 11 Grid search results (AP) for SurfsUp using a soft-quantization approach and no size dependent scaling, on each ProSPECCTs subset. The y-axis is the radius $RÅ$ and the x-axis is the size of the codebook $V$. From left to right, top to bottom the dataset subsets are: P1, P1.2, P2, P3, P4, P5, P5.2, P6, P6.2 and P7.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1</th>
<th>P1.2</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P5.2</th>
<th>P6</th>
<th>P6.2</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeeplyTough</td>
<td>0.95</td>
<td>0.98</td>
<td>0.90</td>
<td>0.76</td>
<td>0.75</td>
<td>0.67</td>
<td>0.63</td>
<td>0.54</td>
<td>0.54</td>
<td>0.83</td>
</tr>
<tr>
<td>ToughGraph (embed)</td>
<td>0.95</td>
<td>0.98</td>
<td>0.96</td>
<td>0.64</td>
<td>0.54</td>
<td>0.61</td>
<td>0.59</td>
<td>0.56</td>
<td>0.60</td>
<td>0.84</td>
</tr>
<tr>
<td>ToughGraph (match)</td>
<td><strong>0.99</strong></td>
<td><strong>1.00</strong></td>
<td><strong>0.99</strong></td>
<td>0.76</td>
<td>0.70</td>
<td>0.67</td>
<td>0.64</td>
<td>0.46</td>
<td>0.46</td>
<td>0.81</td>
</tr>
<tr>
<td>SurfsUp (HQ BoF)</td>
<td>0.96</td>
<td>0.99</td>
<td>0.98</td>
<td>0.81</td>
<td><strong>0.76</strong></td>
<td>0.60</td>
<td>0.59</td>
<td><strong>0.77</strong></td>
<td><strong>0.77</strong></td>
<td>0.84</td>
</tr>
<tr>
<td>SurfsUp (SQ BoF)</td>
<td>0.98</td>
<td><strong>1.00</strong></td>
<td><strong>0.99</strong></td>
<td><strong>0.84</strong></td>
<td><strong>0.76</strong></td>
<td>0.60</td>
<td>0.57</td>
<td>0.60</td>
<td>0.60</td>
<td>0.85</td>
</tr>
<tr>
<td>SurfsUp (HQ BoF+scale)</td>
<td>0.96</td>
<td>0.99</td>
<td>0.98</td>
<td>0.80</td>
<td><strong>0.76</strong></td>
<td>0.73</td>
<td>0.80</td>
<td>0.72</td>
<td>0.72</td>
<td>0.86</td>
</tr>
<tr>
<td>SurfsUp (SQ BoF+scale)</td>
<td>0.96</td>
<td>0.99</td>
<td><strong>0.99</strong></td>
<td>0.77</td>
<td>0.70</td>
<td>0.68</td>
<td>0.71</td>
<td>0.56</td>
<td>0.56</td>
<td>0.86</td>
</tr>
<tr>
<td>SurfsUp (HQ ss-BoF)</td>
<td>0.96</td>
<td>0.98</td>
<td>0.97</td>
<td>0.82</td>
<td><strong>0.76</strong></td>
<td>0.59</td>
<td>0.58</td>
<td>0.70</td>
<td>0.70</td>
<td>0.82</td>
</tr>
<tr>
<td>SurfsUp (SQ ss-BoF)</td>
<td>0.96</td>
<td><strong>1.00</strong></td>
<td><strong>0.99</strong></td>
<td>0.81</td>
<td>0.73</td>
<td>0.60</td>
<td>0.57</td>
<td>0.62</td>
<td>0.60</td>
<td>0.85</td>
</tr>
<tr>
<td>SurfsUp (HQ ss-BoF+scale)</td>
<td>0.94</td>
<td>0.98</td>
<td>0.98</td>
<td>0.77</td>
<td>0.73</td>
<td><strong>0.74</strong></td>
<td><strong>0.86</strong></td>
<td>0.60</td>
<td>0.61</td>
<td>0.85</td>
</tr>
<tr>
<td>SurfsUp (SQ ss-BoF+scale)</td>
<td>0.96</td>
<td><strong>1.00</strong></td>
<td><strong>0.99</strong></td>
<td>0.77</td>
<td>0.72</td>
<td>0.68</td>
<td>0.72</td>
<td>0.57</td>
<td>0.56</td>
<td><strong>0.87</strong></td>
</tr>
</tbody>
</table>

**Best rank / 25**

|          | 4-6 | 1-6 | 7-8 | 2   | 3-7 | 3   | 1   | 1   | 1   | 2-3 |

Table 4.2 AUC results for DeeplyTough\textsuperscript{176}, ToughGraph and all variants of the SurfsUp algorithm. The ranking is computed for the best performing SurfsUp variant, calculated from the results presented by Ehrt et al.\textsuperscript{175} and including DeeplyTough and ToughGraph.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$P_1$</th>
<th>$P_{1.2}$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>$P_4$</th>
<th>$P_5$</th>
<th>$P_{5.2}$</th>
<th>$P_6$</th>
<th>$P_{6.2}$</th>
<th>$P_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeeplyTough</td>
<td>0.78</td>
<td>0.90</td>
<td>0.57</td>
<td>0.76</td>
<td>0.77</td>
<td>0.33</td>
<td>0.28</td>
<td>0.37</td>
<td>0.37</td>
<td>0.45</td>
</tr>
<tr>
<td>ToughGraph (embed)</td>
<td>0.84</td>
<td>0.92</td>
<td>0.80</td>
<td>0.75</td>
<td>0.54</td>
<td>0.29</td>
<td>0.25</td>
<td>0.35</td>
<td>0.44</td>
<td>0.46</td>
</tr>
<tr>
<td>ToughGraph (match)</td>
<td>1.00</td>
<td>0.95</td>
<td>0.93</td>
<td>0.80</td>
<td>0.76</td>
<td>0.35</td>
<td>0.29</td>
<td>0.27</td>
<td>0.27</td>
<td>0.45</td>
</tr>
<tr>
<td>SurfsUp (HQ BoF)</td>
<td>0.80</td>
<td>0.89</td>
<td>0.91</td>
<td>0.80</td>
<td><strong>0.76</strong></td>
<td>0.28</td>
<td>0.27</td>
<td><strong>0.71</strong></td>
<td><strong>0.71</strong></td>
<td>0.46</td>
</tr>
<tr>
<td>SurfsUp (SQ BoF)</td>
<td>0.90</td>
<td>0.96</td>
<td><strong>0.97</strong></td>
<td><strong>0.83</strong></td>
<td>0.75</td>
<td>0.28</td>
<td>0.25</td>
<td>0.48</td>
<td>0.48</td>
<td><strong>0.47</strong></td>
</tr>
<tr>
<td>SurfsUp (HQ BoF+scale)</td>
<td>0.77</td>
<td>0.89</td>
<td>0.93</td>
<td>0.78</td>
<td>0.74</td>
<td>0.36</td>
<td><strong>0.40</strong></td>
<td>0.63</td>
<td>0.63</td>
<td><strong>0.47</strong></td>
</tr>
<tr>
<td>SurfsUp (SQ BoF+scale)</td>
<td>0.82</td>
<td>0.96</td>
<td><strong>0.97</strong></td>
<td>0.77</td>
<td>0.71</td>
<td>0.35</td>
<td>0.35</td>
<td>0.41</td>
<td>0.41</td>
<td><strong>0.47</strong></td>
</tr>
<tr>
<td>SurfsUp (HQ ss-BoF)</td>
<td>0.75</td>
<td>0.83</td>
<td>0.89</td>
<td>0.78</td>
<td>0.72</td>
<td>0.28</td>
<td>0.26</td>
<td>0.59</td>
<td>0.59</td>
<td>0.46</td>
</tr>
<tr>
<td>SurfsUp (SQ ss-BoF)</td>
<td>0.84</td>
<td>0.97</td>
<td>0.96</td>
<td>0.79</td>
<td>0.72</td>
<td>0.28</td>
<td>0.25</td>
<td>0.48</td>
<td>0.48</td>
<td>0.46</td>
</tr>
<tr>
<td>SurfsUp (HQ ss-BoF+scale)</td>
<td>0.71</td>
<td>0.83</td>
<td>0.94</td>
<td>0.74</td>
<td>0.70</td>
<td><strong>0.37</strong></td>
<td><strong>0.40</strong></td>
<td>0.47</td>
<td>0.48</td>
<td>0.46</td>
</tr>
<tr>
<td>SurfsUp (SQ ss-BoF+scale)</td>
<td>0.85</td>
<td><strong>0.97</strong></td>
<td><strong>0.97</strong></td>
<td>0.77</td>
<td>0.71</td>
<td>0.34</td>
<td>0.33</td>
<td>0.42</td>
<td>0.42</td>
<td><strong>0.47</strong></td>
</tr>
</tbody>
</table>

Table 4. 3 Average precision results for DeeplyTough$^{176}$, ToughGraph and all variants of the SurfsUp algorithm. Ranks are not calculated since Ehrt et al.$^{175}$ did not report this metric.
SurfsUp (SQ BoF) and (SQ ss-BoF) performed the best over P1 and 1.2 and thus are the least sensitive to variations within binding site definition. This makes intuitive sense since minor deviations causing assignment of a patch to a different codeword will be smoothed across the probability distribution in the soft-quantization procedure. The distance-dependent distinction between positive and negative binding site pairs (SQ BoF) is displayed in Figure 4.12. A Welch’s two sample t-test performed on the distance distributions for positive and negative pairs, results in a t-statistic of 324.9 and a p-value $< 2.20e^{-16}$ indicating a significant difference between scores assigned to positive and negatively labelled pairs. SurfsUp performs considerably better than DeeplyTough retrieving many more positive pairs. We believe that this is due to the histogram-based representation enabling the similarity to focus on areas of local similarity. The patch-based representation also facilitates a decreased sensitivity to binding site flexibility since local regions may not change much with small deformations. This hypothesis is reflected by a strong performance for all SurfsUp methods over the NMR ensemble dataset P2. Note that global residue and interactions-based methods such as Cavbase$^{167}$, IsoMIF$^{129}$ and TIFP$^{169}$ perform significantly worse here due to their inherent sensitivity to the binding site definition. ToughGraph (embed) follows this trend while ToughGraph (match) performs remarkably for the first three datasets, likely due to the cross-graph messaging function focusing on obvious correspondences despite differing representation facilitated by differently sized ligands.

The decoy datasets P3 and P4, measure the discrimination ability of algorithms by introducing five artificial mutations into identical binding sites. SurfsUp has a remarkable capability to discriminate sites differing in chemical properties (P3) using the SQ BoF.
algorithm, ranking second out of all the algorithms benchmarked by Ehrt et al. The algorithm however has a harder time when both shape and chemical properties are mutated (P4), however still performs favorably compared to other tools with a ranking of 3-7.

Datasets P5/.2 is where SurfsUp struggles to differentiate between binding site classes, with relatively lower scores for unscaled BoFs. The sets compiled by Kahraman et al. contain binding sites containing the molecules/molecular classes: GLC, PO4, Steroids, AMP, Heme, FMN, ATP, NAD and FAD where P5 removes PO4 from consideration. Analysis of false positives reveals that SurfsUp often assigns high scores to pairs of nucleotides, for example SurfsUp (SQ BoF) assigns a chi squared score of 0.91 to a comparison between an ATP bound site (1AYL) and an AMP bound site (12AS), since both molecules contain an adenosine motif is it not feasible that both sites share some form of local similarity? Since our method inherently focuses on local features rather than the site as a global entity differentiating such sites will be a difficult task. Analysis of false positives reveals many cases where binding sites are completely different shapes or the binding of one side of the pair mediated by metal ions. This is especially true for PO4 where binding sites seem to have little homology. As others have pointed out before a simple way to improve performance with this dataset is to include the scale of the binding site. The simple inclusion of the size dependent scaling function lifts the performance of SurfsUp to a top three contender. Note that the other tools within this top three also utilise size in their scoring functions. The effect is even more pronounced for P5.2 where PO4 binding sites are re-introduced. The small size of these sites makes them simple to differentiate when including a size-based scoring function. This trend is observable when comparing to the results collected by Ehrt et al. and, indeed for unscaled BoF variants and
DeeplyTough\textsuperscript{176}, where the majority of binding site comparison tools decline in performance from P5 to P5.2 (17 out of 22 tested). In the cases where performance doesn’t decline (e.g. IsoMIF and SiteHopper) most include some form of size dependent scoring.

SurfsUp performs reasonably on the P6/6.2 dataset, considering no method reported by Ehrt et al. was able to make any meaningful judgements. This may simply be because of the nature in which the dataset was constructed whereby some positive pairs contain no observable residue correspondences. The high performance of HQ BoF seems to be related to the fact that it can reject negatives with a higher accuracy (high specificity).

Performance on the final dataset P7 is comparable with the best performing methods across the tested set. Adding scale benefits performance since negatives can be more reliably rejected and most positive examples are bound to similarly sized ligands. A clear distance dependent distinction is witnessed for the dataset as displayed in Figure 4.12. A Welch’s two sample t-test performed on the distance distributions for positive and negative pairs, results in a t-statistic of 532.8 and a p-value \(< 2.20e^{-16}\) indicating a significant difference between scores assigned to positive and negatively labelled pairs.

From our results we suggest that SurfsUp (SQ BoF) is the most versatile variation of the algorithm achieving the highest average scores across the ProSPECCTs dataset. Surprisingly the spatially sensitive variation adds little benefit given the extra memory and computation time required. Soft quantization schemes prove more effective in most cases and whether or not scaling should be used must be considered on a case-by-case basis, for instance if one is only interested in retrieving sites bound to similarly sized ligands. Based
on the best performing SurfsUp (SQ BoF) algorithm we present some additional metrics for understanding the strengths and weaknesses of our approach.

Figure 4. 12 Boxplots displaying the distance dependent distinction between positive and negative pairs of binding sites, in the P1 dataset (a) and the P7 dataset (b). The chi squared similarity is computed using the SurfsUp (SQ BoF) variant. The purple line represents the optimal threshold computed from the ROC curve using Youden’s J statistic.\(^{187}\)
In line with the benchmarking performed by Ehrt et al\textsuperscript{175}, a sensitivity and specificity analysis was performed for both datasets P1 and P7, being the most representative of realistic use cases. Sensitivity indicates a binding site comparison tools ability to rank similar pairs highly while specificity indicates the ability to reject negative pairs. Both metrics were calculated from the optimal threshold defined by the Youden’s J statistic\textsuperscript{187}. For ease of comparison we display only the results for the most similar algorithms (and best performing) to ours, namely: DeeplyTough (machine learning), Pocket-Match (histogram)\textsuperscript{166}, RAPMAD (histogram)\textsuperscript{165}, TM-Align (Cβ-based alignments)\textsuperscript{174} and, Shaper (pharmacophore/surface)\textsuperscript{171} (Table 4.4).
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cut-off</th>
<th>Specificity</th>
<th>Sensitivity</th>
<th>Rank (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pocket-Match</td>
<td>0.16</td>
<td>0.86</td>
<td>0.73</td>
<td>3</td>
</tr>
<tr>
<td>RAPMAD</td>
<td>0.89</td>
<td>0.51</td>
<td>0.71</td>
<td>5</td>
</tr>
<tr>
<td>Shaper</td>
<td>0.89</td>
<td>0.94</td>
<td>0.57</td>
<td>17</td>
</tr>
<tr>
<td>TM-Align</td>
<td>0.49</td>
<td>0.99</td>
<td>0.70</td>
<td>6</td>
</tr>
<tr>
<td>DeeplyTough</td>
<td>-0.29</td>
<td>0.88</td>
<td>0.87</td>
<td>2</td>
</tr>
<tr>
<td>SurfsUp (SQ BoF)</td>
<td>0.85</td>
<td>0.93</td>
<td>0.90</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cut-off</th>
<th>Specificity</th>
<th>Sensitivity</th>
<th>Rank (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pocket-Match</td>
<td>0.26</td>
<td>0.95</td>
<td>0.70</td>
<td>7-8</td>
</tr>
<tr>
<td>RAPMAD</td>
<td>0.97</td>
<td>1.00</td>
<td>0.42</td>
<td>21</td>
</tr>
<tr>
<td>Shaper</td>
<td>0.98</td>
<td>0.97</td>
<td>0.54</td>
<td>16</td>
</tr>
<tr>
<td>TM-Align</td>
<td>0.38</td>
<td>0.98</td>
<td>0.79</td>
<td>1</td>
</tr>
<tr>
<td>DeeplyTough</td>
<td>-0.05</td>
<td>0.87</td>
<td>0.70</td>
<td>7-8</td>
</tr>
<tr>
<td>SurfsUp (SQ BoF)</td>
<td>0.81</td>
<td>0.86</td>
<td>0.73</td>
<td>3-4</td>
</tr>
</tbody>
</table>

Table 4. 4 Sensitivity and specificity analysis for datasets P1 and P7. The cut-offs are calculated using the Youden’s J statistic\textsuperscript{187} and ranks are calculated from the sensitivity using the results published by Ehrt et al.\textsuperscript{175}
The results of the sensitivity/specificity analysis were particularly interesting. Both machine learning-based methods, ours and DeeplyTough, outperform all other methods in their ability to recover similar binding sites from identical sequences bound to diverse ligands (P1). Perhaps this indicates that our hypothesis that traditional handcrafted features couples with tight optimisation data does lead to over-optimisation where simple cases may be missed. Machine learning based methods seem to have the potential to introduce less bias for the task. The effect for SurfsUp is the most pronounced achieving both a higher sensitivity and specificity than DeeplyTough. We think that using a local rather than a global descriptor is the main reason for the higher result, reducing sensitivity to minor local deviations and flexibility of the binding site. It is however, also clear that this extra sensitivity comes with a decreased specificity (albeit small for SurfsUp) indicating a susceptibility to false positive labelling. Strikingly RAPMAD has a near random specificity indicating it has a lot of trouble ranking negatives lower than positives. It seems that the histogram generation approaches of Pocket-Match and SurfsUp are more robust in this context. Shifting attention to dataset P7, we note that SurfsUp achieves a high sensitivity relative to other methods (tied 3rd with IsoMIF) with DeeplyTough also performing within the top quarter of tools (rank 7-8). We do however observe a specificity cost with both approaches, being amongst the lowest ranks for this metric. The result indicates a high false positive rate for this particular set. In some aspects this is difficult to interpret given high performance in other areas, there are realistically two feasible explanations: (1) both tools are able to identify more remote similarities which other tools have not been optimized to find, or (2) the algorithms simply rank negative pairs too highly. In the case of DeeplyTough this could either be a positive or negative effect of training on a weakly-
labelled dataset and with SurfsUp, this could be due to the inherent nature of the descriptor, i.e. not considering the relationships between surface patches. Analysing false positives systematically here is intractable due to high imbalance in the set (115 positives, 56,284 negatives). We are inclined to suggest that tools that can identify more remote similarities with higher ’false positive’ rates are likely a better fit for an off-target prediction pipeline rather than a drug repurposing objective where high similarity is particularly important.

4.3.4.2 TOUGH-C1 Benchmark

To further validate the SurfsUp algorithm, a binding site classification task was performed using the TOUGH-C1 database\cite{196}. Two distinct objectives were evaluated: differentiation of nucleotide binding sites from the control set and differentiation of heme binding sites from the control set. We followed the same procedure used by Pu et al.\cite{196}, for the evaluation of the binding site comparison tool G-LoSA\cite{185}, where for each binding site in the nucleotide-binding and control set, pairwise SurfsUp chi squared similarities are computed against all nucleotide binding sites excluding self-pairs (i.e. $\chi^2 = 1.0$). ROC analysis was performed where entry rankings were computed from the average chi squared score for each entry. The same process is repeated for the heme-binding classification task. The SurfsUp parameters used were as follows: soft-quantization, no scaling $R = 6.0$ and $V = 512$. We chose to omit scaling for this experiment as others have shown that heme and nucleotide binding sites can be distinguished solely on shape\cite{190,265}, which would present an overestimation of the ability of our algorithm to classify binding sites based on local shape and chemical features alone.
Results were compared with numerous algorithms including: DeepDrug3D\textsuperscript{196}, ScanProsite\textsuperscript{266}, G-LoSA\textsuperscript{185}, AutoDock Vina\textsuperscript{267}, and HOG/PCA. Two further branches of DeepDrug3D were also compared: volume and shape. The volume branch simply computes the volume occupied within a voxel grid, while the shape branch retrains the 3D-CNN using only occupancy data. Results are presented in Table 4.5. DeepDrug3D is a 3D-CNN approach developed specifically for this particular dataset, utilising voxel-based representations of protein binding sites. HOG/PCA is an extension of DeepDrug3D, computing a histogram of gradients feature for the voxel grids, and subsequently performing PCA on the feature for dimensionality reduction (1,204 elements). ScanProsite is a sequence-based approach in which the presence of specific PROSITE signatures is used to make a target classification. G-LoSA is a binding site comparison approach and is thus the most similar method to SurfsUp. The approach taken by G-LoSA however is alignment-dependent and thus is more computationally demanding. The final algorithmic approach, AutoDock Vina is docking software and is used to perform a reverse virtual screen against both nucleotide and heme binding sites.
Table 4. 5 Performance of various algorithms for classifying ligand binding sites included in the TOUGH-C1 dataset. Table (A) nucleotide class, and (B) heme class. All results from approaches other than SurfsUp are taken from the TOUGH-C1 publication. Performance is evaluated using the measures: accuracy (ACC), precision (PPV), sensitivity (TPR), specificity (TNR) and the area under the ROC-curve (AUC). *Soft-quantisation and a codebook size of $V = 512$ is used for both algorithms.
DeepDrug3D outperforms all approaches, achieving an AUC of 0.986 and 0.987 for nucleotide and heme binding sites respectively. This is hardly unexpected considering that it was trained specifically for this purpose as opposed to the majority of other methods evaluated. It is also important to realise that while some care was taken to minimise bias when training DeepDrug3D, through generating a non-redundant dataset in terms of sequence homology, no analysis was performed to measure structural diversity across training and validation folds. It is probable that proteins containing similar structural folds exist across training and validation sets introducing an over estimated performance.

SurfsUp is ranked second for nucleotide classification with an AUC of 0.862/0.837 outranking the method trained on occupancy data alone (AUC 0.824). Although shape alone clearly displays some competence in classification, nucleotides are flexible molecules and hence it appears that the addition of chemical properties considered by SurfsUp aids classification. This hypothesis is supported by the decreased specificity when shape features are used alone. Both ScanProsite and SurfsUp achieve remarkable specificity (0.970, 0.991) on this dataset, however, while ScanProsite struggles to classify nucleotides successfully, reflected by a sensitivity value of 0.411, SurfsUp (BoF) achieves a sensitivity of 0.741. Furthermore, while DeepDrug3D achieves a high positive classification rate, SurfsUp (BoF) can distinguish negatives more reliably based on the optimal threshold. Intriguingly SurfsUp (BoF) also achieves a very high precision (0.986) indicating that the majority of nucleotide binding sites a discovered in the top-ranking positions. Computed enrichment factors (Table 4.6a) further highlights this with the top 20-30% of the recovered entries consisting entirely of nucleotide binding sites. SurfsUp
struggles to distinguish the remaining nucleotide binding sites from the control set and while this may be a weakness of the current method, others have argued that there is often limited structural similarity between some nucleotide binding sites and that shared binding could be accredited to lower-specificity of the ligand or conformational flexibility.\textsuperscript{34}

(A)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$EF_{1%}$</th>
<th>$EF_{5%}$</th>
<th>$EF_{10%}$</th>
<th>$EF_{20%}$</th>
<th>$EF_{30%}$</th>
<th>$EF_{40%}$</th>
<th>$EF_{50%}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurfsUp (BoF)</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>1.89</td>
<td>1.61</td>
</tr>
<tr>
<td>SurfsUp (SS-BoF)</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>1.99</td>
<td>1.78</td>
<td>1.56</td>
</tr>
</tbody>
</table>

(B)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$EF_{1%}$</th>
<th>$EF_{5%}$</th>
<th>$EF_{10%}$</th>
<th>$EF_{20%}$</th>
<th>$EF_{30%}$</th>
<th>$EF_{40%}$</th>
<th>$EF_{50%}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurfsUp (BoF)</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>1.98</td>
<td>1.69</td>
</tr>
<tr>
<td>SurfsUp (SS-BoF)</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>1.94</td>
<td>1.64</td>
</tr>
</tbody>
</table>

Table 4. \textit{6} Enrichment factors for the SurfsUp algorithm with respect to the retrieval of nucleotide (A) and heme (B) binding sites within a diverse set of control structures defined within the TOUGH-C1 dataset.\textsuperscript{196} Soft-quantisation and a codebook size of $V = 512$ is used for both algorithms.
Generally, all methods perform better on the heme classification task. This may be attributed to the fact that heme is considerably more rigid molecule compared to nucleotides and that the binding sites are more structurally conserved. The same observation was made by Kahraman et al.\textsuperscript{190}, who analysed shape variations within protein-ligand binding sites within the PDB, measuring distances between spherical harmonics decompositions. They found that while steroid, AMP, PO\textsubscript{4} and heme binding sites deviate less in shape, the nucleotides ATP, FAD and NAD occupy significantly different conformations when bound to non-homologous proteins, an observation also made by Stockwell and Thornton\textsuperscript{268}. Conserved shape therefore is likely not a pre-requisite for shared binding. The relative structural conservation of heme binding sites is reflected by a high AUC for the shape-based approach (0.952). SurfsUp on the other hand remains fairly consistent achieving an AUC of 0.885 ranking fourth after DeepDrug3D (0.987), shape (0.952) and G-LoSA (0.917). The consistency of the approach demonstrates the ability of SurfsUp to recover binding sites with local similarities with a lower tolerance to flexibility. Despite the lower ranking of AUC we note that SurfsUp (BoF) achieves the highest precision and second highest sensitivity after DeepDrug3D. Enrichment factors (Table. 4.6 b), demonstrate a similar pattern as seen for the nucleotide dataset. The ss-BoF variant of SurfsUp achieves a particularly high precision on this particular dataset although we note that the simpler BoF variant outperforms in all other scenarios as was the case across the ProSPECCTs benchmarks.

The ability of SurfsUp to match or outperform both trained (shape) and non-trained methods in binding site classification tasks, demonstrates the power of the latent feature space learnt by the autoencoder in differentiating shape and chemical properties on
molecular surfaces. We postulate that a trained method using raw latent features, in a similar manner to the MaSIF\textsuperscript{221} approach could have the ability to compete with DeepDrug3D. This could be an interesting avenue for further research. The current version however has the benefit of a large domain of applicability and can therefore be applied to binding sites outside of a three-way classification scenario as highlighted in the following case studies and the ProSPECCTs benchmarks.

4.3.5 Case Studies

The application of SurfsUp to real biological problems was studied using a series of case studies. The first case study tested the ability of the algorithm to retrieve enzymes catalysing the same reaction, the second replicates a recent study identifying kinase inhibitors as potential synapsin-1 inhibitors and the third analyses the similarity between metalloproteinase enzymes. All case studies were performed using data within the sc-PDB database\textsuperscript{192} and the best set of parameters identified for the SurfsUp algorithm: soft-quantization, no scaling $R = 6.0$ and $V = 512$.

4.3.5.1 Retrieval of Enzymes Catalysing the Same Reactions

To test the ability of the SurfsUp algorithm to retrieve related binding sites, we design an experiment in which the objective is, given an enzyme, to return all enzymes catalysing the same reaction. The assumption being that enzymes sharing catalytic function will share 3D and chemical similarity of their catalytic centers. To realise this, Enzyme Commission (EC)
numbers are assigned to all enzymes within the sc-PDB dataset\textsuperscript{192}, non-enzymes are removed along with enzymes with less than 5 entries, resulting in a total of 8942 protein structures. The EC number is a simple numerical classification scheme for enzymes, with each unique number specific to a certain enzyme-catalysed reaction rather than to identical protein structures. This classification scheme, in contrast to UniProt identifiers, thus may classify enzymes from different organisms, or unique protein folds (non-homologous isofunctional enzymes) under the same identifier. Due to the nature of the classification this experiment tests the ability of the algorithm to retrieve similar binding sites in the top ranks despite a potential lack of global homology.

For an initial experiment we first test the algorithms ability to retrieve carbonic anhydrase II (CA-II) (EC number 4.2.1.1) enzymes from the EC labelled subset. CA-II is involved in the inter-conversion of carbon dioxide and water into carbonic acid, protons and bicarbonate ions and is associated with glaucoma, malignant brain tumours and renal, gastric and pancreatic carcinomas. Inhibitors of CA-II are used to relieve the symptoms of glaucoma\textsuperscript{269}. The structure 1OQ5 is matched against all EC annotated binding sites, including 171 labelled as CA-II (i.e. the same as the query) which are defined as correct hits. The result of the retrieval is illustrated by the ROC and PR curves in Figure 4.13. SurfsUp achieves an AUC of 0.98 and an AP of 0.85 illustrating a strong ability to retrieve binding sites of enzymes with the same function. Note also the high early enrichment of the retrieval, with 89\% of all CA-II entries retrieved within 2\% (179 entries) of the dataset and 96\% recovered within 10\% (894 entries). The remaining CA-II entries are not retrieved simply because the binding site is a separate site with alternate function. For example, in
entry 4HEY, 4-methylimidazole is bound to the protein in alternate locations to the catalytic site acting as activity enhancers.

Figure 4. 13 ROC and PR curves illustrating the retrieval results of carbonic anhydrase II (EC 4.2.1.1) binding sites using the SurfsUp algorithm. A carbonic anhydrase binding site (1OQ5) was used as a query, compared with all EC annotated binding sites within the sc-PDB database\textsuperscript{192}. The retrieval experiment demonstrates the ability of the algorithm to retrieve enzymes catalysing the same reaction (i.e. same EC number).

Considering the positive results achieved in the initial retrieval experiment we extend the experiment to consider all EC classifications, using all 8942 structures as queries. The retrieval ability is measured using AUC and the ability to retrieve the correct EC classification in the top 1, 3 and 1% ranks of sorted results. SurfsUp achieves an average AUC of 0.80±0.15 and average top 1, 3 and 1% scores of 78.2%, 85.2% and 95.5%
respectively, results are summarised in Table 4.7. The top 1 score indicates that 6992 out of 8942 queries returned an enzyme of the same classification in the first ranked position, further illustrating the ability of the algorithm to retrieve enzymes with the same function.

<table>
<thead>
<tr>
<th>Top Positive (%)</th>
<th>1</th>
<th>3</th>
<th>1%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.2</td>
<td>85.2</td>
<td>95.5</td>
</tr>
</tbody>
</table>

Table 4.7 The ability of SurfsUp (BoF) to retrieve binding sites with the same Enzyme Commission (EC) number within the top ranks. In contrast with UniProt identifiers EC numbers can refer to proteins with different protein folds, since different enzymes may catalyse the same reaction due to the effects of convergent evolution.

While SurfsUp was able to correctly predict the EC classification for enzymes in the sc-PDB from the top rank in 78.2% of the entries, it is of interest to understand failure cases. Two main causes of failure are observed: (1) Within an EC class, proteins bind functionally different molecules at separate binding sites and (2) binding site definitions are very different due to the size/shape of the bound ligand or the binding site is flexible and varies largely in its surface shape. The first failure case is actually positive for the algorithm considering we should not expect high similarities between functionally different binding sites on the same protein structure. An example of this case is found in the protein: L-amino acid oxidase (LAAO, EC 1.4.3.2) which contains three domains: a flavin-adenine dinucleotide (FAD) binding site a substrate-binding domain and a helical domain. The sc-
PDB entry, 2JAE, is representative of the FAD binding site while other entries such as the one for 1F8S represent a region of the active-site bound to an inhibitor, 2-aminobenzoic acid (BE2). SurfsUp is of course unable to assign 2JAE the correct EC number when comparing it to these other entries. In this case the matter is also further complicated since BE2 binds in three discrete positions within the entrance to the active site making comparisons between binding sites defined by the inhibitor difficult. The second case is common to any binding site comparison tool employing a global measure of similarity. Such methods may struggle to retrieve sites with only partial local similarities, especially in the case where the binding site definition differs in size or sites are diverse in shape due to the binding of large flexible molecules which can be accommodated in multiple diverse conformations. The second failure case is difficult to account for in many cases due to the difficulty of defining a binding site deterministically. We identify that SurfsUp struggles to retrieve Hepatitis C virus NS3 serine proteinase (EC 3.4.21.98) where the majority of entries comprise the RNA binding site. Due to the large nature of this particular binding site inhibitors bind at multiple different locations within the site and hence binding site definitions are different as a result. Despite this SurfsUp is less sensitive to failure case two than other approaches as exemplified by the conducted benchmarks.

4.3.5.2 Synapsin-1

The traditional dogma of: “one drug, one target” in the pharmaceutical industry is an outdated paradigm, and we are now aware that drug-like small-molecules interact with on average 6 to 11 proteins in addition to their intended target. This view certainly played
a predominant role in the high attrition rate in clinical trials, due to off-target interactions causing toxicity, side-effects and adverse drug reactions\textsuperscript{2}. Systematic approaches for identifying off-targets (\textit{anti-targets}) have been proposed using chemogenomic data and small molecule similarities\textsuperscript{242} and have been successfully used to discover several off-targets for known drugs. The issues with ligand/drug-centric approaches however is two-fold: (1) they are inherently restricted by a lack of target annotation and thus are useless when considering orphan targets, and (2) the similarity property principal, inferring that structural similarity implies biological similarity is only correct 30\% of the time\textsuperscript{271}. The alternative is to use target-based approaches to quantify the similarity between binding sites with the assumption that similar sites should recognise similar ligands. Despite numerous proposals for binding site comparison algorithms, very few have reported results of off-target predictions through systematic binding site comparison\textsuperscript{272}.

Notably in a publication by Defranchi et al.\textsuperscript{272}, systematic binding site comparisons revealed synapsin I, an ATP-binding protein regulating synaptic plasticity and synaptogenesis\textsuperscript{273}, as a potential off-target for serine/threonine kinase inhibitors. The authors made this observation when performing a target-based virtual screen using a binding site of the pan-kinase inhibitor, staurosporine, in a proto-oncogene Pim-1 serine/threonine protein kinase (1YHS). A subsequent virtual screening of the sc-PDB using a synapsin-1 binding site (1AUX), revealed high rankings of various kinases including: proto-oncogene Pim-1 serine/threonine protein kinase (PIM1), Casein kinase II subunit alpha (CKII) and Cyclin-dependent kinase 2 (CDK2). Inhibitors of these three targets and five further low scoring hits: cAMP-dependent protein kinase catalytic subunit \( \alpha \) (PKA), Heat shock protein HSP-90\( \alpha \) (HSP-90\( \alpha \)), Serine/threonine-protein kinase Chk1
(CHK1), DNA topoisomerase II and diacylglycerolkinase (DGK), were purchased for in vitro screening. Their results shown in table 4.8, indicate nanomolar affinities for synapsin-1 for inhibitors of the most similar kinases classes, promoting synapsin-induced F-actin bundling, while others demonstrated no significant affinity. The top three hit molecules are shown in Figure. 4.14.

<table>
<thead>
<tr>
<th>Inhibitor</th>
<th>Target</th>
<th>IC50 (µM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staurosporine</td>
<td>Pan-kinase</td>
<td>0.31±0.09</td>
</tr>
<tr>
<td>(R)-Roscovitine</td>
<td>CDK2</td>
<td>1.00*</td>
</tr>
<tr>
<td>Quercetagetin</td>
<td>PIM1</td>
<td>0.15±0.08</td>
</tr>
<tr>
<td>7015980251</td>
<td>CKII</td>
<td>0.50*</td>
</tr>
<tr>
<td>4072-2730</td>
<td>CHK1</td>
<td>N/A</td>
</tr>
<tr>
<td>R59022</td>
<td>DGK</td>
<td>N/A</td>
</tr>
<tr>
<td>Novobiocin</td>
<td>DNA topoisomerase II</td>
<td>N/A</td>
</tr>
<tr>
<td>CCT018159</td>
<td>HSP-90α</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 4. 8 In vitro screening results of various kinase inhibitors against synapsin-1 as reported by Defranchi et al.272. Inhibitors marked with N/A for the IC50 readout display no significant activity against synapsin-1. *Standard deviations not reported.
Figure 4. The top three kinase inhibitor hits against synapsin-1 identified by\textsuperscript{272}, not including the pan-kinase inhibitor staurosporine.

The results presented by Defranchi et al.\textsuperscript{272} clearly demonstrate the potential of off-target prediction using systematic binding site comparisons. The approach used by the authors involves the SiteAlign algorithm\textsuperscript{168}, yet interestingly they also state that to state-of-the-art approaches: SitesBase\textsuperscript{181} and SiteEngine\textsuperscript{172} fail to recover the same result. Indeed use of Shaper\textsuperscript{171} through the sc-PDB web interface also fails to retrieve these kinase binding sites. Intrigued by this finding we aim to test the ability of the SurfsUp algorithm by conducting a virtual screen against the sc-PDB using the ATP-γS bound synapsin-1 binding site of 1AUX. Encouragingly both remaining synapsin-1 entries are recovered with the top ten hits, Table 4.9 amongst other proteins in the serine/threonine protein kinase family. The most common hit is the mitogen-activate protein kinase 1 (MAPK-1) protein
representing 50% of the top hits, with another MAPK (MAPK-10) ranked second. Both CKII and CDK2 proteins are also retrieved at ranks 3 and 5 respectively reflecting previously reported results\textsuperscript{272}. A RANSAC alignment of the highest ranking, experimentally tested hit (CKII, 1PJK) reveals a significant degree of surface shape similarity and eight amino acid correspondences displaying similar chemical properties and 3D positions (Figure. 4.15), despite a sequence identity of only 16.9%. Such local similarities potentially justify the shared affinity for the CKII inhibitor, 7015980251.

<table>
<thead>
<tr>
<th>PDB</th>
<th>Chi-squared</th>
<th>Name</th>
<th>HET</th>
</tr>
</thead>
<tbody>
<tr>
<td>1PX2</td>
<td>0.86</td>
<td>Synapsin-1</td>
<td>ATP</td>
</tr>
<tr>
<td>3KVX</td>
<td>0.84</td>
<td>Mitogen-activated protein kinase 10</td>
<td>FMY</td>
</tr>
<tr>
<td>1PJK</td>
<td>0.82</td>
<td>Casein kinase II subunit alpha</td>
<td>ANP</td>
</tr>
<tr>
<td>4H3P</td>
<td>0.82</td>
<td>Mitogen-activated protein kinase 1</td>
<td>ANP</td>
</tr>
<tr>
<td>1GII</td>
<td>0.81</td>
<td>Cyclin-dependent kinase 2</td>
<td>1PU</td>
</tr>
<tr>
<td>4FV8</td>
<td>0.81</td>
<td>Mitogen-activated protein kinase 1</td>
<td>E63</td>
</tr>
<tr>
<td>4FV9</td>
<td>0.81</td>
<td>Mitogen-activated protein kinase 1</td>
<td>E71</td>
</tr>
<tr>
<td>4FUY</td>
<td>0.81</td>
<td>Mitogen-activated protein kinase 1</td>
<td>EK2</td>
</tr>
<tr>
<td>1TVO</td>
<td>0.81</td>
<td>Mitogen-activated protein kinase 1</td>
<td>FRZ</td>
</tr>
<tr>
<td>1PK8</td>
<td>0.81</td>
<td>Synapsin-1</td>
<td>ATP</td>
</tr>
</tbody>
</table>

Table 4. 9 The top ten ranked hits from a virtual screening against the sc-PDB database\textsuperscript{192} using a synapsin-1 query (1AUX) bound to phosphothiophosphoric acid-adenylate ester (ATP-$\gamma$S). Both synapsin-1 entries are retrieved within the top ten hits (1PX2, 1PK8).
For each protein target we further report the iterative AUC (Table 4.10), calculated by ranking binding sites by decreasing similarity to synapsin-1 and computing the rank distribution for binding sites sharing the same protein name (true positives). The results further reinforce our previous findings with seven serine/threonine kinases retrieved with the highest AUC including: MAPK-1, MAPK-10, CDK2, MAPK-8 and CKII. Of these, inhibitors of CDK2 and CKII were found to be active against synapsin-1 (Table. 4.8). Furthermore, Quercetagetin, which is also active against synapsin-1 has demonstrated activity against MAPK-8\textsuperscript{274} indicating that this may also represent a valid judgment of similarity. It is of interest to note that SurfsUp did not identify PIM1 as a relevant hit (AUC 0.17) despite its inhibitor Quercetagetin being identified as the most potent tested kinase inhibitor against synapsin-1. On further investigation we find that RANSAC binding site alignments fail between PIM1 kinases and the synapsin site within 1AUX due to different surface shapes despite the existence of amino acid correspondences within the site. While this may represent a potential failure case of surface-based comparison approaches it is pertinent to mention that the PIM1 kinase inhibitor, Quercetagetin, is a promiscuous kinase inhibitor\textsuperscript{34} and also binds to various hydrolases, reductases and transporter proteins\textsuperscript{275}. Additionally, polyphenol containing molecules are often considered within the spectrum of assay interfering compounds\textsuperscript{276}, and further experiments with more selective molecules should be considered before making a definitive conclusion.
### Table 4

<table>
<thead>
<tr>
<th>Protein Kinase</th>
<th>Iterative AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mitogen-activated protein kinase 1</td>
<td>0.93</td>
</tr>
<tr>
<td>Mitogen-activated protein kinase 10</td>
<td>0.92</td>
</tr>
<tr>
<td>Cyclin-dependent kinase 2</td>
<td>0.92</td>
</tr>
<tr>
<td>Mitogen-activated kinase 8</td>
<td>0.90</td>
</tr>
<tr>
<td>Casein kinase II subunit α</td>
<td>0.89</td>
</tr>
<tr>
<td>Aurora kinase A</td>
<td>0.88</td>
</tr>
<tr>
<td>Serine/threonine-protein kinase Nek2</td>
<td>0.86</td>
</tr>
<tr>
<td>cAMP-dependent protein kinase catalytic subunit α</td>
<td>0.42</td>
</tr>
<tr>
<td>Heat shock protein HSP-90α</td>
<td>0.32</td>
</tr>
<tr>
<td>Serine/threonine-protein kinase Chk1</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Table 4. 10 Iterative AUC scores for select protein kinases ($N > 10$) when performing a virtual screen using a synapsin-1 binding site (1AUX) against the sc-PDB database. The top seven represent the highest ranked protein-kinases while the bottom three rows are protein kinases for which inhibitors were tested against synapsin-1 with no significant activity. The low rankings of the protein kinases: PKA, HSP-90α and CHK1 by SurfsUp, reflects the results computed using SiteAlign and is also reinforced by the in vitro results for inhibitors of these targets against synapsin-1. The results demonstrate that while one may consider that detecting similarities between ATP binding sites trivial both SurfsUp and SiteAlign are able to rank these sites in a biologically relevant manner, reflected in recorded activity profiles for respective inhibitors.
Figure 4. 15 A RANSAC surface alignment between synapsin-1 (1AUX), bound to ATP-\(\gamma\)S and casein kinase II subunit \(\alpha\) (1PJK) bound to phosphomethylphosphonic acid adenylate ester (ANP). Panel (a) displays the aligned surfaces and the associated ligand superposition, and panel (b) shows the eight amino acid correspondences displaying similar chemical properties and position.
The results of this case study exemplify the potential to use SurfsUp for off-target prediction. SurfsUp was able to find clear similarities between the binding site of synapsin-1 and various serine/threonine protein kinases. This result was also verified by \textit{in-vitro} ATP competition assays\textsuperscript{272} verifying the biological validity of similarity judgements. Our results reinforce the attractive hypothesis made by Defranchi et al. that observed effects of kinase inhibitors on neurotransmitter release and presynaptic function may be partially ascribed to direct binding to synapsin-1. While synapsin-1 has not been considered as a drug-target to-date, the observations made may be used to inform the design of selective kinase inhibitors or the repurposing of available kinase inhibitors.

4.3.5.3 Matrix Metalloproteinase

High throughput screening (HTS) is a fundamental process in modern drug-discovery with the ability to identify potential lead molecules within collections of compounds. One fundamental issue with this approach is that selectivity is not accounted for. Selectivity is of primary concern when ranking potential lead compounds especially when considering the activity profiles within functionally related proteins. Determination of selectivity is thus usually achieved using a counter screen against a small collection of sequenced-related targets, although this method is inherently limited since sequence homology is not a good indication of functional/active site similarity. It is now widely understood that many distantly related proteins share similar active sites due to the effects of convergent evolution\textsuperscript{129} and consequently have significant overlap in their inhibitor sensitivity profiles.
Here, a case study of matrix metalloproteinases is conducted to demonstrate the ability of SurfsUp to identify targets which should be considered when designing selective inhibitors.

Matrix metalloproteinases (MMPs) have been identified as a potential drug target due to their implication in pathologies, including: cancer, central nervous system disorders, cardiovascular disorders and autoimmune diseases including rheumatoid- and osteoarthritis\textsuperscript{277}. Despite efforts, only one MMP inhibitor (MMPI), Periostat, has been approved for use by the FDA for the treatment of periodontal diseases. Early efforts involving broad-spectrum hydroxamate and non-hydroxamate-type inhibitors, failed in late stage clinical trials due to severe musculoskeletal side effects\textsuperscript{277}, more than likely due to the highly complex nature of human MMPs, of which many share a similar catalytic site. Furthermore, studies have identified MMP isoforms with a protective role in angiogenetic and metastatic events\textsuperscript{278}, (anti-targets) rendering broad-spectrum MMP inhibition approaches potentially problematic. Cross-reactivity between different MMP isoforms and other pro-survival members of the metzincin super-family: a disintegrin and metalloproteinases (ADAMs) and ADAMs with thrombospondin motifs (ADAMTSs) has been suggested as the most likely cause of clinical trial failure\textsuperscript{277}. The search for target-selective MMPIs is still a hot-topic, and knowledge of structural similarities will be key to future developments.

In this case study a collagenase-3 (MMP-13, PDB ID: 1FM1) binding site is used as a query for a target-based virtual screen against the sc-PDB database (16,250 binding site entries) using the SurfsUp algorithm. MMP-13 is a key potential drug target due to its implications in many pathologies including various cancers and arthritis which is characterised by an increased expression of both MMP-13 and MMP-1\textsuperscript{279}. However, while many inhibitors have been designed, the majority have failed clinical trials presumably due
to a lack of target selectivity. The premise behind this case study is thus to use binding site comparison to identify potential sources of cross-reactivity. A summary of metzincins contained in the sc-PDB is provided in Table 4.11.

<table>
<thead>
<tr>
<th>Metalloproteinase (metzincin)</th>
<th>Number of Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMP-2 (72 kDa type IV collagenase)</td>
<td>1</td>
</tr>
<tr>
<td>MMP-3 (stromelysin-1)</td>
<td>25</td>
</tr>
<tr>
<td>MMP-8 (neutrophil collagenase)</td>
<td>4</td>
</tr>
<tr>
<td>MMP-9 (gelatinase B)</td>
<td>9</td>
</tr>
<tr>
<td>MMP-10 (stromelysin-2)</td>
<td>1</td>
</tr>
<tr>
<td>MMP-11 (stromelysin-3)</td>
<td>1</td>
</tr>
<tr>
<td>MMP-12 (macrophage metalloelastase)</td>
<td>24</td>
</tr>
<tr>
<td>MMP-13 (collagenase-3)</td>
<td>22</td>
</tr>
<tr>
<td>MMP-16</td>
<td>1</td>
</tr>
<tr>
<td>MMP-20 (enamelysin)</td>
<td>1</td>
</tr>
<tr>
<td>ADAM2</td>
<td>1</td>
</tr>
<tr>
<td>ADAM17</td>
<td>12</td>
</tr>
<tr>
<td>ADAMTS1</td>
<td>1</td>
</tr>
<tr>
<td>ADAMTS4</td>
<td>1</td>
</tr>
<tr>
<td>ADAMTS5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.11 A summary of metalloproteinases within the metzincin superfamily contained within the sc-PDB database. Only entries for which SurfsUp BoF features could be calculated are counted. Note that there are many other isoforms of MMP/ADAM/ADAMTS which are not included within the sc-PDB or do not have any 3D structural data available.
The virtual screening (Figure 4.16) retrieved many MMP binding sites with MMP-13 and MMP-3 being the most common isoforms within the top 20 hits. The next highest-ranking isoform is MMP-9 at rank eight (chi squared 0.87). Encouragingly the majority of MMP isoforms were retrieved within the top 10% of the dataset with only MMP-20 and MMP-10 isoforms being returned at low ranks. Interestingly by changing the query MMP-13 binding site to 3O2X MMP-20 can be retrieved within the top 10% whereas no query structure retrieves the MMP-10 isoform. The fact that changing the query structure can return initially low ranked structures suggests that there is an inherent sensitivity to binding site definition. This sensitivity is likely a result of the initial definition of input surface patches, perhaps a more robust definition or using all surface vertices as patch locations would reduce this observed behaviour. Further analysis of scores for retrieved MMP-13 reveals a score distribution from 1.00 to 0.56, which while above one standard deviation from the mean similarity is a fairly wide distribution for a single isoform. Despite the wide range of scores, the iterative AUC for MMP-13 (Table 4.12) retrieval is 0.93 indicating a high confidence similarity between binding sites of this isoform. This result highlights sequence independence of the matching and the fact that often multiple structures are required to get high confidence results due to a certain degree of sensitivity. Note that this observation is common in binding site comparison tools since the true binding site region is difficult to define (Shaper retrieves only four MMP-13 entries in the top ranks). Residue-based methods are less susceptible in this regard since direct correspondences exist however, they are significantly worse at finding similarities between sites with a less obvious correspondence set. We expect that using molecular dynamics ensemble
representations of binding sites may also be a way to artificially generate structural diversity for more confident matching.

Strikingly the first non-MMP, ADAMTS5 is discovered at rank 12 (chi squared 0.86, 2RJP), indicating a high binding-site similarity between these two metalloproteases. Observations such as this are particularly important, since a simple sequence alignment would not be able to identify this relationship, due to a low global homology (11.7% identity). This also gives further weight to the hypothesis that cross-reactivity between MMPIs and the ADAM/ADAMTS enzyme families could be a reason for the lack of clinical trial successes to date. This hypothesis is further reinforced by the fact that all ADAM and ADAMTS enzymes (except ADAM2) were retrieved in the top 10% of results, and the iterative AUC score (Table. 4.12) for the ADAM17 isoform is 0.98. Activity-based protein profiling (ABPP) experiments with four MMP-13 inhibitors against a diverse set of metalloproteinases also reveal cross-reactivity between the compound (2-[(4-Phenylphenyl)sulfonylamino]pentanedioic acid PubChem CID: 644601, AID: 735) and the highest ranked non-MMP, ADAMTS5 as well as ADAM8 and various MMP isoforms\textsuperscript{277}. A RANSAC alignment (Figure. 4.17) between MMP-13 (1FM1) and ADAMTS5 (2RJP) reveals numerous conserved amino-acids including three key histidine’s bound to catalytic zinc.
Figure 4. 16 Virtual screening results for a collagenase-3 (MMP-13) binding site (1FM1) screened versus the sc-PDB database. Protein entries are ranked (left) by decreasing chi squared similarity and three key metalloproteinase families are highlighted: MMPs (blue cross), ADAMs (red cross) and ADAMTSs (green cross). The mean (µ), mean plus one standard deviation (µ + σ) and top 10% of ranks are indicated by red, purple and green lines respectively. The box-plot (right) illustrates the distribution of chi squared scores across the virtual screen, with the mean marked by a green triangle and the median marked by the orange line.
This particular alignment is also interesting since the defined binding site of 2RJP is larger than that of 1FM1 with the ligand extending deeper into the hydrophobic sub-pocket (S1’). Despite size difference 2RJP is ranked highly and aligns well indicating the algorithms relative insensitivity to definition, focusing on local overlap of similarity. Although studies have been conducted to determine the factors for MMP selectivity\textsuperscript{280}, these results suggest that both ADAM and ADAMTS should also be included in further studies. Cross-reactivity between MMP isoforms is potentially just as important as MMP-ADAM/ADAMTS cross-reactivity. A recent review of the protective roles of MMPs in tumour progression highlights MMP-3, 8, 9 and 12 as having ‘protective’ roles, in contrast to the initial hypothesis that all MMPs would be pro-tumorigenic\textsuperscript{278}. We note all of these MMP isoforms are retrieved amongst the highest ranks: 2, 19, 8 and 24 respectively (top 0.1%), with high iterative AUCs, where applicable (>0.90). High similarity between these sites emphasises the difficulty in designing selective inhibitors for a given MMP isoform. It is important to mention that while certain MMP isoforms may appear to me more similar to each other according to the chi squared metric, inferring selectivity from a ‘fuzzy’ representation such as BoF is a bad idea since often selectivity is due to minor local dissimilarities which may not be captured in enough detail to quantify accurately. A full selectivity analysis is out of scope for this work and has been conducted elsewhere\textsuperscript{280}. It would be an interesting avenue for further research to see if the learnt latent features could be useful for selectivity analysis through identifying regions within superimposed binding sites that are significantly different to each other. When designing inhibitors these different regions may be targeted appropriately.
<table>
<thead>
<tr>
<th>Metalloproteinase</th>
<th>Iterative AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMP-3 (stromelysin-1)</td>
<td>0.99</td>
</tr>
<tr>
<td>MMP-12 (macrophage metalloelastase)</td>
<td>0.96</td>
</tr>
<tr>
<td>MMP-13 (collagenase-3)</td>
<td>0.93</td>
</tr>
<tr>
<td>ADAM17</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 4. 12 Iterative AUC scores for metalloproteinase ($N > 10$) enzymes when performing a virtual screen using a collagenase-3 (MMP-13) enzyme binding site (1FM1) against the sc-PDB database.

Apart from metalloproteinases in the MMP, ADAM and ADAMTS families the virtual screening also retrieved a collection of other zinc metalloproteinases including: Snake venom metalloproteinase atrolysin-D, Snake venom metalloproteinase BaP1, Zinc metalloproteinase-disintegrin-like VAP2B and Karilysin despite a lack of sequence homology. The results clearly exemplify the ability of SurfsUp to identify distantly related enzymes based on local similarities. It is also interesting to analyse entries with unexpectedly high rankings. SurfsUp ranks the proteins: Camphor 5-monoxygenase (rank 56), Adenosine receptor A2a (rank 48) and Renin (rank 140) all within the top 1% with high iterative AUC ($>0.80$) on all accounts. On closer inspection of protein-ligand complexes within these classes it is clear that interactions with inhibitors are predominantly mediated by hydrophobic interactions. Ligands of the respective proteins: 3OIA, 3VG9, 3G6Z project motifs into deep, hydrophobic and concave regions of the binding sites. These local regions of similarity and general hydrophobicity and shape of the binding site
may well be the responsible for the high ranking of these proteins, since BoF does not take into account spatial relationships. RANSAC alignments between 1FM1 and the aforementioned proteins, fails to superimpose ligands in a reasonable way. Indeed Ehrt et al.\textsuperscript{175} question whether binding site comparison is appropriate for highly hydrophobic cavities where similarity may not always imply a common adopted binding conformation. For such cavities many feasible superpositions are potentially possible.

Figure 4. 17 A RANSAC surface alignment between MMP-13 (1FM1) and ADAMTS5 (2RJP)

Considering alignments fail for certain targets in the top 1\% of the screened dataset, an alignment-based re-scoring of the top 1\% is computed in an attempt to retrieve only the most similar sites. Alignments are computed with the RANSAC approach between 1FM1
and the top ranked hits from the virtual screen. Entries are ranked using the fitness score which measures the overlapping area after alignment (number of inliers divided by the number of points in target), the results are shown in Figure 4.18. The results show a clear distinction between proteins within the metzincin family and those that are not. The majority of metalloproteinases are recovered with a fitness $F > 0.5$, indicating a high quantity of vertex correspondences and thus a highly conserved shape. The lowest ranked result (fitness 0.26), is an MMP (MMP-12, 4H49) complexed with a large twin inhibitor (bi-functional) formed by chemically linking two different ligands. The large size increase of the binding site definition due to the larger ligand (4629 vs 2129 vertices) is clearly reflected by the alignments’ fitness. Intriguingly, despite high rankings with the BoF descriptor members of the ADAMTS family are retrieved at lower ranks than the majority of MMPs/ADAMs, we postulate that this occurs due to compositional changes in the of the ‘S-loop’ which significantly impacts the shape of the binding site surface²⁸⁰. Perhaps differences within this loop region may be targeted in order to obtain selectivity to particular isoforms. The results of this case study indicate demonstrate the abilities of the SurfsUp algorithm, but also illustrates that in many cases a structural superposition may be required to obtain a more accurate ranking of hits. This is especially true in the case of predominantly hydrophobic cavities where global shape may not be present and hence many possible correct superpositions.
Figure 4. 18 Re-scoring of virtual screening results for a collagenase-3 (MMP-13) binding site (1FM1) screened versus the sc-PDB database. Protein entries are ranked (left) by decreasing alignment fitness and three key metalloproteinase families are highlighted: MMPs (blue cross), ADAMs (red cross) and ADAMTSs (green cross). Other proteins are marked by a black cross. The mean ($\mu$) and mean plus one standard deviation ($\mu + \sigma$) are indicated by red and purple lines respectively. The box-plot (right) illustrates the distribution of fitness for all pairwise comparisons, with the mean marked by a green triangle and the median marked by the orange line.
4.3.6 Run-time analysis

Runtimes were collected using a computer equipped with an Intel® Core™ i9-9900K (3.6 GHz), 64 GB of memory and an NVIDIA® GeForce® RTX 2080 Ti GPU.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Average Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface computation (Inc. properties)</td>
<td>13.45±4.23</td>
</tr>
<tr>
<td>Surface patch extraction</td>
<td>0.024±0.040</td>
</tr>
<tr>
<td>BoF calculation</td>
<td>0.057±0.060</td>
</tr>
<tr>
<td>Pairwise comparison</td>
<td>0.0000625</td>
</tr>
</tbody>
</table>

Table 4. 13 Runtimes for each stage of a SurfsUp computation. Note that the first three steps are usually performed before performing virtual screening experiments and the results saved. Virtual screening experiments are thus rapid, computing

4.4 Conclusions

Quantifying or locating similarities between pairs of protein binding sites may assist with multiple stages of the drug discovery process, particularly in the identification of off-target interactions or functional annotations. When targets are annotated with ligand data similarities may also aid fragment-based drug design, suggesting novel scaffolds or potential bioisosteric replacements. In this work we propose an novel approach for binding comparison, SurfsUp, which uses an unsupervised, point cloud autoencoder, for learning local, rotation invariant surface features. Learnt, or latent, features are paired with a bag-
The SurfsUp method was validated on biologically relevant problems, demonstrating the ability to classify enzymes belonging to the same enzyme classification, identify synapsin-1 as a target for kinase inhibitors as well as identify relationships between distant members of a protein family with a similar active site. The case studies demonstrate the potential utility of the method during early drug discovery where the speed and superior retrieval ability make interacting with data in real-time possible.
Future work may focus on optimizing the neural network architecture, especially in terms of efficiency. We believe that smaller latent features will still be able to retain a significant level of information required for the desired task. Lower surface resolutions will also help in this respect. Another angle of research may consider a more robust surface point sampling scheme when computing BoF features, picking the most distinctive surface features. Experimenting with further applications is an interesting angle, and we can foresee use in protein-protein interaction prediction (larger patch radius), or the application to small molecule surfaces for bioisosteric replacement suggestion. One may wish to use the latent features downstream or in a transfer-learning based approach. In regards to binding site matching, combination with other methods in a meta predictor style approach could be useful using a consensus-based scoring for increased ranking confidence. The future of deep-learning in structural learning is exciting and we hope to observe novel developments, especially applied to the molecular surface which to date has been relatively underused.
Chapter 5: ScaffoldGraph Software

5.1 Introduction

A key component of the drug discovery pipeline is partitioning molecular data into subsets of structurally related entities. Chemical space partitioning is essential considering that pharmaceutical patents cover structural classes as opposed to individual molecules, and that the similarity-property (SPP) states, structurally similar molecules are more likely to share similar properties than dissimilar ones\(^5^9\). With this in mind a structure-based classification scheme should generate structural classes in which a property of interest is largely homogenous when compared to a class produced by a random partitioning scheme. Chemical structure derived partitioning schemes broadly fall into one of two categories: descriptor-based or rule-based. Descriptor-based, or molecular-fingerprint, schemes utilise vector representations of molecules which may be clustered through the use of well-established statistical clustering algorithms. Ideally the distance between two molecular descriptors should also reflect the difference in molecular properties. Rule-based classification schemes rely on handcrafted rules, defining distinctive structural features, determined by expert knowledge. Molecular scaffolds are one such example of such a scheme, with the most famous being the molecular-framework devised by Bemis and
The “Bemis and Murcko scaffold” (BM) is simply constructed through the removal of terminal side chains from a molecular structure. Rule-based schemes have the advantage that they are data-agnostic, deterministic and each class center corresponds to a chemically intuitive sub-structural entity or “chemotype”.

Despite the benefits afforded by a simple scaffold-based representation, the simplicity is occasionally problematic since there is an ambiguity over what portion of the molecule represents the “core-structure” or most descriptive element. This ambiguity means that simple BM scaffold-based partitioning structures generate a very large number of distinct clusters. Hierarchal based scaffold representations\textsuperscript{146} were proposed as a potential solution to this problem where all, or an intuitive subset, of a molecule’s potential scaffolds are considered. The most well-known of these representations is the scaffold tree\textsuperscript{146} which defines “chemically intuitive” rules to remove peripheral, linked, or heteroatom-deficient rings first and retaining, heteroatom-rich and fused rings as a “core structure”. The benefit of this approach is that scaffolds contacting the same core structure, yet different cyclic substituents will be recognised as related in contrast with the simplistic BM side chain flattening scheme. Hierarchical based scaffold representation can be used for structural classification, but also can be used for active-series identification\textsuperscript{282}, database diversity analysis\textsuperscript{283} and the identification of overrepresented structural classes\textsuperscript{284}.

Despite the benefits of hierarchical scaffold relationship structures, implementation is reasonably difficult, hence software solutions are limited and commonly proprietary, limiting their use in many settings. Open-source solutions are limited, have many limitations or are geared toward a specific usage application. Here we present ScaffoldGraph: an open-source Python package and command-line tool for the generation
and analysis of hierarchical scaffold relationships, with the capability of processing large sets of input molecules. Intended for use within drug discovery pipelines, ScaffoldGraph extends existing implementations, increasing the flexibility and accessibility of this type of analysis whilst also improving performance.

5.1.1 Molecular Scaffolds

The molecular scaffold is a core concept in medicinal chemistry, commonly used to represent the core structure of molecules upon which functional groups can be attached. The molecular scaffold plays a significant role in determining global molecular properties and the basic shape and geometric orientation of substituents. As such, scaffolds form an important part of modern drug discovery, especially in techniques such as parallel synthesis, where combinatorial compound libraries are constructed around various central ring-systems, or in “scaffold-hopping” where the objective is to discover a new molecular structure through the modification of the core structure. The term “scaffold” is often used synonymously with other terms, including “framework”, “fragment”, “chemotype”, or “substructure”, however, these terms are often used to describe other types of structures. For example, a fragment may represent a particular portion of the core structure and include various substituents or “R-groups”. For the purpose of this document, the term “scaffold” is used when referring to the core structure of a molecule or series of molecules. The interest in molecular scaffolds stems from a chemical-series centric approach to drug discovery, in which an initial hit compound is subjected to various modifications in order to optimize a set of relevant chemical properties. While considering
one chemical series, defining a core structural unit is fundamentally irrelevant since all
structures share the same core unit. However, when considering sets of chemical series or
organizing large sets of heterogeneous molecules, one may benefit from defining molecular
scaffolds so that datasets can be classified and organized clearly and reproducibly\textsuperscript{287}.

Molecular scaffolds have been shown to exhibit a clear association with a compound’s activity towards a particular biological target. In particular, Evans et al. \textsuperscript{288} observed in 1988 that certain (‘privileged’) structures may be used as templates for
derivatization to discover novel ligands binding to multiple targets. This seminal work
identified benzodiazepine and substituted indole as key scaffolds for generating
Cholecystokinin A Receptor (CCK) agonists. Over time the privileged scaffold definition
has been interpreted in numerous different ways and may also refer to a scaffold that,
through judicious modification, yields molecules that are preferentially active against a
particular target with high affinity\textsuperscript{289}. These so-called privileged structures can thus
establish the starting point for generating chemical screening libraries.

While fundamentally the idea that a particular scaffold may exclusively react with
only one type of target or target family is questionable, as we now know the traditional
pharmacological view of “one molecule, one target” is false, preferential binding patterns
of certain core structures to particular targets has been noted. Indeed, Welsch et al. in 2010
\textsuperscript{290} published a broad review of privileged scaffolds compiled from medicinal chemistry
literature, including the classical case of the benzodiazepine scaffold found in many ligands
of ion channels and G-protein coupled receptors (GPCRs). Clearly, while a scaffold may
not guarantee selectivity to a particular biological target, it may contain favorable
pharmacophoric features for a particular receptor or preferentially orient pharmacophoric functional groups in a way that is conducive to molecular recognition.

The link between scaffold and activity may also be observed from a different perspective. Instead of identifying a single scaffold with the ability to generate highly selective and active compounds, one may identify structurally distinct molecules sharing similar activity rather than structure. This objective is referred to frequently as “scaffold hopping”\textsuperscript{148}. In this process, the core structure of a molecule is replaced with another to optimize its biological properties. It is also worth noting that the key idea behind virtual screening is to predict novel active molecules in a screening library given a set of molecules with known activity. A key measure of success for such a campaign is identifying molecules with novel structural classes and high measured activity in subsequent validation experiments. Evidently, there is a clear link between molecular scaffolds and bioactivity whether it be directly, via shape or binding interactions, or indirectly through judicious modification. It is important to note however that when associating bioactivity values with scaffolds, it is often the case that the term “active scaffold” refers to the compounds derived from the specific molecular core, and does not indicate that the scaffold itself has intrinsic activity to the target, which should be determined individually.

5.1.2 Scaffold Definition

Despite the liberal application of scaffold theory in medicinal and computation chemistry, the actual definition of a scaffold is not particularly well defined across the literature, with multiple representations being considered\textsuperscript{287}. For example, a scaffold may be calculated for
a chemical series by determining the entire collection’s maximum common structure (MCS). Although for extensive collections of heterogenous molecules, this method is computationally intractable and thus unfeasible for a large proportion of applications. Due to the subjective nature of defining a scaffold, the choice of definition ultimately depends on the specific application. For example, the Markush representation is applied in medicinal chemistry patents, where the scaffold is depicted with multiple independently variable groups (R-groups). This generalized depiction is used to protect intellectual property where a company makes a general claim for the usage of the core structure without revealing the exact molecular structure to competitors. Although varying definitions are useful in such applications, the lack of a consistent definition in cheminformatics presents a reproducibility issue when comparing various studies. This intrinsic subjectivity is easy to explain due to the ambiguity over which portion of the molecule is considered the “core structure”.

In a seminal article published in 1996, Bemis and Murcko\textsuperscript{281} introduced the commonly recognized hierarchical organization scheme dividing molecules into three groups: ring systems (frameworks), linkers and R-groups, where a linker defines a continuous sequence of atoms forming a connection between two or more ring systems. The well-known “Bemis and Murcko (BM) scaffold/framework” is thus the union of the ring systems and the linkers connecting them, disregarding the various R-groups the molecule may contain. The authors analyzed a collection of 5120 known drugs, identifying 2506 different scaffolds where 42 of the most frequent scaffolds represented a quarter of the entire collection, this is unsurprising when most known drugs are derived from natural products. From the BM scaffold, a “cyclic skeleton” (CSK) or “graph-framework” can be
derived ignoring atom type, hybridisation and bond order, representing the topology of a particular molecule. The CSK scaffold may be further generalised by a deeper level of abstraction obtained by normalising all rings and linkers to unit size yielding “reduced cyclic skeletons (RCS)” or “Oprea scaffolds”\textsuperscript{291}. In the Bemis and Murcko article, the authors found that approximately fifty percent of the known drug collection was represented by thirty-two of the most common CSKs generated, indicating that the diversity of drug “shapes” was relatively limited.

Utilising the hierarchical definition of BM scaffolds, CSKs and RCSs has provided a consistent and generally applicable, efficient approach for scaffold generation and organisation and is thus ubiquitous in the cheminformatics literature. Despite its widespread adoption, the BM scaffold is not without its problems, for example, when classifying chemical series, unique BM scaffolds may represent similar structures that may only differ by the change of a single heteroatom or bond order, the CSK is thus often considered when a more abstract view is required. The primary issue, however, is that the addition of a new ring to the structure, as per definition, will create a distinct BM scaffold (although the molecule in these cases may be better considered an analogue), complicating tasks such as the assessment of scaffolds for combinatorial chemistry. The border between what constitutes a molecule’s core and a substituent is especially fuzzy; a medicinal chemist may disagree with another complicating computation representation that must remain consistent. Scaffold organisation/classification schemes have been proposed to combat this ambiguity, which rather than focusing on one core structure, emphasise a collection of organised scaffolds generated by a set of chemical transformations\textsuperscript{146}, or through the removal of peripheral rings and linkers. The generated scaffolds from a set of
compounds create a hierarchical scaffold relationship structure which can be analysed in a manner that removes ambiguity resulting in the selection of a single representative scaffold.

5.1.3 Hierarchical Scaffold Relationships

As a way to reduce ambiguities introduced through the use of the BM scaffold, various hierarchical structural organisation/classification schemes based on sub-structural relationships have been proposed, including HierS\textsuperscript{284}, the scaffold tree (ST)\textsuperscript{146} and the scaffold network (SN)\textsuperscript{292} (Figure 5.1). These approaches are all conceptually similar in that they compute a graph-structured systematic organisation of BM scaffolds through the iterative removal of rings/ring systems from the original scaffold. In all cases, the original BM scaffold definition is extended to include exocyclic double bonds and double-bonded substituents attached to linkers. This BM extension is commonly the default approach taken by modern cheminformatics software packages since exocyclic double bonds have an effect on the aromaticity of ring-systems.

The HierS approach\textsuperscript{284} begins with the BM scaffold and iteratively removes ring systems, where a ring system is defined as a set of atoms and bonds in which every atom and bond is a member of a cycle; in this sense, polycyclic/fused ring systems are not dissected further. Through repeated cycles of recursive peripheral ring removal, an acyclic directed network graph is created with scaffolds of decreasing size forming a hierarchical relationship. The authors use the constructed hierarchy to identify over-represented structural features through hierarchy traversal, removing compounds in a given set without sacrificing the structural diversity landscape. The ST algorithm, also begins with BM
scaffolds, using a “leaf-to-root” decomposition scheme in which peripheral rings are removed from the BM scaffold. The decomposition is guided by a set of predefined, “chemically intuitive” prioritisation rules, selecting a parent scaffold from the generated scaffold pool. These prioritisation rules can be seen as defining the “most-peripheral” ring at a given stage of decomposition. This process continues creating branches of structural relationships until only a single ring remains, resulting in a directed tree structure. A collection of scaffold trees may form a larger tree structure in the case where there are no disconnected trees, or a forest of trees where larger trees are disconnected (i.e. do not share a common scaffold).

Due to the decomposition schemes used in a hierarchical organisation, these data structures often contain “virtual scaffolds” representing scaffolds that are not a direct BM scaffold of any molecule in the collection. Virtual scaffolds may represent starting points for the synthesis of new compounds or be used for activity prediction where virtual scaffolds are annotated with the same activity as their nearest “real” neighbour in the hierarchy. Datasets of unknown activity may then be searched using prioritised virtual scaffolds as a substructure query, identifying candidate molecules for further experimental validation. Similarly, STs generated from natural product sets, and bioactivity annotated datasets can be merged by combining leaf-to-root pathways. The resulting structure may be used for prospective activity annotation identifying potential lead compounds.

In addition to activity prediction, the ST structure can be utilised to identify chemical series where the underlying activity distribution is significantly different from the background distribution. The distributions can be compared using non-parametric statistical tests in a process coined compound set enrichment (CSE) through analogy to
gene set enrichment used in microarray analysis\textsuperscript{282,292}. Intuitively scaffolds within the hierarchy represent a chemical series in which all compounds of the series contain that particular scaffold. Hence, the activity of such series’ can be mapped back to the ST structure, making it possible to observe how activities distribute across the hierarchy. Therefore, the statistical significance of a scaffold node represents its perceived ability to generate molecules more active in comparison to the global background activity. Significantly enriched scaffolds may be further explored as part of a drug discovery campaign. It is important to note that, while this method can be used to identify active series, there is no claim that the scaffold itself is responsible for the enrichment of bioactivity, which can be attributed to structural features across the whole molecule, including terminal side chains. However, if the identified scaffold contains or displays relevant pharmacophoric features in an optimal geometric arrangement, this particular scaffold may be preserved as a common structural core for a series of active compounds, analogous to the privileged scaffold concept. By focusing evaluations around prioritised compound series, rather than specific hit compounds, selected from high-throughput screening (HTS) campaigns, the opportunity to identify novel structure-activity relationships (SARs) is increased.

Inspired by the finding that the majority of compounds active against human targets display substructure relationships and/or are topologically equivalent, Hu and Bajorath\textsuperscript{294} extended ST analysis incorporating “leaf-to-leaf” (non-hierarchical) substructure relationships. A leaf-to-leaf relationship represents the similarity between a pair of scaffolds rather than a direct hierarchical relationship. Indeed, it is also possible that some of these relationships will already be present in the “leaf-to-root” hierarchy. The authors
suggest that this information enriched structure aids in selecting virtual scaffolds for activity prediction. Virtual scaffolds are prioritised based on whether their nearest neighbour scaffolds are “active” or involved in sub-structure relationships with other scaffolds through leaf-to-leaf relationships.

It is evident that including more information about sub-structural relationships found within collections of molecules will improve the ability to make reasoned decisions. To this end, the scaffold network approach (SN) \(^{292}\) was created to improve the CSE active-series identification protocol. The improved representation and CSE algorithm were shown to identify approximately twice as many statistically significant active scaffolds compared to the scaffold tree-based approach. The SN approach can logically be seen as a combination of both the ST and HierS methods, where the decomposition is exhaustive, generating the set of all possible decompositions through the removal of peripheral rings (i.e. also dissecting fused ring systems). Networks constructed from multiple molecules can be aggregated to form a large multipartite directed acyclic graph (DAG). The SN deals with the ambiguity of determining the “most peripheral” ring, through predefined chemical rules. Although the ruleset used for ST construction was created with the idea that the root scaffold should represent the most chemically characteristic scaffold for the parent, this may not hold in all cases and is intrinsically subjective (a medicinal chemist may disagree with this assignment). Despite the improvements of the SN approach and its success in active-series identification, the SN has yet to be as widely adopted in the scientific literature (71 citations vs 332 citations, information from google scholar, accessed 09/06/2021), perhaps this is due to the increased complexity of analysing a large DAG structure versus a more intuitive tree structure, or the fact that the tree structure lends itself better for
clustering and visualization of data due to the decreased complexity. It may also be argued that dissecting fused ring systems is not always a valid approach; for example, the steroid scaffold is a large fused ring system with no identifiable “peripheral” member. As with chemical similarity computations the best approach should be selected based on the problem in question.

Figure 5. An illustration of various hierarchical scaffold relationships constructed from the molecule, Amodiaquine. The scaffold tree decomposition$^{146}$ follows the dotted arrows ending with pyridine while the scaffold network$^{292}$ incorporates all drawn routes. The HierS approach$^{284}$ does not decompose ring systems and thus only explores the routes marked by green arrows.
5.2 ScaffoldGraph

ScaffoldGraph is an open-source (MIT licensed) Python library with a command-line interface with the ability to build and analyse hierarchical scaffold relationship structures. The software was developed based on the need for a consistent approach for clustering molecules without, the many singletons as produced by fingerprint-based clustering algorithms and in a way, which is easy to visualize. Despite extensive use of hierarchical scaffold relationships in the cheminformatics literature, the approach is difficult to implement and thus there is a lack of open-source implementations. The existing open-source implementations, Scaffold Network Generator (SNG) and Scaffold Hunter present many limitations which limit their use in practice. SNG provides a command-line utility for both ST and SN generation, but does not include HierS networks and is limited to processing 10,000,000 molecules due to the memory constraints of the Java virtual machine (the source code of SNG is also currently not available). On the other hand, Scaffold Hunter is a graphical application which is aimed at visualisation of compound collections as opposed to calculating scaffold relationships explicitly. The nature of the application limits its use when one wishes to perform bespoke analysis or use protocols that are not implemented in the software. Furthermore, the graphical user interface of ScaffoldHunter is limited to 200,000 molecules which may be insufficient for high-throughput applications. ScaffoldGraph aims to extend previous implementations with additional features and through the provision of a library of methods for the construction and analysis of “scaffold graphs”. The software emphasises flexibility and extensibility, integrating into the Python scientific stack, enabling efficient data analysis and the ability to integrate into drug discovery pipelines.
5.2.1 Technical Details

ScaffoldGraph is built upon the popular open-source cheminformatics package, RDKit\textsuperscript{132}, providing utilities for working with graph-based representations of molecules. The package is used to read data from chemical file formats and for fragmenting molecules in hierarchical decomposition processes. The graph-structure of scaffold relationships inherits from the NetworkX\textsuperscript{298} Digraph object providing a familiar interface while implementing methods specific to exploring these specialised structures. Scaffold nodes are represented in the graph by a hash of their canonical SMILES string representation while molecule nodes are represented by a hash of the molecules identification or if not provided a generated key. Thanks to Python’s dynamic typing system, arbitrary data can be assigned to both types of nodes and also edges between them simplifying algorithm implementation. ScaffoldGraph objects are not limited to containing two types of nodes; a user may find it useful to add other nodes such as nodes representing protein targets. Each specialised representation inherits from the ScaffoldGraph object further enabling the addition of representation specific methods and properties. This style of architecture also makes it relatively easy for a user to implement their own scaffold relationship graph representation within the confines of the software. ScaffoldGraph can be accessed from the GitHub repository\textsuperscript{299} or installed using the Python package installer, Pip, or the Python dependency management software Anaconda.
5.2.2 Features

The ScaffoldGraph software has the ability to compute multiple forms of hierarchical scaffold relationships and also contains generic analysis tools which can be applied to multiple different drug-discovery problems. The software emphasises flexibility enabling integration into existing or new drug-discovery pipelines.

5.2.2.1 Representations

ScaffoldGraph has the ability to compute three types of hierarchical scaffold relationship structures natively. These include: the scaffold network\textsuperscript{292}, scaffold tree\textsuperscript{300} and HierS\textsuperscript{284} representations. All representations take the form of a directed acyclic graph (DAG) where relationships between scaffolds or molecules and scaffolds have an associated direction (parent $\rightarrow$ child). A tree structure is simply a restricted form of a DAG, with the restriction that a child node may only belong to one parent (no cycles may exist). The graph structure enables fast depth and breadth-first traversal across nodes and edges.

Scaffold prioritisation rules, for scaffold tree generation, follow that of the original publication\textsuperscript{146}. However, a set of generic and, customisable rule bases are provided so that a user has the ability to control prioritisation to their requirements. There is no limitation to how many molecules can be processed, opposed to existing software, other than the size of available memory. ScaffoldGraph has been used to process 280 million molecules from Enamine REAL\textsuperscript{301}, substantially more than existing tools are capable of. In a typical use case memory usage is non-linear since only unique scaffolds are retained in the data structure.
5.2.2.2 Analysis

ScaffoldGraph is packaged with a set of general analysis tools that can be applied to a broad set of applications or are commonly used in the literature. Due to the flexibility of the software more bespoke analysis protocols may be implemented on a case-by-case basis. In brief the package contains tools for computing: compound set enrichment (CSE) from HTS data\textsuperscript{292}, cumulative scaffold frequency\textsuperscript{283}, over represented scaffold class identification\textsuperscript{284} and scaffold class diversity selection.

5.2.2.3 Visualisation

Visualisation is key to interpreting results from any form of predictive modelling. When prioritizing scaffolds, it is critical that the structural landscape can be interpreted easily. Analysing the landscape around a prioritized entity may reveal the structural changes that lead to the perceived increase in activity of a chemical series. Comparing prioritised scaffolds and their associated series’ may open the door for scaffold-hopping opportunities. While not specifically designed for visualisation, unlike a software such as Scaffold Hunter, ScaffoldGraph provides an interactive visualisation tool for use in Jupyter, based upon the Cystoscape graph visualisation library\textsuperscript{302}. 
Figure 5. A screen capture of ScaffoldsGraphs Cytoscape-based visualisation tool which functions within a Jupyter notebook environment. Here Molecules are coloured yellow and scaffolds are coloured blue. A user can customise colours based on embedded node properties such as bioactivity. The above graph was constructed from randomly extracted assay data and truncated to scaffolds and molecules that are decedents of the scaffold shown on the left hand side (Naphthyl-Oxazole).
The tool is able to generate visualisations of any scaffold graph or scaffold graph subgraph, with nodes containing depictions of the corresponding chemical structure. The graph layout and style are highly customisable and can be tailored toward a particular requirement. For example, the background colour of a node may be coloured corresponding to a particular property. In the case of a CSE experiment the colour may reflect the calculated significance for a particular scaffold Figure 5.2.

5.2.2.4 Miscellaneous

ScaffoldGraph contains various miscellaneous function which may be useful to a user of the software, from selecting subgraphs based on properties to generating bipartite representations of scaffold hierarchies. The most notable feature however is the generation of further scaffold representations which may be added to an existing graph or used as a foundation to build new hierarchy representations. The software includes the ability to generate cyclic-skeletons (CSKs) and reduced cyclic-skeletons (RCSs) also commonly referred to as “Oprea scaffolds” or ring-topology scaffolds.

5.2.2.5 Command Line Interface

ScaffoldGraph provides a command-line interface (CLI) which is analogous to the interface provided by the Scaffold Network Generator (SNG), allowing drop-in replacement into existing drug-discovery pipelines. The process consists of two stages: generate and aggregate. The generation step produces intermediate scaffold graphs (scaffold tree, scaffold network, or HierS) from SDF or SMILES inputs which can then be
aggregated into a final combined graph representation which can then be queried to select only scaffolds of interest. The two-stage process allows multiple intermediates to be calculated in parallel before aggregation, reducing the total computation time required. At the generation phase users also have the option to change the ring filter size (default removes molecules with more than 10 rings), remove specific isotope information, remove disconnected fragments and discharge and deradicalize molecules before processing. The final output is a TSV file consisting of four columns: a numerical scaffold identifier, the number of rings in the scaffold (hierarchy), the scaffold SMILES string and the identifiers of sub-scaffolds (parents). Instead of the TSV output a user may specify to output results in an SDF format with the title set to the numerical identifier and the sub-scaffold identifiers stored in the properties portion of the file.

5.2.3 Benchmarking

The ScaffoldGraph software was benchmarked and compared with the existing software Scaffold Network Generator (SNG)\textsuperscript{296}, since it objectively the most similar software. Single process benchmarks show that ScaffoldGraph is approximately 2x faster than SNG on equivalent datasets; a considerable difference when analysing high-throughput data sources.
5.2.3.1 Technical Details

Benchmarks were performed using molecules extracted from a “drug-like” subset of the ZINC15 database\textsuperscript{303}. All molecules with greater than 10 rings were removed since SNG filters these by default and larger ring systems take a disproportionately longer amount of time to process. The final dataset amounts to a non-redundant set of 9,604,669 molecules. Benchmarks are repeated over random subsets of the data to simulate input diversity. All benchmarks were performed on an Intel Core i9-9900K 3.6 GHz with 64 GB of RAM, using ScaffoldGraph version 1.1.2, RDKit version 2021.09.03 and NetworkX version 2.6.3.

5.2.3.2 Results

In the first benchmark we compared the performance between the different algorithms using varying sized sets of randomly sampled molecules. The results are shown in Figure 5.3. Both the scaffold tree and HierS display similar performances while the scaffold network is significantly less performant. When using a sample of 100,000 molecules the scaffold network begins to show a larger standard deviation over runs. This occurs because the scaffold network takes significantly longer to run on molecules with a greater number of rings due to the recursive algorithm used.
In the second benchmark samples of 100,000 molecules were used to construct each representation, using a parallelised constructor. Parallelisation over one, two, four, eight and sixteen cores were attempted. The results are displayed in Figure 5.4. The use of multiple cores has a significant benefit when constructing scaffold graphs, almost quartering the time required when applying four cores. With more than four cores the effects are less substantial although with a larger dataset the benefits will be more pronounced.
Figure 5. Performance benchmark for the scaffold tree, network and HierS algorithms implemented in ScaffoldGraph, parallelised over multiple CPU cores.

For the final benchmark, ScaffoldGraphs performance is compared to SNG, computing a scaffold network from a random 150,000 molecule subset. ScaffoldGraph took 15 minutes 25 seconds, while SNG took 27 minutes 6 seconds. When parallelised over four cores ScaffoldGraph took just 4 minutes 29 seconds. The performance benefit of ScaffoldGraphs implementation is significant, especially when analysing large numbers of molecules.
5.3 Case Studies

To present the utility of the ScaffoldGraph software two case studies are presented. The first case study presents the use of ScaffoldGraph in a pharmaceutical industry setting and the second exemplifies the flexibility of the software, demonstrating a protocol for the identification of privileged and target-family privileged scaffolds using bioactivity data extracted from the ChEMBL database\textsuperscript{49}. Both case studies demonstrate the utility and application of the software in both industrial and academic research projects.

5.3.1 Diverse Compound Selection

As previously emphasised, methods for measuring molecular similarity and diversity are of paramount importance in several stages of the drug discovery pipeline. The success of high-throughput screening (HTS) campaigns, for lead identification, has a strong dependence on the quality, diversity and quantity of molecules that are selected. \textit{In-silico} diversity selection approaches are thus frequently applied when constructing screening libraries, based on the assumption that a diverse set of molecules will increase the chance of recovering bioactive ‘hit’ compounds. Diversity selection techniques rely on molecular descriptors, rule-based classification schemes or combined approaches for portioning collections of molecules into structurally related compound classes. Compounds can then be selected, from each class with less structural redundancy, reducing the cost and time required for screening. Ligand-based virtual screening methods tend to be high-throughput in nature and therefore are less reliant on an initial selection procedure. However, with the
advent of very large enumerated chemical databases, containing in the order of billions of molecules, more complex, 3D based ligand virtual screening approaches become intractable.

Orion Pharma, developed a diversity selection pipeline based on the ScaffoldGraph software for enumerating a central nervous system (CNS) related compound screening library derived from the Enamine REAL enumerated library (1200 million compounds). Orion use the Cresset Blaze software for virtual screening, comparing the 3D shape and electrostatic properties of molecules, which has the ability to screen 1.75 million molecules per-hour. The throughput of Blaze is impressive, yet to screen the entirety of Enamine REAL would take approximately 29 days which is unacceptable for high-throughput experiments. The developed pipeline reduces the compound pool from 1200 million to 32 million, an amount that can be screened in a much more manageable time of approximately one day.

The pipeline begins with a simple set of chemical property and substructure-based filters, relevant to CNS drug targets, reducing the molecule pool from 1200 million to 280 million. From this reduced collection, a scaffold-tree based clustering algorithm is employed with each molecule being assigned a level one or level two scaffold (one-ring or two-ring scaffold), where level one scaffolds are used for molecules with only two resultant scaffolds after decomposition. The classification procedure yields 773,000 scaffold classes where 86% of classes contain 50 molecules or less and the remaining contain from 51 to over a million molecules.
Figure 5. An illustration of the pipeline used by Orion Pharma for selecting a diverse CNS relevant subset from the Enamine REAL enumerated database.
Scaffold classes containing less than 50 molecules were aggregated into a pool of approximately 5 million molecules and a MaxMin diversity picking algorithm, using Tanimoto distance between ECFP4 fingerprints\textsuperscript{74}, was applied, adding diverse molecules to the final pool. For each remaining scaffold class, the diversity picking algorithm was applied independently and the resultant molecules added to the final pool. At the end of the selection process a pool of 32 million molecules remains, representing the diverse, CNS relevant reduced screening library which may be screened in a reasonable time compared to the billion-molecule Enamine collection. The full pipeline is shown in Figure. 5.5.

### 5.3.2 Privileged Scaffold Identification

The original privileged scaffold (PS) concept, introduced by Evans in 1988\textsuperscript{288}, describes structural elements that are found in ligands across multiple, diverse targets. The privileged scaffold concept has been interpreted differently across the literature since its introduction and often is used to describe scaffolds that lead to specificity towards targets of a particular protein family or even a particular target receptor\textsuperscript{289}. This divergence in interpretation can easily be explained given that avoiding off-target interactions early in the drug discovery process is paramount to avoiding downstream complications. Scaffolds with the perceived ability to target receptors of a specific protein family selectively are often referred to as target-family privileged scaffolds (TFPS). Historically TFPS have been identified through tabulating common scaffolds and their intra-family occurrences. This method is insufficient since scaffolds that occur with high-frequency among ligands targeting a particular family may not be truly selective for that family.
Taking inspiration from the CSE method\textsuperscript{202} a protocol was designed for the identification of ‘privileged’ scaffolds within protein families in terms of bioactivity, identifying scaffolds enriched within multiple families. The assumption made is that a “true” privileged scaffold should lead to molecules enriched for high bioactivity when compared to activity measurements taken from molecules derived from non-privileged scaffolds i.e. the background distribution. Molecules enriched for activity across targets only belonging to one family may represent TFPS, whereas, molecules enriched for activity across a diverse set of targets belonging to multiple protein families may represent promiscuous scaffolds in line with the original concept. To demonstrate the protocol, bioactivity data (IC50) for five major drug targets was extracted from ChEMBL\textsuperscript{49}. The five targets include: protein kinase (PK), G-protein coupled receptors (GPCR), voltage-gated ion channels (VGIC), nuclear hormone receptors (NHR) and ligand-gated ion channels (LGIC). For each family, level two ChEMBL subfamily classifications were retrieved for each target. All activity data was negative log normalised (pIC50) and duplicates and outliers removed. Remaining molecules were used to build a scaffold network representation using the ScaffoldGraph software, with extra nodes added representing protein targets and protein-family annotations. Molecule → target edges were each annotated with the corresponding bioactivity.

CSE was applied iteratively to scaffolds belonging to each protein subfamily. For each subfamily the background distribution is established by aggregating all pIC50 data for assays within the corresponding subfamily. For each scaffold a second probability distribution is constructed from pIC50 values of children with assay data belonging to the protein subfamily. Both probability distributions are transformed into empirical cumulative
distribution functions (ECDFs) and tested for equality using a one-sided non-parametric Kolmogorov-Smirnov (KS) statistical test. P-values are computed for each scaffold using the KS statistic $D_{\text{max}}$ (largest difference between ECDFs):

$$p = D_{\text{max}} \sum_{j=0}^{[n(1-D_{\text{max}})]} \left[ \binom{n}{j} \left(1 - D_{\text{max}} - \frac{j}{n}\right)^{n-j} \left(D_{\text{max}} + \frac{j}{n}\right)^{j-1} \right]$$

A small KS statistic or high p-value means that the null hypothesis should not be rejected in favour of the alternative. In this case the alternative hypothesis is that the scaffold is linked to bioactivity. The p-value not only reflects the difference between distributions but also the confidence based on sample size. Bonferroni multiple-hypothesis testing correction was applied to each scaffold hierarchy independently since scaffolds at different hierarchies cannot be considered independent due to the fact that relationships between hierarchies implies conserved structural elements. The correction was applied as the critical significance level (0.01) divided by the number of scaffolds in the corresponding hierarchy. The results of the analysis are displayed in Table 5.1.
<table>
<thead>
<tr>
<th>Superfamily</th>
<th># Subfamilies</th>
<th># Molecules</th>
<th># pIC50</th>
<th># Scaffolds</th>
<th># Privileged</th>
</tr>
</thead>
<tbody>
<tr>
<td>P. Kinase</td>
<td>7</td>
<td>95,767</td>
<td>180,442</td>
<td>71,583</td>
<td>1,859</td>
</tr>
<tr>
<td>GPCR</td>
<td>3</td>
<td>62,979</td>
<td>105,368</td>
<td>63,933</td>
<td>988</td>
</tr>
<tr>
<td>VGIC</td>
<td>5</td>
<td>23,208</td>
<td>31,685</td>
<td>22,576</td>
<td>418</td>
</tr>
<tr>
<td>NHR</td>
<td>6</td>
<td>12,184</td>
<td>20,417</td>
<td>11,200</td>
<td>184</td>
</tr>
<tr>
<td>LGIC</td>
<td>10</td>
<td>9,740</td>
<td>16,384</td>
<td>6,095</td>
<td>119</td>
</tr>
</tbody>
</table>

Table 5.1 Results of the privileged scaffold identification approach, displaying the number of subfamilies for each super family, the number of molecules, bioactivity data points, scaffolds and identified privileged scaffolds within the superfamily.

The number of privileged scaffolds reflects the number of molecules within each family. While it is true that some protein families may be harder to drug than others, the metric used is independent of any form of bioactivity thresholds. We perform a deeper analysis on two identified scaffolds (Table 5.2): biphenyl and biphenyl-tetrazole (Figure. 5.6). Both have been reported as privileged within the literature and thus represent an interesting validation. Biphenyl is identified within three protein superfamilies: GPCR, NHR1 and VGIC within the subfamilies: GPCRA, NHR1 and VGSC respectively. The analysis is confident about the GPCR assignment with a low p-value well below the critical threshold. The assignment to NHR and VGIC is less confident since the number of activity points analysed is significantly lower. From the observation it is tempting to label biphenyl as a privileged scaffold within the confines of the original concept. Indeed, biphenyl is a substructure of 30 approved drugs spanning multiple ATC classes. Addition of the tetrazole
moiety to biphenyl seems to create a TFPS, adding selectivity considering that biphenyl-tetrazole is only identified as privileged for GPCRA targets. The scaffold is present in 8 approved drugs targeting GPCRs: Valsartan, Olmesartan, Losartan, Candesartan, Irbesatan, Tasosartan and Fimasartan.

(a) Biphenyl

<table>
<thead>
<tr>
<th>Family</th>
<th>Subfamily</th>
<th># Targets</th>
<th># pIC50</th>
<th>(-\log(p\text{-value}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPCR</td>
<td>GPCRA</td>
<td>172</td>
<td>5156</td>
<td>124</td>
</tr>
<tr>
<td>NHR</td>
<td>NHR1</td>
<td>11</td>
<td>691</td>
<td>20</td>
</tr>
<tr>
<td>VGIC</td>
<td>VGSC</td>
<td>9</td>
<td>958</td>
<td>42</td>
</tr>
</tbody>
</table>

(b) Biphenyl-tetrazole

<table>
<thead>
<tr>
<th>Family</th>
<th>Subfamily</th>
<th># Targets</th>
<th># pIC50</th>
<th>(-\log(p\text{-value}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPCR</td>
<td>GPCRA</td>
<td>16</td>
<td>1304</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 5. 2 Identified privileged scaffolds: (a) biphenyl, and (b) biphenyl-tetrazole. The p-value is negative log normalised for readability.

Sequence and structural analysis of GPCRA binding sites located between the transmembrane helices: 3, 5 and 6, conducted by Bondensgaard et al.\textsuperscript{304} reveals a conserved subpocket composed of mainly aromatic and non-polar residues rendering it complementary to biphenyl-tetrazole in addition to other moieties including 2-phenyl-
indole and spiro-piperidine-indane. Conserved regions of binding sites are often referred to as privileged pockets since they provide an explanation for the existence of privileged scaffolds. The observations made validate our identification of biphenyl-tetrazole as a TFPS. It is important to mention however that even a TFPS may not be universal to all targets within a protein family. Further analysis of structural data reveals the conserved subpocket as being present in growth hormone receptors and angiotensin II receptors but not in melanocortin receptors\textsuperscript{304}, highlighting that all cases must be analysed in a case-by-case basis.

Figure 5. 6 Chemical structures of 2-biphenyl-tetrazole and biphenyl, both identified as privileged scaffolds using a bioactivity enrichment-based approach. Biphenyl is considered a classical privileged scaffold while the tetrazole moiety generates selectivity through interaction with a conserved subpocket present in proteins belonging to the GPCRA family.
While the proposed method is able to identify and categorize known privileged scaffolds, it is important to understand potential pitfalls. The main issue is that chemical data includes sources of bias; including the fact that particular scaffolds appear frequently in commercial screening libraries. Certain fragments are preferred in these sets due to available synthetic routes or as an artifact of design methods. As a result, noise is injected into available bioactivity data. Noise can only be overcome through a large quantity of measurements. Despite pitfalls we expect that similar approaches may be useful in structure-based drug design pipelines.

5.4 Conclusion

The ScaffoldGraph software provides a flexible, extensible and user-friendly interface for the calculation of hierarchical scaffold relationships. The software has the ability to calculate multiple types of relationship structures and is significantly faster than available open-source implementations, making it the most appropriate available solution for high-throughput data analysis. The software is packaged under the open-source friendly MIT license and is open to contributions from users. The presented case studies further reinforce the utility and flexibility of the implementation showing uses in industrial and academic applications. Future additions to the package may focus on the visualisation of scaffold hierarchies. Visualisations could include: tree-maps, circle-packing or dashboard-based interfaces for scaffold exploration. Further contributions could also include new scaffold
hierarchy representations, new analysis tools, or perhaps novel scaffold tree prioritization rules. We believe that the ScaffoldGraph software will provide a beneficial resource for both academia and industry.
References


https://doi.org/10.1038/nrd3681.


https://doi.org/10.1038/nrd1468.


(15) D’Amato, R. J.; Loughnan, M. S.; Flynn, E.; Folkman, J. Thalidomide Is an


https://doi.org/10.1016/j.drudis.2018.01.018.


*Nat. Chem. Biol.* **2008**, *4*, 682.


https://doi.org/10.2174/1568026615666150112103510.

(22) Kim, T.-W. Drug Repositioning Approaches for the Discovery of New


(28) Hoehndorf, R.; Oellrich, A.; Rebholz-Schuhmann, D.; Schofield, P. N.; Gkoutos,


https://doi.org/10.1126/science.1132939.


(44) Kuhn, M.; Letunic, I.; Jensen, L. J.; Bork, P. The SIDER Database of Drugs and

https://doi.org/10.1093/nar/gkv1075.


https://doi.org/10.1093/nar/gkj120.


https://doi.org/10.1093/nar/gkw1074.


(56) Brown, A. S.; Patel, C. J. A Review of Validation Strategies for Computational

https://doi.org/10.1093/bib/bbw110.


https://doi.org/10.1021/jm401411z.


https://doi.org/https://doi.org/10.1002/qsar.200330831.


(62) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug


(69) Grant, J. A.; GallarDO, M. A.; Pickup, B. T. A Fast Method of Molecular

https://doi.org/10.1002/(SICI)1096-987X(19961115)17:14<1653::AID-JCC7>3.0.CO;2-K.


https://doi.org/https://doi.org/10.1002/jcc.21307.


https://doi.org/https://doi.org/10.1107/S0021889883010985.


Rost, B. Twilight Zone of Protein Sequence Alignments. *Protein Eng.* **1999**, *12* (2), 85–94.


(112) Buchan, D. W. A.; Shepherd, A. J.; Lee, D.; Pearl, F. M. G.; Rison, S. C. G.;
        Thornton, J. M.; Orengo, C. A. Gene3D: Structural Assignment for Whole Genes
        and Genomes Using the CATH Domain Structure Database. *Genome Res.* **2002**,  
        12 (3), 503–514.

        Lees, J. Gene3D: Extensive Prediction of Globular Domains in Proteins. *Nucleic

        Committee of the International Union of Biochemistry and Molecular Biology on

(115) Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.;
        Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T. Gene Ontology: Tool for the

(116) Schlicker, A.; Domingues, F. S.; Rahnenführer, J.; Lengauer, T. A New Measure
        for Functional Similarity of Gene Products Based on Gene Ontology. *BMC

(117) Zhao, C.; Wang, Z. GOGO: An Improved Algorithm to Measure the Semantic

(118) Sillitoe, I.; Dawson, N.; Thornton, J.; Orengo, C. The History of the CATH


(126) RCSB. Ligand Expo Downloads http://ligand-expo.rcsb.org/ld-download.html
   (accessed Jan 31, 2019).

(127) Van Rossum, G. Python Software Foundation. Python Language Reference

(128) Pandas. Python Data Analysis Library — pandas: Python Data Analysis Library

(129) Chartier, M.; Najmanovich, R. Detection of Binding Site Molecular Interaction
   https://doi.org/10.1021/acs.jcim.5b00333.

(130) Laskowski, R. A. PDBsum: Summaries and Analyses of PDB Structures. *Nucleic
   Acids Res.* **2001**, *29* (1), 221–222.

(131) Wallace, A. C.; Laskowski, R. A.; Thornton, J. M. LIGPLOT: A Program to

(132) Landrum, G. RDKit: Open-Source Cheminformatics Software. **2016**.

(133) Sanders, M. P. A.; McGuire, R.; Roumen, L.; de Esch, I. J. P.; de Vlieg, J.; Klomp,
   J. P. G.; de Graaf, C. From the Protein’s Perspective: The Benefits and Challenges
   28–38. https://doi.org/10.1039/C1MD00210D.

(134) O’Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.;


233–249.


(149) The Web framework for perfectionists with deadlines | Django


(157) Robinson, I.; Webber, J.; Eifrem, E. *Graph Databases: New Opportunities for
(158) RCSB PDB - REST Web Service http://www.rcsb.org/pdb/software/rest.do
(accessed Jan 31, 2019).

(159) Wang, G.; Dunbrack, R. L. PISCES: A Protein Sequence Culling Server.


(160) Söding, J. Protein Homology Detection by HMM–HMM Comparison.


https://doi.org/10.1093/bioinformatics/bti125.


*32* (2), 309–311.

(162) Velankar, S.; Best, C.; Beuth, B.; Boutselakis, C. H.; Cobley, N.; Sousa Da Silva, 
A. W.; Dimitropoulos, D.; Golovin, A.; Hirshberg, M.; John, M. PDBe: Protein 

(163) Kellenberger, E.; Schalon, C.; Rognan, D. How to Measure the Similarity Between 

(164) Lemieux, R. U.; Spohr, U. How Emil Fischer Was Led to the Lock and Key 


(184) Xie, L.; Xie, L.; Bourne, P. E. A Unified Statistical Model to Support Local


293

https://doi.org/https://doi.org/10.1016/j.jmb.2007.01.086.


DrawVenn. 2022.


Bianchini, M.; Belahcen, A.; Scarselli, F. A Comparative Study of Inductive and Transductive Learning with Feedforward Neural Networks *BT - AI* 2016 Advances in Artificial Intelligence; Adorni, G., Cagnoni, S., Gori, M., Maratea, M., Eds.; Springer International Publishing: Cham, 2016; pp 283–293.


https://doi.org/10.1038/s42256-021-00418-8.


https://doi.org/10.1038/s41598-021-03206-x.


(231) Bai, Y.; Ding, H.; Bian, S.; Chen, T.; Sun, Y.; Wang, W. Simgnn: A Neural
Network Approach to Fast Graph Similarity Computation. In *Proceedings of the
Twelfth ACM International Conference on Web Search and Data Mining*; 2019; pp
384–392.

(232) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient
3263(98)00002-3.

Message Passing for Quantum Chemistry. In *International conference on machine
learning*; PMLR, 2017; pp 1263–1272.

(234) Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated Graph Sequence Neural

arXiv1412.6980 2014*.

(236) Gao, M.; Skolnick, J. APoc: Large-Scale Identification of Similar Protein Pockets.

https://doi.org/10.1093/bioinformatics/btt024.


(243) Govindaraj, R. G.; Naderi, M.; Singha, M.; Lemoine, J.; Brylinski, M. Large-Scale


(264) Wang, Y.; Huang, H.; Rudin, C.; Shaposhnik, Y. *Understanding How Dimension*


https://doi.org/10.1093/nar/gkl124.


https://doi.org/http://dx.doi.org/10.2174/156802607780636690.

(270) Peón, A.; Naulaerts, S.; Ballester, P. J. Predicting the Reliability of Drug-Target


(277) Georgiadis, D.; Yiotakis, A. Specific Targeting of Metzincin Family Members with Small-Molecule Inhibitors: Progress toward a Multifarious Challenge. 


https://doi.org/10.1021/jm9602928.


https://doi.org/10.1021/ci100203e.

(283) Langdon, S. R.; Brown, N.; Blagg, J. Scaffold Diversity of Exemplified Medicinal


(290) Welsch, M. E.; Snyder, S. A.; Stockwell, B. R. Privileged Scaffolds for Library


https://doi.org/10.1021/ci2000924.


(296) Matlock, M. K.; Zaretzki, J. M.; Swamidass, S. J. Scaffold Network Generator: A


(299) Scott, O. B. ScaffoldGraph. 2020.


