Structure-Based Ligand Discovery:

Elaborating Fragment Hits *in silico*

Susan Helen Leung

St Hilda's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas Term 2019

Acknowledgements

These past four year have been a journey. As I near the end this thesis, writing these acknowledgements has made me realise and appreciate all the more, the incredible people I have met along the way.

Firstly, thank you to my supervisors: Garrett, Paul, Frank and Mike for letting me work on this project with you. Although, such an interdisciplinary project has its challenges, I have undeniably gained many skills and learnt many things from each of you.

To Greg, thank you for the opportunity of working with you on the Google Summer of Code project. You were a great mentor, and I learnt an incredible amount. It gave me invaluable exposure and opportunities (also a great break from the DPhil)! Also, thank you for making RDKit – the contributions and friendly community that has grown as a result is incredible. I hope I will have the opportunity to go to future RDKit meetings.

To all PhDs and Postdocs who I have worked with, you have made me the scientist I am today. Tony, thank you for your support during my time at Diamond, I really enjoyed my time working with you and you always took the time, despite having none, to show me around. To everyone up the hill, even though I did not spend much time there, I always felt welcomed and I enjoyed the random catch-ups, so thank you.

Thank you to OPIG members past and present. Thank you to those in my office 2.03 – Anne, Carlos, Lucian, and previously 2.17, I have had many laughs and I always found someone to talk or rant to if needed. To Jin, right from the start of my DPhil you always provided me with friendly guidance and wisdom, so thank you. Konrad, although I did not speak to you much, you were always willing to give advice and guidance for the times I did. Aleks, despite enforced fun, I enjoyed our chats and social times outside of the enforced. Catherine, thank you for the career chats and you are the most prepared person I know, more prepared than me at finding a job despite finishing next year.

Hannah, you have had a similar DPhil experience and your strong will and determination was admirable. Laura, I have many great memories of time spent with you, and thank you for introducing me to triathlon and excellent TV. A special thanks to Lucian, I could not have done Chapter 5 without you. But seriously, thank you so much for patiently explaining everything to me.

To the St Hilda's ladies, thank you for having dinner with me so many times. To the SABS ladies, Qinrui, Isabelle and Clare, you are so inspirational to me. You have no idea how much I look up to you. To my previous housemates, Alex and Joe, I already miss our pub times and rants. I hope we can meet up soon.

Thank you to Gustavo for your support. Even when I was stressed and not the most pleasant, you were still there to cook for me and listen to my worries.

Finally, I would like to thank my family for their continuous support over the past eight years at Oxford. I have moved in and out of places countless number of times and you were always ready, without question, to help me. I really could not have done this without you.

Abstract

The high attrition rates of drug discovery have been a key motivation for the development of computational methods to design better, safer, more promising molecules. Hit-to-lead development is often driven by the subjective decisions of medicinal chemists. There is growing interest in developing more objective tools to explore the vastness of chemical space. Fragment-based drug discovery offers the advantage of a high coverage of chemical space through the identification of fragment hits, which are typically more weakly binding and smaller than drug-like molecules. The process of fragment elaboration aims to increase their potency and potentially other objectives. However, currently there is no agreed best method for how to elaborate fragment hits using all structural information acquired.

This thesis describes the development of computational methods to tackle this problem. I firstly propose a workflow and describe how I applied it to a prospective study involving nudix hydrolase NUDT7, a target of interest to the Structural Genomics Consortium. The workflow involves reaction enumeration, protein-ligand docking and selection of candidates that show a conserved binding pose in their docking results. In the prospective study, I developed four hypotheses based on potential protein-ligand interactions, to further select the candidates. As a result, 105 amides were prioritised and synthesised using semi-automated synthesis. I then soaked 78 crude reactions into crystals of the target protein, which resulted in six protein-ligand crystal structures, five of which were novel. During application of this workflow, I used RMSD of the common substructures to measure the conservation of binding mode; however, I proposed that this may not be the most appropriate measure for comparisons between fragments and their elaborated counterparts. Hence, this was the motivation for the development of SuCOS.

SuCOS is an open-source combined shape and chemical feature overlap score. Through three studies, I compared the use of SuCOS to RMSD and protein-ligand interaction fingerprints (PLIFs) and explored the strengths and weaknesses of each, using a dataset of X-ray crystal structures of paired elaborated larger and smaller molecules bound to the same protein. My redocking and cross-docking studies showed that SuCOS had notable advantages over RMSD and PLIF similarity. I also showed that reranking with SuCOS performed better than the native AutoDock Vina score at differentiating actives from decoy ligands using the DUD-E dataset. As SuCOS is measured between one reference and one query molecule, I investigated the use of two group fusion methods – cumulative and max – when there are multiple reference structures, which is often the case after a fragment screening campaign. However, there was no group fusion method that consistently performed best for the four target datasets I validated on.

Finally, I investigated the use of Bayesian optimisation for ligand-based and structure-based virtual screening. Optimisation was performed over discrete chemical space, so essentially prioritises which molecules to make next from an input set. I investigated the influence of different molecular representations and different kernels on the performance of Bayesian optimisation. For the two ligand-based experiments, Morgan fingerprints with the Tanimoto kernel showed the best performance. For the structure-based Bayesian optimisation experiments, I investigated two structure-based representations: vectorised RDKit pharmacophoric feature maps and PLIFs. However, the results showed that there was no clear advantage to using either structure-based representation over 2D fingerprints such as Morgan fingerprints.

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where stated otherwise by reference or acknowledgment, the work presented is entirely my own.

Susan Helen Leung
Monday, 11 May 2020

Contents

Chapter 1 Introduction 1
1.1 The Drug Discovery Process 1
1.2 Virtual Screening 3
1.2.1 Drug-Likeness Filters 5
1.2.1.1 Lipinski's Rule of Five 6
1.2.1.2 Rapid Elimination Of Swill (REOS) 6
1.2.1.3 Pan-Assay Interference Compounds (PAINS) 7
1.2.1.4 Molecular Obesity and Ligand Efficiency 8
1.2.2 Molecular Diversity 9
1.2.3 Ligand-Based Virtual Screening 12
1.2.3.1 Molecular Similarity 13
1.2.3.2 Quantitative Structure-Activity Relationships 18
1.2.3.3 Supervised Machine Learning 19
1.2.3.4 Shape Similarity 21
1.2.3.5 Ligand-Based Pharmacophores 23
1.2.4 Structure-Based Virtual Screening 25
1.2.4.1 X-Ray Crystallography 25
1.2.4.2 Protein-Ligand Docking 29
1.2.4.3 Protein-Ligand Interaction Fingerprints 34
1.2.4.4 Postprocessing Docking Strategies 36
1.3 *De Novo* Molecular Design 39
1.3.1 Multi-Objective Optimisation 40
1.3.2 Compound Generation 42
1.4 Fragment-Based Drug Discovery 45
1.5 Project Aims 49
Chapter 2 Prospective Study – Designing Follow-Up Compounds for NUDT7 53
2.1 Introduction 53
2.1.1 NUDT7 54
2.1.2 Prior State of the Art and Chapter Aims 56
2.2 Methods 58
2.2.1 Workflow for *in silico* Elaboration of Fragment Hits and Docking 58
2.2.2 Reactions and Reagents 59
2.2.3 Protein Preparation for Docking 60
2.2.4 Ligand Preparation for Docking 61
2.2.5 Docking 62

2.2.6 Calculation of RMSD 62
2.2.7 Calculation of Protein-Ligand Interaction Fingerprints (PLIFs) 63
2.2.8 Experimental procedure for amide coupling using the Opentrons liquid-handling robot 65
2.2.9 Soaking crude reaction mixtures into NUDT7 crystals 66
2.2.10 Data Collection and Structure Solution 67
2.3 Results and Discussion 67
2.3.1 Choosing Follow-Up Compounds to x1237 67
2.3.2 Synthesis of the Follow-Up Compounds 74
2.4 Conclusions 80
Chapter 3 Comparison of Similarity of Binding Mode Measures for Elaborated Fragments 83
3.1 Introduction 84
3.1.1 Prior State of the Art and Chapter Aims 86
3.2 Methods 87
3.2.1 Downloading and Filtering of the Malhohtra and Karanicolas Ligand Pair Set 89
3.2.2 Preparation of the Malhotra and Karanicolas Ligand Pairs 89
3.2.3 Protein Preparation for Docking 90
3.2.4 Ligand Preparation 91
3.2.5 Docking 92
3.2.6 Calculation of RMSD 93
3.2.7 Calculation of Protein Ligand Interaction Fingerprints (PLIFs) 93
3.2.8 Calculation of SuCOS 94
3.3 Results and Discussion 97
3.3.1 Part I: Comparison of MCS-RMSD, TvPLIF and SuCOS Between the Ligands in the Aligned Crystal Structures of the Malhotra and Karanicolas Ligand Pair Set 97
3.3.2 Part II: Using All-RMSD, TnPLIF and SuCOS to Rescore the Redockings of the Malhotra and Karanicolas Ligand Pair Set 108
3.3.3 Part III: Comparison of All-RMSD/MCS-RMSD, TvPLIF and SuCOS on the Cross-Docked Larger Ligand into the Smaller Ligand's Protein Structure of the Malhotra and Karanicolas Ligand Pair Set 116
3.3.4 Computational Efficiency of MCS-RMSD, PLIF Similarity, and SuCOS 125
3.4 Conclusion 126
Chapter 4 Investigating the Ability of SuCOS to Classify Actives & Decoys in DUD-E 129
4.1 Introduction 130
4.1.1 DUD-E dataset 130

4.1.2 Measuring the Performance of Classification Models 131
4.1.3 Data Fusion 133
4.1.1 Prior State of the Art and Chapter Aims 135
4.2 Methods 137
4.2.1 Using SuCOS for Virtual Screening 137
4.2.2 Preparation of Datasets 138
4.2.3 Tanimoto SuCOS 139
4.2.4 Clustering of Fragments 140
4.2.5 Comparison of Data Fusion Methods 141
4.3 Results and Discussion 142
4.3.1 Comparison of the Native AutoDock Vina Score with Rescoring Using SuCOS for Virtual Screening Using the DUD-E Dataset 142
4.3.2 Investigating Optimal Weightings for SuCOS Across the DUD-E Dataset 144
4.3.3 Investigating Group fusion methods for Combining Multiple Fragment-Protein Structures to Use as References in Virtual Screening 146
4.3.3.1 Beta-Secretase 1, BACE1 150
4.3.3.2 Cyclin-Dependent Kinase, CDK2 156
4.3.3.3 Carbonic Anhydrase 2, CAH2 162
4.3.3.4 Trypsin 1, TRY1 168
4.1.1.1 Summary 173
4.4 Conclusions 175
Chapter 5 Using Bayesian Optimisation for Ligand-Based & Structure-Based VS 179
5.1 Introduction 179
5.1.1 Introduction to Bayesian Optimisation 180
5.1.2 Probabilistic Surrogate Models 183
5.1.3 Acquisition Function 184
5.1.4 Greedy Search and Random Search 185
5.1.5 Choice of Kernel 185
5.1.6 Molecular Representations 187
5.1.7 Multi-Armed Bandit Problem 188
5.1.8 Prior State of the Art and Chapter Aims 189
5.2 Methods 191
5.2.1 MMP-12 Dataset Preparation 191
5.2.2 Malaria Dataset Preparation 192
5.2.3 Running the Bayesian Optimisation 193
5.2.4 Evaluation of Method Performance 195
5.2.5 Building Machine Learning Models 195

5.2.6 Preparation of Datasets from PDBbind....197
5.2.7 Vectorised Pharmacophoric Features....198
5.2.8 ElectroShape....200
5.3 Results and Discussion....201
5.3.1 Bayesian Optimisation for Ligand-Based Screening: Using the MMP-12 dataset....201
5.3.1.1 Gaussian Process Regression....208
5.3.1.2 What to make next in the MMP-12 series?....208
5.3.2 Bayesian Optimisation for Ligand-Based screening: Using the Malaria Dataset....210
5.3.3 Structure-Based Bayesian Optimisation using RDKit Pharmacophoric Feature Maps....214
5.3.4 Structure-Based Bayesian Optimisation: Using the CAH2 Dataset to Explore PLIFs and RDKit pharmacophoric Feature Maps....222
5.4 Conclusions....225
Chapter 6 Conclusions and Future Directions....230
6.1 Summary and Future Work....231
6.2 Concluding Remarks....235
Bibliography....237
Appendix A : Chapter 2....254
A.1 Code for synthesis protocol for Opentrons....256
A.2 Tables listing the reagent SMILES, hypotheses and well locations for the 105 amide-forming reactions....261
A.3 LCMS traces for the six amide follow-up hits....268
A.4 LCMS traces for the three starting material hits....274
Appendix B : Chapter 4....277
B.1 Common crystallographic additives that were excluded....277
Appendix C : Chapter 5....301

List of Figures

Figure 1.1 The stages of drug discovery. 1
Figure 1.2. The typical process of virtual screening. 4
Figure 1.3. Example of some of the functional groups filtered out by REOS. 7
Figure 1.4. Example structures to demonstrate the similarity principle 13
Figure 1.5. The design, make, test experimental cycle and the *de novo* design cycle 39
Figure 1.6. Illustration of a Pareto front for a multi-objective optimisation problem 41
Figure 1.7. The eleven disconnection rules of RECAP and the substructures that define a poised fragment. 43
Figure 1.8. Optimisation of fragment hits versus optimisation of HTS hits 46
Figure 1.9. Illustrations of the different fragment optimisation and growing strategies. 48
Figure 2.1. Initial fragment hits from screening NUDT7 with DSPL fragment library. 55
Figure 2.2. Second generation hits for NUDT7 and my follow-up strategy for one hit. 56
Figure 2.3 Flow chart showing the overview of the workflow to propose fragment-hit follow-ups. 58
Figure 2.4. Reaction scheme for coupling and acylation reactions. 65
Figure 2.5. *In silico* filtering of the MolPort and PEB reagent databases 68
Figure 2.6. The four hypotheses that prioritised which follow-up compounds to make 72
Figure 2.7. Crude reaction soaking resulted in six X-ray protein-ligand structures 76
Figure 2.8. Starting materials were seen in three X-ray crystal structures 79
Figure 3.1. Overview of study performed in Chapter 3. 88
Figure 3.2. The distribution of the resolutions of the PDB structures in Chapter 3 98
Figure 3.3 SuCOS is a good open-source alternative to the COS metric. 99
Figure 3.4. (a) Comparing the differences in SuCOS values when using *TM-align* 101
Figure 3.5. Comparison of three conservation of binding mode metrics on the Malhotra and Karanicolas ligand pair set. 102
Figure 3.6. Example of a false negative in Figure 3.5b. 105
Figure 3.7. Example of a false positive in Figure 3.5b 106
Figure 3.8. TvPLIF for larger and smaller ligand pairs using the same protein conformation 107
Figure 3.9. Choosing by affinity consistently performs the worst at picking the docking pose which most closely resembles the crystal ligand pose 110
Figure 3.10. Redocking ligand of 1o39: example of a disadvantage of using TnPLIF 112
Figure 3.11. Three examples of FPs and FNs for each metric 114
Figure 3.12. Distributions of each metric for the cross-docking poses to L_S^X and L_L^A 117
Figure 3.13. The distribution of Pearson correlations for each metric in the cross-docking study 120

Figure 3.14. The distribution of Spearman's ρ and Kendall's τ for each metric in the cross-docking study 121
Figure 3.15. Example of a case where using SuCOS produced a negative R_P. 122
Figure 3.16. Example of a case where using RMSD produced a negative R_P 123
Figure 3.17. Investigating the effect of altering the weights of feature overlap and shape overlap in SuCOS 124
Figure 3.18. Execution times (wall clock) for calculation of MCS-RMSD, TvPLIF (Arpeggio and post-processing Arpeggio) and SuCOS 126
Figure 4.1. Confusion matrix for treating SuCOS as a binary classifier. 131
Figure 4.2. Workflow for the preparation of datasets for investigating data fusion with SuCOS and the DUD-E dataset 138
Figure 4.3. Clustering $frags_{all}$ to give $frags_{clustered}$. 141
Figure 4.4. ROC AUCs for rescoring with SuCOS versus the native AutoDock Vina scoring function for the 102 targets of the DUD-E dataset 143
Figure 4.5. Investigating the optimal weights for SuCOS for all the DUD-E targets .. 145
Figure 4.6. A group fusion rule is required to combine multiple SuCOS similarities .. 147
Figure 4.7. The distribution of the resolutions of the protein-fragment PDB structures for the four targets investigated in Chapter 4 148
Figure 4.8. The reference DUD-E ligand and the $frags_{all}$ and $frags_{clustered}$ for BACE1 151
Figure 4.9. The distribution of ROC AUCs achieved for each fragment for BACE1 .. 152
Figure 4.10. The DUD-E reference ligand and reference fragment with the highest ROC AUC for BACE1 152
Figure 4.11. The distribution of ROC AUCs for each BACE1 fragment, grouped by cluster 153
Figure 4.12. Illustration of the BACE1 fragment clusters 155
Figure 4.13. The DUD-E reference ligand, $frags_{all}$ and $frags_{clustered}$ for CDK2 157
Figure 4.14. The distribution of ROC AUCs achieved for each fragment for CDK2 .. 157
Figure 4.15. The best fragment and the DUD-E reference ligand for CDK2 show similar pharmacophoric features 158
Figure 4.16. The three best and three worst CDK2 reference fragments by ROC AUC 158
Figure 4.17. The distribution of ROC AUCs for each CDK2 fragment, grouped by cluster 159
Figure 4.18. Illustration of the CDK2 fragment clusters 161
Figure 4.19. The DUD-E reference ligand, the best and worst reference fragments for CAH2 163
Figure 4.20. The distribution of ROC AUCs achieved for each fragment for CAH2 .. 164
Figure 4.21. The distribution of ROC AUCs for each CAH2 fragment, grouped by cluster 166
Figure 4.22. Illustration of the CAH2 fragment clusters 167

Figure 4.23. The DUD-E reference ligand, the best and worst reference fragments for TRY1 169
Figure 4.24. The distribution of ROC AUCs achieved for each fragment for TRY1 169
Figure 4.25. The TRY1 clusters resulting from two different cutoffs, t=0.8 and t=0.6 171
Figure 4.26. The distribution of ROC AUCs for each TRY1 fragment, grouped by cluster 172
Figure 5.1. An example of a Bayesian optimisation run over seven iterations. 181
Figure 5.2. The common biaryl sulfonamide scaffold in the MMP-12 dataset 192
Figure 5.3. Schematic showing how the pharmacophoric features are vectorised for each ligand 199
Figure 5.4. Histogram showing the distribution of the pIC_{50}s of the 1,880 compounds of the MMP-12 dataset. 202
Figure 5.5. The evolution of the best pIC50 found against the iteration number 203
Figure 5.6. Histogram showing similar distributions of molecular similarity while varying the number of bits in the Morgan fingerprints for the MMP-12 dataset 204
Figure 5.7. The distribution of similarities when calculated using MACCS fingerprints 205
Figure 5.8. Comparison between the different methods for the recovery rate of desirable molecules 206
Figure 5.9. Distribution of all the points sampled during each optimisation for each method for the MMP-12 dataset. 207
Figure 5.10. The molecules that Bayesian optimisation and the trained GPR model suggests to make next 209
Figure 5.11. Histogram showing the distribution of the pEC_{50}s of the 18,924 compounds of the malaria dataset. 211
Figure 5.12. Evolution of the maximum pEC_{50} for the different Bayesian optimisation methods for the malaria dataset 211
Figure 5.13. Recovery rate of the top 10^{th} percentile ($pEC_{50} \geq 6.75$) of the malaria dataset for the various Bayesian optimisation methods, alongside random sampling. 212
Figure 5.14. Distribution of all pEC_{50}s sampled during the Bayesian optimisations with the malaria dataset 213
Figure 5.15. Targets used to investigate vectorised pharmacophoric features as the Bayesian optimisation search space 215
Figure 5.16. Histograms showing the distribution of activity data for the four targets 216
Figure 5.17. The distribution of the resolutions for the PDBs of the four targets 216
Figure 5.18. Evolution of the maximum pIC_{50} or $pK_{i/d}$ found with iteration number 218
Figure 5.19. Recovery rate of top decile most potent compounds for each target 219
Figure 5.20. Mean distributions of all points sampled during each optimisation 221
Figure 5.21. The distribution of the $pK_{i/d}$ values of the CAH2 database used to investigate PLIFs as the Bayesian optimisation search space 223

Figure 5.22. . The distribution of the resolutions for the PDBs in the CAH2 dataset set used in Section 5.3.4 for the investigation of PLIFs as the Bayesian optimisation search space.....223

Figure 5.23. Evolution of the maximum pIC_{50} or $pK_{i/d}$ found with iteration number. .223

Figure 5.24. Recovery of the top decile most potent compounds ($pK_{i/d} \geq 9$) in the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space 224

Figure 5.25. Distribution of all the $pK_{i/d}$ values found during each optimisation, in the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space 225

Figure A.1. The distribution of molecular weights of the prospective amide candidates that were docked into NUDT7.....254

Figure A.2 The predicted docked pose for x0090 had a RMSD > 2 Å with respect to the crystal pose.....254

Figure A.3. Protein Ligand Interaction Fingerprint (PLIF) heatmap for the 1st and 2nd generation hits for NUDT7 255

Figure B.1. Investigating how the ROC AUC changes with varying the weight of the components of SuCOS for each DUD-E target. 288

Figure B.2. Heat map showing the distance matrix between all of the 34 BACE1 fragments and the dendrogram obtained from hierarchical clustering 300

Figure C.1. Evolution of the maximum pEC_{50} for the different Bayesian optimisation methods for the malaria dataset with standard deviation errors shown 301

Figure C.2. Evolution of the maximum pIC_{50} or $pK_{i/d}$ found with iteration number for the four targets used to investigate vectorised pharmacophoric features, with standard deviation errors shown.....303

Figure C.3. Bayesian optimisation was run with the various methods on the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space with standard deviation errors shown 304

List of Tables

Table 2-1. Filters and reaction SMARTS for generation of amide follow-ups. 70
Table 2-2. Summary of the follow-up compounds of x1237 chosen for synthesis. 74
Table 2-3. Summary of the results from the crude reaction screen 77
Table 2-4. Details of the structures containing starting materials. 79
Table 3-1. Notation used to describe the structures and metrics used in Chapter 3. 88
Table 3-2. Examples of cases where MCS-RMSD is inappropriate..... 103
Table 3-3. The numbers of TPs, TNs, FPs and FNs for each metric and criterion..... 113
Table 3-4. Summary of docking the larger ligand into the smaller ligand's protein.... 118
Table 4-1. Notation of the different data fusion methods investigated in Chapter 4.... 142
Table 4-2. The mean AUC ROC and RE for the AutoDock Vina score versus SuCOS. 143
Table 4-3. The targets used to investigate different group fusion methods..... 146
Table 4-4. ROC AUCs achieved by group fusion methods for BACE1 152
Table 4-5. ROC AUCs achieved by group fusion methods for CDK2..... 157
Table 4-6. ROC AUCs achieved by group fusion methods for CAH2..... 164
Table 4-7. ROC AUCs achieved by group fusion methods for TRY1 169
Table 4-8. Summary of the group fusion results for the four targets..... 175
Table 1-1. R^2 and MSE values for the GPR models trained on the MMP-12 dataset .. 208
Table 1-2. Targets used to investigate vectorised pharmacophoric features. 214
Table A-1. Summary of the 25 acyl chlorides used in the acylation reactions..... 262
Table A-2. Summary of the 80 carboxylic acids used in the coupling reactions. 267
Table B-1. Number of molecules parsed by RDKit for each DUD-E target. 279
Table B-2. DUD-E AUC ROC for AutoDock Vina and SuCOS. 280
Table B-3. DUD-E ROC enrichment at 0.5% for AutoDock Vina and SuCOS..... 281
Table B-4. DUD-E ROC enrichment at 1% for AutoDock Vina and SuCOS..... 282
Table B-5. DUD-E ROC enrichment at 2% for AutoDock Vina and SuCOS..... 283
Table B-6. DUD-E ROC enrichment at 5% for AutoDock Vina and SuCOS..... 284
Table B-7. PDB IDs, ligand IDs and AUC results for each fragment for BACE1..... 289
Table B-8. PDB IDs, ligand IDs and AUC results for each fragment for CDK2..... 292
Table B-9. PDB IDs, ligand IDs and AUC results for each fragment for CAH2 297
Table B-10. PDB IDs, ligand IDs and AUC results for each fragment for TRY1 299

Abbreviations

ADME	Absorption, distribution, metabolism and excretion
All-RMSD	RMSD using all atoms in both reference and query molecules
AUC	Area under the curve
AUROC	Area under the ROC curve
BACE1	Beta-Secretase 1
CAH2	Carbonic Anhydrase 2
CDK2	Cyclin-Dependent Kinase
cLogP	Computed octanol-water partition coefficient
COS	Combined overlap score
DLS	Diamond Light Source
ECFP	Extended-connectivity fingerprint
EI	Expected improvement
EF	Enrichment factor
EM	Electron microscopy
FBDD	Fragment-based drug discovery
FN	False negative
FNR	False negative rate
FP	False negative
FPR	False positive rate
GP	Gaussian process
GPR	Gaussian process regression
H-bond	Hydrogen bond
HA	Heavy atoms
HTS	High throughput screening
LCMS	Liquid chromatography mass spectrometry
LE	Ligand efficiency
LLE	Lipophilic ligand efficiency
MCS	Maximum common substructure
MCS-RMSD	RMSD using the maximum common substructure
MD	Molecular dynamics
MFP	Morgan fingerprint
ML	Machine learning
MMP	Matched molecular pair
MMP-12	Matrix metalloproteinase 12
MW	Molecular weight
MMS	Matched molecular series
MSE	Mean square error
nbits	Number of bits
PDB	Protein Data Bank
PDBQT	Protein Data Bank-like file format containing atomic partial charges and AutoDock atom types used by AutoGrid and AutoDock
PQR	Protein Data Bank-like file format containing atomic partial charges and van der Waals radii
PAINS	Pan-assay interference compounds
PLIFs	Protein-ligand interaction fingerprints
QSAR	Quantitative structure-activity relationships

RBF	Radial basis function
REOS	Rapid elimination of swill
RF	Random forest
RMSD	Root mean square deviation
ROC	Receiver operating characteristic
R_P	Pearson correlation coefficient
RT	Retention time
SDF	Structure data file
SIFt	Structural interaction fingerprint
SMILES	Simplified molecular-input line-entry system
SPR	Surface plasmon resonance
TN	True negative
TNR	True negative rate
TnPLIF	Tanimoto PLIF
TP	True positive
TPR	True positive rate
TRY1	Trypsin 1
TvPLIF	Tversky PLIF
vdW	Van der Waals
VS	Virtual screening

Chapter 1 Introduction

1.1 The Drug Discovery Process

Drug-like chemical space is estimated to consist of on the order of 10^{60}-10^{100} small molecules (Drew et al., 2012). Complete synthesis and experimental screening of all chemical space for bioactive molecules is impossible, so finding an efficient method of exploring this space has been of great interest to researchers and pharmaceutical companies.

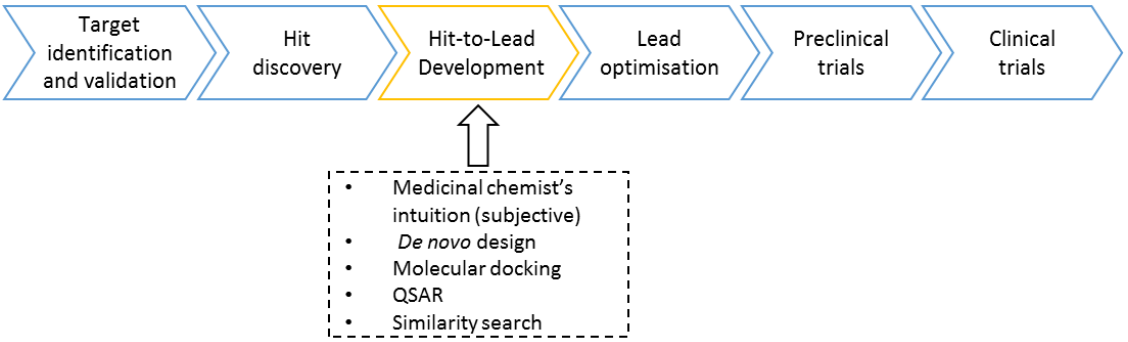


Figure 1.1 The stages of drug discovery from early target identification and validation to clinical trials. Hit-to-lead development is highlighted as it is the stage that this thesis focuses on. This stage is traditionally driven by the subjective decision making of medicinal chemists based on their personal prior experience and what is easiest to synthesise. However, several computational approaches exist, which can be used as an alternative, some of these are listed.

It was estimated that cost for the development of a new approved drug has risen from US$ 800 million in 2001, to US $2.6 billion in 2014, to US $3 billion now (Mak and Pichika, 2019; Mullard, 2014; DiMasi et al., 2015). Furthermore, the estimated time taken for end-to-end development of a drug is over twelve years (Mohs and Greig,

2017). The escalating costs and time invested is alarming and the financial pressures are undeniable incentives for the pharmaceutical industry to optimise and increase the efficiency of each stage of the drug discovery pipeline, in order to reduce the overall attrition rates. The two major sources of attrition have been reported to be lack of efficacy and clinical safety or toxicology, which are both problems that appear in the late stages of drug discovery (Roberts et al., 2014; Hopkins, 2008).

The process of drug discovery traditionally starts with target identification where the activation or inhibition of a biological entity, such as a protein, or pathway is identified as causing a therapeutic effect on a disease (Figure 1.1) (Hughes et al., 2011). This target-based approach will be the focus of this thesis; however, other approaches such as phenotypic drug discovery have become increasingly popular in recent years, and is where a target does not need to be identified. Instead, the drug candidates are assessed by their therapeutic effect on the disease model, for example in cells, tissues or at the whole-organism level. (Moffat et al., 2017).

In target-based drug discovery, after the target has been identified, further validation on the assay and the target may be required before the hit identification and lead discovery stage which requires an intensive search for the small molecule or biological therapeutic. In the case of small molecule drug discovery, once a target has been selected and validated, 'hit' molecules are then identified – these are molecules that show some desired activity in the compound screening assay. Not all the hits may progressed to the next stage, so triaging may occur based on factors such as structural diversity, potency, novelty and patentability/intellectual property protection. The hit-to-lead phase then follows, where each hit is chemically refined to produce the desired qualities – typically high potency, high selectivity for the chosen target, good physicochemical properties and good efficacy *in vivo*. Preclinical studies including *in*

vitro and *in vivo* tests must also show that candidates are safe to use before testing in humans and possess good pharmacodynamic and pharmacokinetic properties. Toxicity of the candidate must likewise be considered, in order to determine the safe ranges of dosage. All these properties are often difficult to optimise simultaneously. Moreover, if large quantities of a drug are to be made, the synthesis of a successful drug candidate needs to be scalable, and a suitable formulation of the drug needs to be identified.

Once these properties are met, the candidate lead(s) can progress into clinical trials, where they are tested in humans. There are three phases of clinical trials and if a candidate passes all phases and are approved by the appropriate regulatory agency, they can be finally marketed. The whole process of drug discovery is extremely costly in terms of time and money. There is therefore increasing interest, and expectation, in the use of computational techniques to decrease the failure rate.

The focus of this thesis is on the hit-to-lead optimisation step. Traditionally it is synthetic chemistry that drives the way medicinal chemists search for small-molecule drug candidates in the hit-to-lead stage (Figure 1.1). The kinds of molecules that are made are often biased by the experience of medicinal chemists, and what is quickest and most reliable to make but there is evidence that this approach is unsystematic and narrow in scope (Nadin et al., 2012; Lipkus et al., 2008; Dow et al., 2012; Kutchukian et al., 2012). A more systematic, unbiased way to explore chemical space is needed.

1.2 Virtual Screening

Virtual screening, or VS, is the use of computational techniques to screen for biologically active molecules within a virtual library of small molecules. VS methods score, rank and/or filter the input library so that the number of compounds to synthesise

or purchase is reduced, so as to reduce costs and increase the likelihood of finding active molecules. VS is typically quick, relatively inexpensive and can be used to search through more compounds in comparison to experimental screens. The initial hit from a VS can then be optimised in the hit-to-lead optimisation process; however, typically many of the leads will not be easily optimisable in all required properties such as potency, pharmacokinetics, selectivity *etc.* Hence, there is a need to ensure that the VS library is diverse (Section 1.2.2) so that there is a greater chance of success in the later stages.

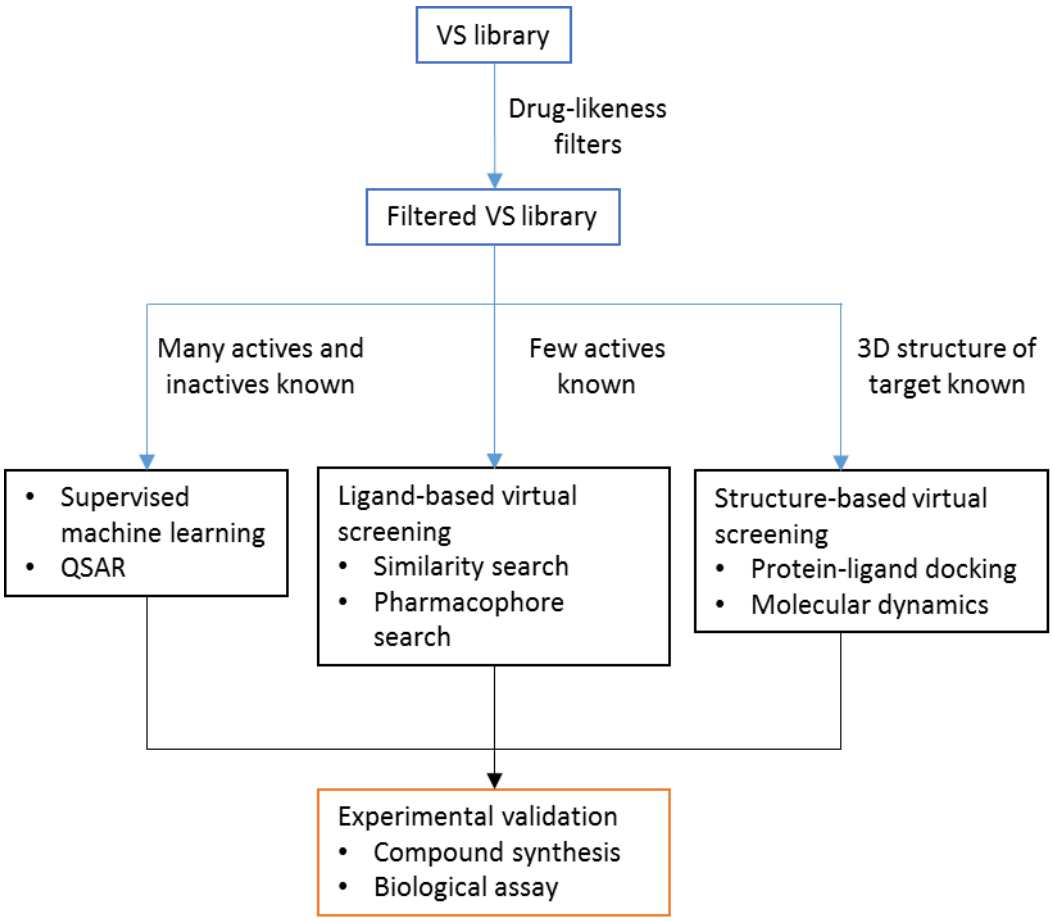


Figure 1.2. The typical process of virtual screening. Firstly, drug-likeness filters may be applied, then depending on what data is available different strategies can be employed to prioritise compounds for experimental validation. Figure is adapted from (Leach and Gillet, 2007b).

The typical virtual screening process is shown in Figure 1.2. Firstly, a VS library needs to be chosen; this library may be from a commercial database, *e.g.* Enamine REAL, or

it may be an in-house set or a virtual library produced from an *in silico* technique. The library can be filtered so that it contains only "drug-like" molecules, then based on the amount and type of data available, a VS technique is applied. The virtual screening methods can be classified into two broad types: ligand-based (Section 1.2.3) and structure-based (Section 1.2.4) although hybrid methods have been developed too (Ripphausen et al., 2010; Drwal and Griffith, 2013). Application of the VS method will rank the candidate compounds. Eventually this will lead to experimental validation *e.g.* compounds are synthesised/purchased and biologically tested.

1.2.1 Drug-Likeness Filters

The concept of "drug-likeness" attempts to identify features of a candidate molecule that are consistent with the majority approved drugs (Arrico et al., 2002; Walters and Murcko, 2002; Leach and Gillet, 2007b). Failure at the later stages of the drug discovery pipeline are more costly than if the failure occurred earlier. Between 2000 and 2010, major reasons for attrition in clinical trials were due to toxicity and poor solubility (Young and Leeson, 2018). Hence, careful filtering at the early stages of the drug discovery pipeline would save a lot of time and money and reduce attrition rates.

Numerous computational techniques aim to quantify the drug-likeness in a molecule in order to increase the success rate of lead compounds in clinical trials (Tian et al., 2015; Daina et al., 2017). The simplest techniques are filters that perform substructural searches to remove molecules containing known toxic and/or reactive groups such as acyl halides.

1.2.1.1 Lipinski's Rule of Five

One of the most prominent drug-likeness rules was conceived when Lipinski and coworkers at Pfizer performed an analysis of a database of drugs and phase II clinical candidates (Lipinski et al., 1997). Their motivation emerged from the poor bioavailability of the optimised high throughput screening (HTS) hits at Pfizer. They found that the majority of drugs and phase II candidates in the database, followed the rules: (1) hydrogen-bond donors ≤ 5; (2) hydrogen-bond acceptors ≤ 10; (3) molecular weight ≤ 500; (4) log$P \leq 5$, where P is the partition coefficient between *n*-octanol and water. As a result of this, the *Lipinski's rule of five* was born; if a candidate violates more than one rule then the candidate will be more likely to have poor absorption or poor bioavailability.

However, there are many examples of approved oral drugs that are exceptions to Lipinski's rule of five, such as atorvastatin (Lipitor) and daclatasvir (Daklinza), which both violate more than one of the rules. Hence, by applying the rule of five as filters, many opportunities may be lost for designing drugs for targets that are less druggable. The term "Beyond the Rule of Five" represents this lost chemical space, and many campaigns now exist to include compounds such as peptides, macrocycles, natural products and their mimetics, which are examples of bRo5 compounds. For recent reviews and perspectives that discuss the importance of looking beyond the rule of five, I refer the reader to (Egbert et al., 2019; Doak et al., 2014; DeGoey et al., 2018).

1.2.1.2 Rapid Elimination Of Swill (REOS)

Numerous chemical functional groups are known to be reactive, unstable or toxic under physiological conditions (Rishton, 1997). In order to eliminate compounds containing

such groups, researchers at Vertex developed the Rapid Elimination Of Swill, or REOS, (Walters et al., 1998). REOS consists of property filters and more than 200 substructural filters, which are defined by SMARTS (Walters and Murcko, 2002). Examples of reactive functional groups that are contained in the list are sulfonyl halides, aldehydes, nitro groups, primary alkyl halides, epoxides, and aziridines (Figure 1.3).

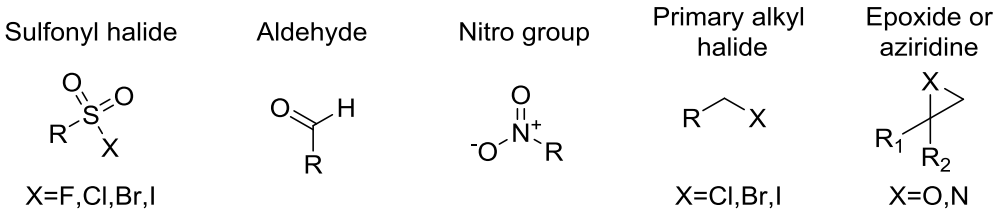


Figure 1.3. Example of some of the functional groups filtered out by REOS.

1.2.1.3 Pan-Assay Interference Compounds (PAINS)

Pan-Assay INterference compoundS, or PAINS, are compounds that appear as false positives in biological assays (Baell and Walters, 2014). These compounds typically display biological activity across multiple protein targets and unlike true positives, they do not form specific interactions by fitting into the binding site of the protein target. There are several ways that a compound can interfere to give the false positive readout, for example, if the compound is fluorescent or highly coloured then if a fluorimetric assay is conducted, the result may be misinterpreted as a positive signal. Another reason is that the compound may aggregate into colloids at certain concentrations, which may cause the protein to be inhibited or activated; hence, giving a false positive readout (Aldrich et al., 2017).

To prevent the occurrence of PAINS and aggregators, several structures have been identified as appearing frequently as false positives *e.g.* rhodanines, a five-membered

heterocyclic compound that contains a thiozolidine core, undergo reactions that irreversibly modifies a protein under light (Baell and Walters, 2014). Computational tools exist to remove such compounds based on similarity and substructure. However, this will not identify all possible PAINS, so further experimental tests should always be performed such as carrying out orthogonal biological assays; if there is a positive readout from all assays then the candidate is less likely to be interfering in the biological assay, *i.e.* is not a PAINS compound.

1.2.1.4 Molecular Obesity and Ligand Efficiency

In the hit-to-lead optimisation process, molecules are typically elaborated to increase affinity. If the only objective is to search for the candidate with the highest affinity, then heavier candidates with large molecular weights will tend to be favoured, as affinity has been found to strongly correlate with molecular weight. Moreover, increasing molecular weight often leads to an increase in lipophilicity, if nonpolar groups such as aromatic groups are added. This results in a concomitant decrease in solubility, which can lead to poor absorption, distribution, metabolism and excretion (ADME) properties. Hann reported the term, *molecular obesity* to describe when molecules are grown too large and too lipophilic during the optimisation for potency process (Hann, 2011). To counteract this effect, measures such as *ligand efficiency* and *lipophilic ligand efficiency* have been introduced.

Ligand efficiency, LE, is defined as the binding affinity divided by the number of heavy atoms (HA) in a candidate molecule (Hopkins et al., 2004). The binding energy can be derived from either experimental measurement or calculated computationally *e.g.* from molecular docking (Section 1.2.4.2). The LE can also be interpreted as the contribution of each atom to the candidate's overall binding energy. Good values for LE are

considered to be greater than 0.3 kcal $mol^{-1}HA^{-1}$ (Keserű et al., 2016; DesJarlais, 2011). However, one disadvantage of LE is that it decreases non-linearly with heavy atom count (Leeson, 2015; Reynolds et al., 2008); hence it cannot be applied equally across all molecular weights without normalisation (Nissink, 2009).

Lipophilic ligand efficiency, LLE and also known as LipE, considers potency with lipophilicity, and is defined as $pIC_{50} - \log P$ (Leeson and Springthorpe, 2007). Generally, lower values of LLE correspond to an increase in the compound's concentration in highly lipophilic cellular membranes. LLE values greater than five indicate that the candidate is less likely to be toxic (Hann, 2011). The LLE measure seeks to maximum potency whilst minimizing lipophilicity. Higher values of potency are desirable as a lower dose is required; hence there is reduced risk of toxicological events. Lower values of lipophilicity are desired as it lowers the risk of the compound being promiscuous or having non-specific target interactions and lowers the concentration in lipid membranes which otherwise can lead to unwanted toxicity.

In conclusion, there have been many efforts in devising metrics, rules and substructure filters to seek higher quality compounds early on in the discovery pipeline, that are more drug-like and less likely to contribute to decreased attrition rate. However, one must bear in mind that these rules are an over-simplification, and there are always exceptions.

1.2.2 Molecular Diversity

Another important concept in library design and virtual screening is molecular diversity. One of the main goals of virtual screening is to find a diverse set of hit molecules that show biological activity but have different chemotypes, so that in the hit-

to-lead optimisation stage, it is more likely that one of the follow-up compounds will have a set of properties that are optimisable. Moreover, a high diversity may increase the chance that some of the hits have novel chemical scaffolds with respect to the target of interest and hence have a greater prospective for intellectual property protection than others. In order to generate such a set of diverse hits, the VS library must also be diverse.

There are numerous methods for ensuring diversity in a VS library. Like similarity searching methods (Section 1.2.3), firstly a molecular descriptor needs to be chosen *e.g.* physicochemical properties, 2D molecular fingerprint, shape descriptors, *etc.* (see Section 1.2.3 for a description of each of these). Once the molecular descriptor is chosen, then the selection of a diverse subset by clustering, dissimilarity-based selection, partitioning/cell-based approaches, or optimisation-based methods, can be applied (Leach and Gillet, 2007a). In this section, I will only talk about clustering techniques; however, for an overview of the other three, I refer the reader to Gillet, 2017.

Clustering is an unsupervised machine learning technique that creates groups based on similarities of a molecular descriptor. Compounds within the same cluster will have similar molecular descriptors and compounds in different clusters will have dissimilar molecular descriptors. The general procedure is: (1) generate molecular descriptors for each compound; (2) calculate the similarity between all compounds to generate a similarity matrix; (3) use a clustering algorithm to generate clusters; and (4) choose a subset of molecules to represent each cluster. The resulting subset will thus be smaller than the original dataset but with guaranteed chemical diversity.

An example of a non-hierarchical clustering algorithm is k-means clustering, in which the user specifies the number of clusters, k, to create. A random selection of k seed molecules are chosen and the rest of the molecules are assigned to the seed that is closest, which form the clusters. The centroid of each cluster is then calculated and molecules are reassigned to the closest centroid. This process of centroid calculation and molecule relocation is repeated until the centroid molecules do not change.

Hierarchical clustering involves building up clusters into a multilevel hierarchy, where clusters at one level are joined together to form clusters at another level and thus creating a dendrogram. There are two main approaches: agglomerative and divisive. Agglomerative clustering is a bottom-up approach where each observation starts off in its own cluster and after each iteration, the pairs of clusters that are nearest to each other are merged. This process is repeated until all observations have merged into one cluster. Divisive clustering is a top-down approach, where all observations start off in the same cluster and the cluster is divided into the two least similar clusters. This process is repeated until all observations are in their own cluster.

When deciding which clusters to merge or divide, the distance between the clusters needs to be calculated and there are various approaches to do so, which include maximum, minimum, unweighted average and distance between centroids. For example, when measuring the distance between two clusters C_1 and C_2, the distance between the two clusters by the maximum approach can be expressed as $\max\{d(p_1, p_2): p_1 \in C_1, p_2 \in C_2\}$ *i.e.* the distance between the two clusters C_1 and C_2 is set to the distance between the two furthest points, p_1 and p_2, where p_1 is from C_1 and p_2 is from C_2. The opposite of the maximum approach is the minimum expressed as $\min\{d(p_1, p_2): p_1 \in C_1, p_2 \in C_2\}$, where the distance between C_1 and C_2 is the distance between the two closest points. The unweighted average approach, also known as the

unweighted pair group method with arithmetic mean algorithm (UPGMA), calculates all pairwise distances between all points in C_1 and C_2 and sets the average to be the distance between C_1 and C_2: $\sum_{p_1 \in C_1} \sum_{p_2 \in C_2} \frac{d(p_1,p_2)}{|C_1|.|C_2|}$. The distance between centroids approach calculates the centroids of C_1 and C_2 and uses the distance between them. There are many other approaches for calculating the distance between two clusters, including weighted average and Ward's method and I refer the reader to (Rokach and Maimon, 2005) for more detail.

The concept of molecular diversity is not only used in the creation of VS libraries, but can also be implemented in the hit-to-lead optimisation stage, where a larger number of candidate compounds may be proposed than is possible to screen and synthesise. An example of such a tool is LLOOMMPPAA developed by Bradley *et al.* which designs and selects a diverse set of follow-up compounds to a fragment hit (Bradley, 2015). The LLOOMMPPAA workflow involves the generation of follow-up compounds *in silico*, generation of 3D conformers, filtering out those that clash with the protein, calculation of protein-ligand interaction fingerprints (PLIFs) (Section 1.2.4.3) and picking a subset based on PLIF diversity. They demonstrated the LLOOMMPPAA method by designing follow-up compounds to a fragment hit of the PHIP bromodomain. This resulted in the synthesis and biological testing of 16 compounds and from these, 12 were active, which also showed a diverse range of interactions with the protein in their crystal structures.

1.2.3 Ligand-Based Virtual Screening

Ligand-based virtual screening, or LBVS, does not require any information about the structure of the protein target, but instead relies on a set of known actives/inactives. Examples of LBVS techniques include similarity-based methods, substructure

searching, quantitative structure-activity relationships (QSAR), pharmacophore-based methods and machine learning methods. I shall discuss each of these in brief detail; however, for a comprehensive review, I refer the reader to the literature (Scior et al., 2012; Ripphausen et al., 2010; Gimeno et al., 2019; Böhm and Schneider, 2000).

1.2.3.1 Molecular Similarity

The premise of similarity-based methods is the *molecular similarity property principle* which states that structurally similar molecules tend to have similar properties (Johnson and Maggiora, 1992). For example, morphine, codeine and heroin are structurally very similar and share similar properties; all show activity against opioid receptors and are highly addictive (Figure 1.4).

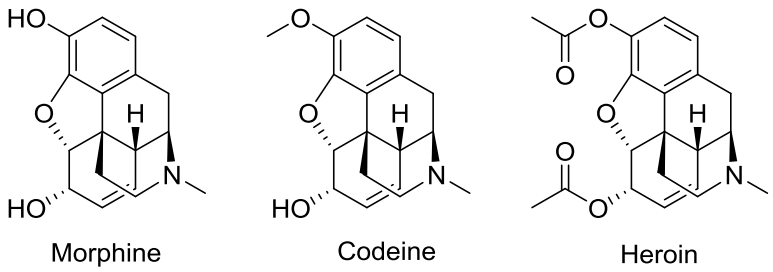


Figure 1.4. Example structures to demonstrate the similarity principle. Morphine, codeine and heroin demonstrate the similarity property principle as they are structurally very similar, and all are opioids that can be used as pain-relief and are highly addictive.

However, the caveat to the similarity principle is that it does not always hold true; compounds that are structurally very similar may have significant differences in biological activity: these instances are termed *activity cliffs* (Maggiora, 2006).

Similarity can be measured in many ways and what constitutes molecular similarity is a subjective matter. The different methods of measuring similarity differ in two components: the structural representation and the quantitative comparison between the molecular representations. Structural representations include physiochemical properties,

topological indices, pharmacophore features, molecular shapes, *etc.*, and here I will discuss some of these.

One of the most widely-used structural representation are molecular fingerprints, which are 1D vectors where each bit captures the presence or absence of a chemical feature or 2D substructural information of a molecule. These molecular fingerprints can be split into broadly four types: substructural key-based fingerprints, topological fingerprints, circular fingerprints and pharmacophore fingerprints (Riniker and Landrum, 2013).

Substructural key-based fingerprints were originally intended for substructure search through chemical libraries, in contrast to other uses such as similarity search, clustering, classification or structure-activity modelling. Each bit in the array corresponds to a substructure and the bit records the presence or absence of the specific substructure. Examples of substructural key-based fingerprints include MACCS and CACTVS FP (Rataj et al., 2018). One of the disadvantages of structural key-based fingerprints is that they do not capture the connections between the substructures and only the presence or absence of them.

Topological fingerprints, also known as path-based fingerprints, encode different atom combinations and their connectivity. Examples of topological fingerprints include atom-pair (AP) fingerprints (Carhart et al., 1985) and topological torsions (TT) (Nilakantan et al., 1987). The former encodes the atom types of a pair of atoms and the connectivity between them, whereas for the latter, each torsion is described by the four connected atoms that make up a torsion along with their atom types. For both, the atom type encodes the element, the number heavy atom connections and the number of π-electrons.

Circular fingerprints encode the circular environments of each atom up to a given radius. A widely used circular fingerprint is the extended-connectivity fingerprints (ECFP) (Rogers and Hahn, 2010). ECFP is one of the most widely used molecular fingerprint and is based on its older variant, the Morgan fingerprint (Capecchi et al.; Morgan, 1965), The process of generating an ECFP fingerprint starts with giving an initial atom identifier to each atom. Then for each atom, the chemical features, such as the element type and the bond type, of its neighbourhood, up to a given bond radius in the molecular graph are recorded into a bit string and compressed to a predefined length. Each atom's identifier is then updated by hashing the properties of the atom's environment. The process is iterated over until the prespecified number of iterations is reached then duplicate identifiers are removed. The ECFP is defined by the remaining unique set of atom identifiers. The difference between the ECFP and the Morgan fingerprint is the generation method; during the generation of the fingerprint, the ECFP algorithm keeps the results of the intermediate atom identifiers, which means the algorithm can terminate faster than the original Morgan algorithm as it does not have to reach identifier uniqueness.

2D-pharmacophore fingerprints identify pharmacophores which are parts of the chemical structure thought to be responsible for biological or pharmacological action and for example include hydrogen bond acceptor/donor groups, aromatic and hydrophobic groups. The exact definitions of the pharmacophoric features and the corresponding substructures vary between different implementations. For example, Gobbi and Poppinger defined a list of SMARTS substructures and their corresponding pharmacophoric features (Gobbi and Poppinger, 1998). For each molecule, once the pharmacophoric features have been identified, calculation of the topological distance between all triplets of pharmacophoric features is performed and then each triplet is

hashed and stored as counts or bits in the fingerprint. In addition to 2D-pharmacophore fingerprints, 3D ligand-based pharmacophores have also been a popular ligand-based representation; in contrast to 2D-pharmacophores, these require 3D conformers of the molecules and not just their molecular topology. I discuss these later in Section 1.2.3.5

It should be noted that there are also protein-ligand interaction fingerprints, PLIFs; however, this can not be regarded as a ligand-based method as it uses information about the receptor (Gimeno et al., 2019) (see Section 1.2.4.3).

There are several metrics that can be used to calculate the similarity between two molecular fingerprints. One of the most widely used is the Tanimoto coefficient (Tanimoto, 1957) and is also known as the Jaccard index (Jaccard, 1912). It is a symmetric metric with values ranging from 0 (not similar) to 1 (identical) and is calculated by:

$$S_{Tan}(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} = \frac{c}{a + b - c} \tag{1.1}$$

where the similarity between molecule A and molecule B is $S_{Tan}(A,B)$, a represents the number of bits set to 1 in molecule A, b represents the number of bits set to 1 in molecule B, and c represents the number of bits set to 1 in molecules A and B. A situation where the Tanimoto index could be used is to measure the similarity between two molecules, where the features of each molecule have equal importance.

Another similarity metric is the Tversky index $S_{Tve}(A,B)$ (Tversky, 1977):

$$S_{Tve}(A,B) = \frac{|A \cap B|}{\alpha|A - B| + \beta|B - A| + |A \cap B|} \quad (1.2)$$

where $\alpha, \beta \geq 0$. Unlike the Tanimoto similarity, the Tversky index is asymmetric provided that the values of α and β are not the same. However, the symmetric Tanimoto similarity is a special case of the Tversky similarity obtained when $\alpha = \beta = 1$.

In this thesis, I used the Tversky index in its two extreme forms, when $\alpha = 1$ and $\beta = 0$ and when $\alpha = 0$ and $\beta = 1$. The former corresponds to placing all importance on the features of A, where Equation (1.2) is reduced to Equation (1.3), and the latter corresponds to placing all importance on the features of B, where Equation (1.2) is reduced to Equation (1.4):

$$S^*_{Tve}(A,B) = \frac{|A \cap B|}{|A - B| + |A \cap B|} = \frac{|A \cap B|}{|A|} \quad (1.3)$$

$$S^*_{Tve}(B,A) = \frac{|A \cap B|}{|B - A| + |A \cap B|} = \frac{|A \cap B|}{|B|} \quad (1.4)$$

In contrast to the Tanimoto index, these two extreme forms of the Tversky index allows substructure and superstructure relationships to be identified. For example for Equation (1.3) the similarity can be interpreted as the fraction of chemical features present in reference molecule A that are also present in query molecule B. A value of 1 would indicate that molecule A is a substructure of molecule B. Hence, this extreme form of the Tversky index could be used to rank a database of query molecules by the fraction of features in common with a reference molecule.

1.2.3.2 Quantitative Structure-Activity Relationships

Quantitative Structure-Activity Relationships, or QSARs, are mathematical models that predict the biological activity from a numerical description of the molecular structure or properties (Sliwoski et al., 2014; Zhang, 2011). They are based on the assumption that the chemical structure of a molecule causes the biological response. The process of building a QSAR model typically involves the following steps: (1) identify a set of actives and inactives or bioactivity data from which to build the model; (2) split this dataset into a training, validation and test set; (3) choose descriptors that describe the molecular structure and/or physiochemical properties; (4) using the training set, generate the mathematical model that relates the molecular descriptors to the biological activity; (5) apply the model to predict biological activities for the validation set and change the model as required; (6) apply to an external test set; and (7) apply to a prospective study to propose which compounds to synthesise and biologically test. The process of splitting the data into a train, validation and test set is familiar in supervised machine learning (Section 1.2.3.3) and indeed early QSAR methods typically involved linear regression and later multilinear regression (Mitchell, 2014). The training set is used to optimise the parameters of the model, the validation set evaluates the model during training, updates the hyperparamters and minimises overfitting. Finally, after the model is trained and validated, the test set is used to evaluate the model, which should test the model's ability to generalize. The model should not see the test set beforehand, in order to provide an unbiased evaluation.

Success of QSAR models depends on the quality of the training data set and the choice of descriptors. Moreover, the model will only be valid for descriptor values that are present in the training set; hence if the prospective screening set has molecules with

descriptor values that are not represented by the training set, then the QSAR model could fail (Sliwoski et al., 2014). Hence, it is crucial that the training set is composed of a broad and diverse set of molecules and that the prospective set is appropriate *i.e.* does not include molecules that are very dissimilar to the training set (Scior et al., 2009).

Traditionally, QSAR assumes a linear relationship between the descriptors and the biological activity; however, many properties are nonlinearly related, hence, non-linear QSAR has emerged, that uses models such as neural networks, and reflects the ever increasing use of machine learning for drug discovery (Vamathevan et al., 2019).

1.2.3.3 Supervised Machine Learning

Machine learning is the use of computer algorithms that automatically learn from previous experiences to perform specific tasks without being explicitly programmed (Mjolsness and DeCoste, 2001). Machine learning can be broadly divided into unsupervised, semi-supervised and supervised techniques. Unsupervised machine learning uses unlabelled data and attempts to identify the underlying structure in the data. An example of a widely-used unsupervised machine learning method is clustering (Section 1.2.2). Semi-supervised machine learning uses data of which only some of it is labelled. Typically the model is trained on the subset of data that is labelled, then the trained model is used to predict values for the unlabelled data, and finally the model is retrained on the all the labelled and pseudo-labelled data.

Supervised machine learning uses labelled training data with independent variables, $X = x_1, x_2, \ldots x_n$ and output or dependent variables, $Y = y_1, y_2, \ldots y_n$, and the machine learning model attempts to learn the mapping function f,

$$Y = f(X) \tag{1.5}$$

Supervised machine learning can be divided into classification and regression techniques. Regression maps a set of input variables to a *continuous* output variable, *e.g.* pIC_{50} activity. In contrast, classification is where the set of input variables are mapped to discrete or *categorical* values, *e.g.* active or inactive. Examples of supervised machine learning methods include random forests (Breiman, 2001), artificial neural networks (Fausett and Laurene, 1994) and support vector machines (Cortes and Vapnik, 1995).

In contrast to similarity searching (Section 1.2.3.1), supervised machine learning requires a sufficient amount of activity data, as it needs to be split into a training set, a validation set and a testing set. What constitutes a sufficient amount of data is a tricky question as the answer is highly dependent on the complexity of the problem; however, factors such as the type of machine learning algorithm, the number of classification classes (for classification tasks) and the number of input features can also affect what is deemed sufficient. Broadly speaking, training involves building up the model by iteratively adjusting its parameters to minimise the difference between the predicted values and the actual values. Validation involves evaluating the model using the validation set and tuning the model's hyperparameters accordingly; this step can also be used to tell if the model is overfitting. Testing involves using the trained model to predict values for the test set and evaluating the trained model on data it has not seen.

An example of a machine-learning dataset could be a set of Morgan fingerprints, calculated from a set of known actives/inactives for a biological target and each having a known pIC_{50} activity. In this situation, the supervised machine learning model would attempt to learn which bits of the fingerprint are important for activity. The bits

corresponding to the important structural features would be given a high weighting in the model. This contrasts to similarity techniques where all bits are assumed to be equally important.

1.2.3.4 Shape Similarity

Shape similarity methods have recently become an increasingly popular *in silico* method in virtual screening campaigns (Kumar and Zhang, 2018). Molecules are 3D structures that have shape, which determines whether they have good complementarity to the shape of the binding pocket of a protein target. This complementarity will enable the ligand and protein to be close enough so protein-ligand interactions can be formed, which are crucial for binding. Therefore, a candidate that has high shape similarity to a known binder would also be expected to fit within the same binding pocket, assuming that there is no large conformational change to the protein binding site.

Shape similarity methods can be broadly classified into two major categories: (i) alignment-free or non-superpositional methods, and (ii) alignment or superpositional methods. The former method is independent of the placement of the reference and query molecule and generally requires less computational time to run. The latter method requires finding the optimal superposition of the reference and query molecule and results can be highly dependent on the superposition method. Alignment-based methods are computationally more expensive, but the results have been shown to be effective for identifying shape similarities and visualisation/interpretation of the results is more straightforward (Kumar and Zhang, 2018).

There are several shape representations including atom-based descriptors, volume-based descriptors and surface-based descriptors. Atom-based descriptors measure the

interatomic distances within each molecule and typically form the basis of alignment-free methods. Volume-based descriptors represent the molecule as occupying a volume, usually using one of two methods: hard sphere (Connolly, 1985) and Gaussian sphere (Grant et al., 1996; Grant and Pickup, 1995) methods. The hard sphere method treats each atom as a sphere and the volume is calculated from the union of all the spheres. The Gaussian sphere method treats each atom in the molecule as a Gaussian sphere and the molecule's volume is calculated by taking the integral over all overlapping Gaussians, using the inclusion-exclusion principle, which means that the overlap of multiple spheres is only counted once. One widely-used method that utilises shape-based screening by the Gaussian sphere method is ROCS (Kearnes and Pande, 2016) and this has been extended to also include pharmacophoric features or 'color'. The 'color' features include six types: hydrogen-bond acceptors, hydrogen-bond donors, anion, cation, rings, and hydrophobic groups. These feature groups are represented as dummy atoms, which are also described by Gaussian spheres. ROCS can measure the shape or color overlap using the Tanimoto coefficient, Equation (1.6), or the Tversky coefficient, Equation (1.7):

$$Tanimoto(A,B) = \frac{O_{AB}}{O_{AA} + O_{BB} - O_{AB}} \tag{1.6}$$

$$Tversky(A,B) = \frac{O_{AB}}{\alpha O_{AA} + \beta O_{BB} - O_{AB}} \tag{1.7}$$

where O_{AB}, O_{AA} and O_{BB}, is the volume overlap between two molecules A and B, the volume of molecule A and the volume of molecule B respectively, and α and β are the parameters for Tversky index. The shape and 'color' Tanimoto or Tversky scores can be used individually but are often combined to give TanimotoCombo or TverskyCombo respectively.

Finally, surface-based methods use representations such as the solvent accessible surface and the van der Waals surface to compare the shape between molecules.

1.2.3.5 Ligand-Based Pharmacophores

IUPAC defines a pharmacophore as "an ensemble of steric and electronic features that is necessary to ensure optimal supramolecular interactions with a specific biological target and to trigger (or block) its biological response" (Wermuth et al., 1998). Examples of pharmacophoric features include hydrogen-bond donors and acceptors, aromatic groups, positive charges, *etc*. Given a set of known active ligands for a biological target, a ligand-based pharmacophoric model can be built from the common pharmacophoric features of all the actives, which are assumed to be the features that give rise to the biological activity. This pharmacophoric model can then be used as a query to search through a virtual screening database of small molecules. The advantage of ligand-based pharmacophore searching is that a high diversity of candidates can be obtained, as they are not restricted to certain molecular scaffolds or substructures, which is a disadvantage of a method such as a similarity search with a 2D molecular fingerprint.

The general procedure for generating a 3D ligand-based pharmacophore is: (1) choose a set of active ligands from which the model will be built; (2) generate/obtain 3D conformer(s) for each active; (3) for each conformer, identify the pharmacophoric features and calculate the distances between the features; (4) align the ligands based on the maximum overlap of the pharmacophoric features; (5) generate a pharmacophore model from the pharmacophoric features that are commonly seen in the actives; (6) validate the model (Gimeno et al., 2019).

Once the pharmacophore model has been validated, the model can then be utilised in a virtual screen, where 3D pharmacophores of conformers of candidate molecules are screened for match or alignment against this pharmacophore model.

In some cases, there may be 3D experimental structural data that contains information about the active conformation of the known active ligands. In this case step (2) can be replaced with direct use of the known activate conformation. However, in the majority of cases, including the candidates used in the virtual screen, no structural data is available; hence, various methods have been used to address the conformational flexibility of molecules, such as generating a set of multiple conformers for each molecule or altering the conformation 'on the fly' when performing the pharmacophore matching/alignment (Hurst, 1994). Further detail of efficient search methods for the matching of 3D pharmacophores can be found at (Seidel et al., 2010).

Successful applications of a ligand-based pharmacophore model include Che *et al.* who used a ligand-based pharmacophore model to discover novel antagonists for the CXC chemokine receptor 2, CXCR2 (Che et al., 2018). They built their pharmacophore model using a set of eight CXCR2 actives which had high activity, similar pharmacophore features and were structurally diverse. They generated numerous pharmacophore models using HIPHOP (Discovery Studio 2.5) and used the top ranked-model for further validation. The model was then used to screen an established database containing designed scaffolds, which resulted in a compound with an IC_{50} of 76 μM and which contained a novel scaffold for CXR2. They further improved the IC_{50} to 14.8 μM by optimisation through SAR of this compound.

1.2.4 Structure-Based Virtual Screening

In contrast to ligand-based methods, structure-based virtual screening methods require knowledge of the 3D structures of the biological target, to model the potential interactions a candidate molecule can make; these favourable interactions are assumed to cause the desired biological effect. In 2016, it was reported that almost 20 drugs now in clinical use have utilised structure-based drug discovery in their developmental pipeline (Irwin and Shoichet, 2016).

Structure-based methods are typically computationally more expensive than ligand-based methods, as they involve modelling of the receptor-ligand complex and calculations to determine the complementarity. One of the most widely used structure-based methods is protein-ligand docking (Section 1.2.4.2). This technique can be fast enough to be able to screen a virtual library on the order of 10^7 molecules.

The protein structure or structures that are used in protein-ligand docking can be derived from experimental techniques such as X-ray crystallography, neutron diffraction, NMR or cryo-EM; however, if an experimentally determined structure is not available, then the structure can be modelled through homology modelling or other protein structure prediction techniques. As the majority of structural data in this thesis has been from X-ray crystallography, the basis of the technique will be briefly discussed.

1.2.4.1 X-Ray Crystallography

Currently, X-ray crystallography remains the most widely-used structure determination technique for protein-ligand structures and therefore plays a key role in the structural-based drug discovery process. The first protein-ligand X-ray structures were submitted

to the Protein Data Bank, PDB, in the 1990s (Patel et al., 2014) and since then the number has grown to over 140,000 structures in 2019, and X-ray structures account for approximately 90% of the total structures in the PDB (PDB Data Distribution by Experimental Method and Molecular Type, accessed 21 November 2019).

X-ray crystallography involves firing an X-ray beam at a protein crystal, which is usually cryo-cooled. The interaction between the X-rays and the regular lattice of electrons in the atoms of the crystal results in a diffraction pattern of X-ray spots that is then captured and processed to give information about the crystal packing symmetry and the size of the repeating unit (Smyth and Martin, 2000). The intensity of the diffraction spots give rise to *structure factors* that form the basis for the calculation of electron density. Various methods can then be applied to improve the quality of the electron density map, until the 3D structure of the protein can be confidently built using the electron density map and protein sequence. A more detailed explanation of the processing steps can be found in the review by Smyth and Martin, 2000. To obtain a protein crystal, the protein must first be purified and concentrated and kept under the correct conditions until crystals form; however, what determines the correct conditions is still not well understood. This protein crystallisation step is widely-known as the slow-step in protein crystal X-ray crystallography, and is indeed not guaranteed to work. Furthermore, certain classes of proteins such as membrane proteins are notoriously difficult to crystallise as they require a lipid membrane environment to be stable; in such cases, alternative structural determination techniques such as cryo-EM can be used instead (Murata and Wolf, 2018).

In crystal structure determination, resolution is the ability to distinguish neighbouring features in an electron density map. If the distance between two objects is smaller than the resolution then they will appear as one blob in the electron density and not as two

separate entities. The limiting factor to resolution is in theory the wavelength of an X-ray *i.e.* 1 Å. However, due to disorder in the crystal and the dynamic nature of proteins, the resolution obtained is always worse. For X-ray crystallography a good resolution is typically considered to be resolution values less than 2.5 Å (Patel et al., 2014). This is because at poorer resolutions *i.e.* greater than 2.5 Å, only the basic contour of the protein can be seen and the atomic structure must then be inferred; hence, there is less certainty in the placement of the atoms of the protein sidechains and bound ligand (if present).

When using X-ray crystal structures, other parameters of the structure are also important to consider. The B-factor or temperature factor is the per-atom positional uncertainty, in contrast to resolution which quantifies the uncertainty for all atoms. As it is per-atom, the B-factor can signify regions of mobility within the protein, *e.g.* flexible loops will have atoms with large B-factors. Another measure is the R-factor which measures how consistent the crystallographic modelled structure is with the experimental X-ray diffraction data and smaller values of R-factor indicate more accurate models, with a perfect score being 0. The R-factor is used to assess the quality of the model, so over-fitting of the data may become an issue; hence, another measure, the free R-factor or R_{free}, was introduced, which leaves out a small percentage (say 10%) of the data and refines on the rest. If the resulting R_{free} value *e.g.* the R-factor for the model built on 90% of the data, is similar to the R-factor then the crystallographic model has not been over-modelled. The difference in R-factor and R_{free} can be seen in their equations as follows:

R-factor
$$R = \frac{\sum ||F_{obs}| - |F_{calc}||}{\sum |F_{obs}|} \tag{1.8}$$

Free R-factor $$R_{free} = \frac{\sum_{testset} ||F_{obs}| - |F_{calc}||}{\sum_{testset} |F_{obs}|} \quad (1.9)$$

where F_{obs} and F_{calc} are the observed and calculated structure factors, respectively.

When inspecting a crystal structure, the occupancy values of the atoms should also be inspected. Like the B-factor, occupancy is also reported per-atom, and is the fraction of molecules in the crystal where that atom occupies that position. For example, if a ligand has two possible distinct binding modes, which occur equally frequently, then the occupancy is 0.5 for each atom for each binding mode.

There are several limitations of X-ray crystallography in drug discovery. As mentioned, one challenge is that the protein must be crystallisable. Membrane proteins, such as GPCRs, are notoriously difficult to crystallise as they require a lipid membrane environment to be stable; hence, alternative structural determination techniques such as cryo-EM may be used instead. Also the crystallisation and cryo-cooled conditions may mean that the conformation of the protein crystal structure is not representative of what it is like under its native conditions. Unlike NMR, the dynamic nature of the protein in solution cannot be studied. To use X-ray crystallography in fragment-based drug discovery (Section 1.4), it is necessary that protein crystals should be soaked at a high fragment concentration, typically >10 mM; hence crystal robustness is also crucial. This high concentration is essential for weakly binding fragments.

The use of synchrotrons for high-throughput screening has sped up the fragment screening process timeline from three months for a conventional laboratory, to one week (Spurlino, 2011). An example of high-throughput crystallography is the XChem facility at the I04-1 beamline at Diamond Light Source that offers the high-throughput fragment screening and with whom I have collaborated in this project. The screening

involves the following steps: (1) imaging and analysis of crystal plates; (2) soaking of crystals with a fragment library; (3) crystal harvesting; (4) collection of X-ray diffraction data; (5) processing of diffraction data and calculation of electron density maps; (6) analysis of electron density maps by PanDDA for detecting bound fragments (Pearce et al., 2017); and (7) refinement of models and preparation for PDB deposition. Their pipeline includes many aspects of automation and robotics to streamline each stage (Collins et al., 2018). Such a high-throughput platform generates large amounts of structural data and currently there is no consensus as to the best way of elaborating the resultant fragment-hits. One widely used method is protein-ligand docking of a virtual library of potential elaborated fragments.

1.2.4.2 Protein-Ligand Docking

Protein-ligand docking, hereafter referred to as 'docking', has two main goals: (i) predict the protein-bound conformation of a candidate compound, and (ii) estimate its binding energy to the protein. Researchers have demonstrated that the former is relatively successful, however, efficient calculation of the predicted protein-ligand binding affinity remains challenging (Plewczynski et al., 2011; Warren et al., 2006). Hence, there is a need for an alternative to prioritising candidates based solely on predicted affinity from molecular docking. Here I outline some of the main concepts of protein-ligand docking; however, for a more comprehensive review, I refer the reader to the following (Kitchen et al., 2004; Sousa et al., 2006; Huang and Zou, 2010; Yuriev et al., 2015).

Docking tools output multiple docked poses, which are the possible positions, orientations and conformations (when flexible) of a compound with respect to the protein, and each has a predicted binding affinity or docking score. A successful

docking will reproduce the experimentally observed binding mode, within 2 Å RMSD (Onodera et al., 2007; Plewczynski et al., 2011; Warren et al., 2006) and this will be the top ranked binding mode or 'pose' amongst all the generated poses.

Protein-ligand docking can be divided into rigid and flexible docking. Rigid docking is where both the ligand and the receptor are treated as rigid and the docking samples only the rotational and translational space of both *i.e.* their orientation with respect to each other. Flexible ligand docking is where the ligand is treated as flexible but the receptor is rigid, meaning that the docking involves sampling of the ligand's conformational space and the orientational space of the ligand and receptor. Lastly, the receptor can also be treated as flexible; however, due their larger size and many degrees of freedom, it is the most challenging to simulate whilst maintaining high computational efficiency (Huang and Zou, 2010).

Protein-ligand docking involves several steps: (1) protein preparation, where the protein side chains are protonated at the specified pH, missing residues and incomplete side chains are added, optimal ionisation and protonation states for protein residues are determined and optimisation of the hydrogen-bond network is performed using a restrained minimisation; (2) if possible, the binding site is identified, to restrict the search space; (3) positional, orientational and (if the ligand is flexible) conformational sampling of the compound within the target binding site to generate possible docked poses; (4) scoring of each pose to estimate the interaction energies.

In flexible docking during the conformational sampling step, a search, optimisation or sampling algorithm must be used. In almost all docking problems, the search space is too large to enumerate given all the degrees of freedom (translational, orientational and conformational). There are numerous sampling algorithms to help find the correct

binding pose, which can be divided into three broad categories: systematic search, random/stochastic search and simulation-based search (Kitchen et al., 2004). Systematic searching involves regular incremental exploration of all dimensions, and the results are deterministic. The disadvantage of this, is that the search space increases enormously with the number of rotatable bonds. As a result, docking tools that use systematic searches divide the ligand into fragments (Ewing et al., 2001) or use libraries containing pre-generated conformers (Miller et al., 1994).

Random or stochastic algorithms make a random change to the ligand's position, orientation, and/or conformation (if flexible), and based on a probability function, this change is either accepted or rejected. Examples of stochastic optimisation algorithms include Monte Carlo search and genetic algorithms. Popular docking algorithms that use stochastic search methods include AutoDock, which implements a form of Monte Carlo search, a genetic algorithm, and a Lamarckian Genetic Algorithm (Morris et al., 2009); GOLD also uses a genetic algorithm (Jones et al., 1995, 1997). Molecular dynamics is the most widely-used simulation-based method, where the atoms of the molecular system are allowed to move and interact for a set time period, according the Newton's equations of motion. However, atomistic simulations are computationally expensive so the simulation may only explore local minima.

The scoring function is also crucial to the success of the docking algorithm as it must be able to differentiate the 'true' pose from the incorrect poses and rank it highly. The methods are typically classified into five broad types: (i) force-field based; (ii) empirical; (iii) knowledge-based scoring; (iv) quantum mechanical methods; and (v) machine learning-based functions. A force field-based scoring function uses a force field which contain parameters to estimate the intermolecular interactions, typically van der Waals and electrostatics, between the molecule and the receptor, and also the

intramolecular interactions within each, but normally within only the ligand. Implicit solvation models are also frequently included to estimate the cost of desolvation of the ligand and the protein. Empirical scoring functions are based on the premise that the binding energy is the sum of individual uncorrelated terms, where the coefficients are determined by regression analysis using binding energies derived from experiments. Knowledge-based scoring functions are based on statistics derived from observations of contacts seen in crystal structures of protein-ligand complexes. The functions will score a contact more favourably if it is frequently seen. Quantum mechanical methods provide a more accurate way of determining the binding energy than force field-based methods as it accounts for all contributions to protein-ligand binding energy *e.g.* electronic polarisation, metal coordination and covalent binding, which are not accounted for in force field-based methods. Hence, it offers advantages, as it is less dependent on parameters derived for a specific system. However, the main disadvantage of quantum mechanical methods is their high computational expense and lengthy calculation times and hence are not suitable for high-throughput use (Cavasotto et al., 2018).

Machine learning-based scoring functions have recently gained popularity and represent a promising alternative to the other four classical approaches discussed above (Shen et al., 2019). As some machine learning methods are able to identify nonlinear relationships between the protein-ligand complex features and the binding affinity, these methods may offer an advantage over the classical approaches and have been reported to often outperform them (Shen et al., 2019). However, disadvantages include the possibility of the model over-fitting and the lack of interpretability of the black-box model.

Protein-ligand docking may fail for a variety of reasons, such as the tool being unable to model the flexibility of the protein. Frequently, only the ligand is treated as flexible; however, several docking tools now exist that are able to model side-chain and/or backbone flexibility. Another solution is to dock the ligand into other conformations of the protein's binding site. Dockings may fail also because the tool is unable to model water networks, insufficient conformational sampling of the bound ligand, and/or inaccurate scoring functions.

An example of a widely used open-source docking program is AutoDock Vina (Trott and Olson, 2010). AutoDock Vina is based on the AutoDock 4 program (Morris et al., 2009), but uses a different scoring function and optimisation algorithm. AutoDock Vina performs flexible docking by using a stochastic search, where a series of steps are taken, each step consists of a mutation and local optimisation, and the Metropolis criterion is used to either accept or reject each step. The mutations involve changes to the position, orientation and torsions of the docked molecule. For its scoring function, AutoDock Vina uses an empirical scoring function to predict protein-ligand binding energies. The parameters for the AutoDock Vina scoring function were tuned using the PDBbind data set as the training set (Wang et al., 2004). Compared to AutoDock 4, AutoDock Vina is approximately two orders of magnitude faster, with improved accuracy and is easier to use as the calculation of grid maps is built in and automatically performed. Like AutoDock 4, since its first report, AutoDock Vina has been used in many other *in silico* tools, one example being AutoGrow (Durrant et al., 2009, 2013). AutoGrow (version 3.0) is a *de novo* design algorithm (Section 1.2.4.4) that uses an evolutionary algorithm to grow fragments and uses AutoDock Vina as the selection operator. It also produces molecules with high synthetic feasibility by using rules of click chemistry.

Large-scale molecular docking of libraries, on the order of 10^8 and more, is not a common practice, possibly due to the known problems with accurate prediction of affinity by docking (Jorgensen, 2004). In spite of this, Lyu *et al.* recently reported docking an ultra-large library, consisting of up to 138 million diverse molecules, by using tens of thousands of computing core hours in order to discover new chemotypes for two unrelated targets: AmpC beta-lactamase and D4 dopamine receptor (Lyu et al., 2019). The screened compounds were all make-on-demand compounds, and they ordered the 44 (AmpC) and 549 (D4) top ranked compounds to be made and tested for biological activity. For AmpC beta-lactamase, they obtained an 11% hit rate, and through further screening of their analogues, improved activity was found for each hit, including a phenolate inhibitor with an activity of 77nM, which is one of the best potencies found for a non-covalent AmpC inhibitor. They obtained crystal structures of this hit, alongside other AmpC inhibitors and confirmed agreement with the corresponding docked poses. The study shows how large-scale docking can be successfully applied in a drug-discovery project to search for new scaffolds and chemotypes. With careful preparation of the VS screening library and appreciation of the limitations of docking score, molecular docking can successfully enable the discovery of novel potent molecules with new chemotypes.

1.2.4.3 Protein-Ligand Interaction Fingerprints

Protein-ligand interaction fingerprints, or PLIFs, are 1D bit vectors that encode the intermolecular interactions made between a protein and bound ligand. They were previously mentioned in Section 1.2.3.1, as they are akin to 2D molecular fingerprints that store the 2D topological information in a 1D bit vector. PLIFs can likewise be compared using similarity metrics such as the Tanimoto index. Examples of their

application in structure-based VS include database mining and filtering, post-processing docked poses to improve the performance of docking, clustering of molecules and understanding activity cliffs.

Numerous implementations of PLIFs exist, each differing in how they define the interactions and what interactions are included; however, there is still no consensus on which is the best featurisation. One of the first reports of PLIFs was in 2004 by Deng *et al.* who made structural interaction fingerprints, SIFts, and applied them to perform three tasks: (1) analysis of docking results; (2) analyse complexes of kinases with different ligands from the PDB; and (3) molecular filtering of a VS library, keeping only ligands showing desired interactions. They defined seven interaction types that represent: (1) if the ligand is in contact with a residue; (2) if the ligand is in contact with a main-chain atom; (3) if the ligand is in contact with a side-chain atom; (4) if there is a polar interaction; (5) if there is a non-polar interaction; (6) if the residue provides a hydrogen-bond acceptor; (7) if the residue provides a hydrogen-bond donor. In a SIFt, each protein residue is represented as seven bits, where each bit is either a 1 or 0, which represents the presence or absence of a particular protein-ligand interaction respectively.

Since then many other variations have been created. For example, Jubb *et al.* reported Arpeggio, an open source program based on Marcou and Rogan's definition of PLIFs (Marcou and Rognan, 2007; Jubb et al., 2017) and reports 15 interaction types. Their method is able to output PyMOL scripts for ease of visualisation of the results. It is open source and is downloadable as a standalone package from GitHub, but it also exists as a user-friendly web-server.

PLIFs have been successfully employed in a number of drug discovery projects. For example, Da *et al.* used SPLIF, Structure Protein-Ligand Interaction Fingerprints, to discover new inhibitors for Mer kinase by performing a VS using docking on a commercial database of compounds, then applied a post-docking filter to measure the similarity of protein-ligand interactions with respect to a reference structure and reported a hit-rate efficiency of 24% (Da and Kireev, 2014).

PLIFs have also been used to assess binding mode conservation. For example Drwal *et al.* used PLIFs alongside shape similarity to assess whether crystallographic additives and fragments bind in a similar way to drug-like ligands (Drwal et al., 2017). The investigation was performed on four different targets and they showed that fragment-protein complexes have largely conserved interactions with respect to their drug-like counterparts; however, conservation of binding mode was not observed for many crystallographic-additives with respect to drug-like ligands.

1.2.4.4 Postprocessing Docking Strategies

As discussed earlier, protein-ligand docking programs often generate not one but multiple possible ligand binding poses, each with an associated docking score. However, due to the difficulties in accurate pose scoring, the near native docking pose is often scored worse than an irrelevant pose; hence, there have been many research efforts that use experimental 3D binding mode information to rescore docked poses.

Marcou and Rognan used PLIFs to postprocess docking poses by rescoring the poses based on their PLIF similarity to an experimental reference structure (Marcou and Rognan, 2007). They investigated three ligand datasets and four docking tools and found that rescoring using PLIF similarity to a given reference was statistically superior

to the conventional docking scoring functions for pose prediction of the top-ranked pose. Since then, many others including Desaphy *et al.* have also investigated PLIF-based tools to postprocess docking poses to improve the docking success with regards to the top-ranked pose (Desaphy et al., 2013).

The idea of rescoring docking poses has also been explored using 3D shape based methods. For example, Anighoro and Bajorath prioritised poses based on their 3D similarity to a known crystallographic ligand and showed that for all of their four target case studies, it gave improved enrichment of active molecules compared to the native docking score (Anighoro and Bajorath, 2016b). To calculate the 3D similarity, they used a property density function, which represents each ligand atom as a Gaussian density function with a width related to its van der Waals radius. The density function of the whole molecule is then calculated by a weighted sum of the individual atom Gaussian density functions (Peltason and Bajorath, 2007).. They compared the ranking of their 3D similarity method against ranking with ROCS and against the native scoring functions of two docking programs: AutoDock 4.2.6 (Huey et al., 2007; Morris et al., 2009) and MOE (Molecular Operating Environment, Chemical Computing Group, Inc), which were used to carry out the docking. They found that their 3D similarity method consistently performed better in terms of AUCs than OpenEye's ROCS and the scoring functions of the two docking programs.

However, all these rescoring methods rely on the docking to generate at least one good pose. Hence to overcome this shortcoming, Kumar and Zhang reported a slightly different approach which involves the superposition of a ligand conformer with high 3D similarity to a known reference ligand, followed by pose refinement using an energy minimisation and side-chain repacking (Kumar and Zhang, 2016a). They used the CSAR 2012 and 2014 benchmark studies containing co-crystal structures from eight

proteins to evaluate their method against state-of-the-art-docking-tools such as RosettaLigand (Meiler and Baker, 2006) and Glide (Friesner et al., 2004) and found that their method performs best, achieving generally better RMSDs with respect to the native ligand binding mode. Furthermore, they concluded that their method has improved performance when reference crystal ligands with high structural or high 3D shape similarities are available. This could be argued as a common observation for all of these methods that rescore docked poses using an experiment reference structure.

In contrast to these reranking methods that use a similarity to a reference structure, consensus scoring has also emerged as method of rescoring and postprocessing docking results (Charifson et al., 1999). Consensus scoring, in terms of protein-ligand docking, is typically when more than one docking program is used to score each molecule and their multiple scores or ranks are combined to yield an overall score that has been shown to give higher success rates in virtual screens when compared to using any of the individual scoring functions. For example, Kukol investigated using three docking programs: AutoDock Vina (Trott and Olson, 2010), AutoDock 4.2 (Huey et al., 2007; Morris et al., 2009) and GemDock (Yang and Chen, 2004) separately and also through various consensus methods (Kukol, 2011). When used separately, AutoDock Vina performed best; however, the most consistent performance was achieved using a consensus score made from the simple combination of AutoDock 4.2 and AutoDock Vina. They measured the performance in terms of early enrichment of known ligands and for their validation set, they used a selection of targets from the Database of Useful Decoys (DUD) dataset (Huang et al., 2006). More recently, Ericksen *et al.* used machine learning techniques to develop two consensus scores which combine eight difference docking scores and demonstrated that their machine learnt approaches achieved better performance than traditional consensus scoring methods, which

performed better than the individual docking scores (Ericksen et al., 2017). They also used ROC AUC and early enrichment to assess the performance on 21 targets from the DUD-E dataset (Mysinger et al., 2012).

1.3 *De Novo* Molecular Design

Virtual screening is performed on a library of compounds which can come from a commercial chemical supplier, a company's in-house compound collection, or popular virtual screening databases such as ZINC. The compounds in these libraries, however, are not novel and represent only a small proportion of chemical space; hence, there is a high possibility that more promising compounds can be found outside of the library.

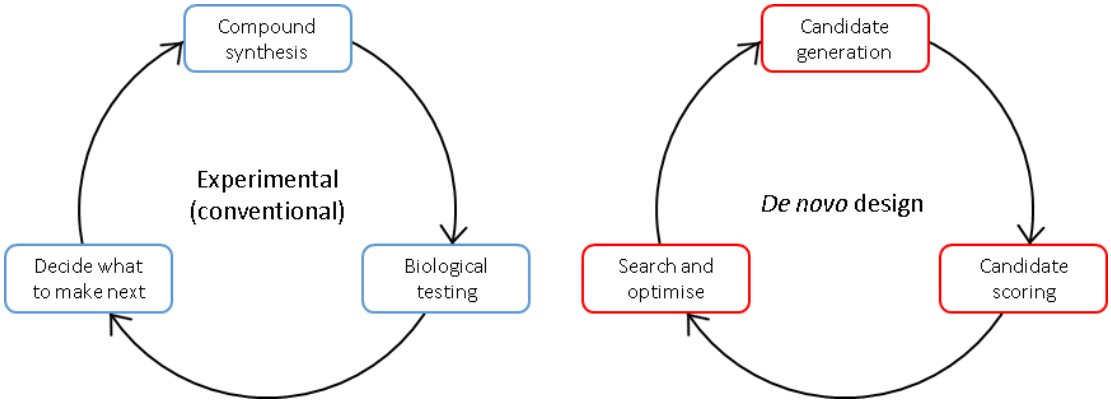


Figure 1.5. The design, make, test experimental cycle is traditionally performed by medicinal chemists. *De novo* design mimics this iterative cycle.

An alternative approach is *de novo* design, which builds molecules *ab initio* to fit a set of constraints, such as the target's binding site. Many *de novo* design tools exist and they differ in how they generate candidates, score them, and the search/optimisation algorithm they use (Hartenfeller and Schneider, 2010, 2011). The iterative cycle of *de novo* design can be likened to the experimental cycle that medicinal chemists traditionally perform in the hit-to-lead discovery process (Figure 1.5). However, one of the biggest challenges that *de novo* design still faces is building in the chemical

synthetic knowledge to ensure synthetic tractability of the novel compounds they propose.

1.3.1 Multi-Objective Optimisation

It is now widely accepted that if a drug candidate is to be successful, then it needs to satisfy multiple pharmaceutically important constraints (Nicolaou and Brown, 2013). Multi-objective optimisation approaches have become commonplace in *de novo* design.

Multi-objective optimisation involves the simultaneous optimisation of two or more objectives; for example, in hit-to-lead optimisation, typically potency is the main objective; however, if other objectives such as synthetic feasibility, solubility, pharmacokinetics, toxicity, selectivity *etc.* are considered early on, then lead compounds are less likely to fail in the later stages of the drug discovery pipeline (Lusher et al., 2011).

In contrast to single objective problems, a simple sort cannot be applied to pick the best n solutions. If the objectives are conflicting and there is not an obvious best-ranked solution *i.e.* one that is highest ranked in all objectives, then choosing the optimal solution is non-trivial. In this case, a *Pareto* rank can be given to each solution, where the rank is the number of other solutions that have better scores in all considered objectives *i.e.* a solution with a Pareto rank of zero has no better alternative solution. Solutions with a Pareto rank of zero are called non-dominated solutions or the Pareto-front (Figure 1.6).

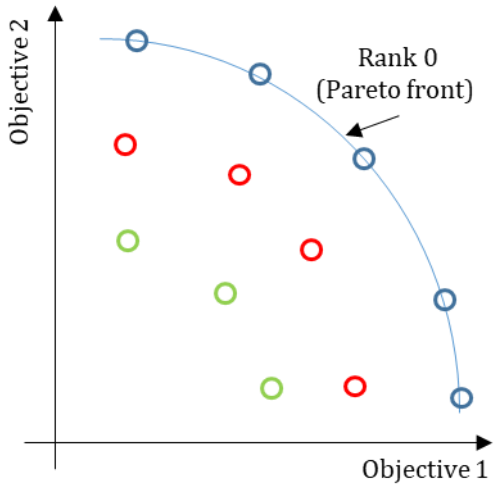


Figure 1.6. Illustration of a Pareto front for a multi-objective optimisation problem involving two compromising objectives. The blue circles represent the solutions that are Pareto rank zero *i.e.* there are no other solutions that dominate them in all objectives considered. The blue curve represents the Pareto-front. The red and green circles represents solutions that are Pareto rank one and two, respectively.

Many *de novo* design algorithms have employed multi-objective optimisation, one example being MOARF, the Multiobjective Automated Replacement of Fragments (Firth et al., 2015). Their workflow uses a rule-based fragmentation scheme (SynDiR) with a pharmacophore fingerprint-based fragment replacement algorithm (RATS) and the potential solutions can be ranked by a ligand-based and/or structure-based multi-objective scoring algorithm. They applied the workflow to optimise seliciclib, an inhibitor of CDK2. In this example application, they considered three ligand-based objectives: the restriction of a physicochemical property space (clogP), ligand-based shape similarity to a known active molecule and predicted biological activity. The activities of the *de novo* structures were predicted using a Random Forest model trained on measured IC_{50} values of 196 input compounds and ECFC4 fingerprints of length 1024 were used as the feature space (Rogers and Hahn, 2010). They synthesised and biologically tested the top performing molecules and the optimal solutions demonstrated biological activity and improved human metabolic stability. Their aim was to produce synthetically feasible solutions in a rapid and objective way. It is

interesting to note that this exemplar application of MOARF was ligand-based and used no knowledge of the binding site structure.

1.3.2 Compound Generation

For a *de novo* design method to be successful and experimentally validated, the synthetic accessibility of the *de novo* design candidates must be considered. Methods to generate compounds can be classed broadly as atom-based or fragment-based (Hartenfeller and Schneider, 2011). Atom-based approaches build up molecules atom-by-atom, whereas fragment-based approaches use molecular fragments. Although atom-based approaches are more likely to sample more chemical space, many molecules generated by this method may not be chemically sensible or drug-like, which is the main reason why atom-based approaches are less popular recently. Fragment-based approaches have the advantage that the molecular fragment building block library can be restricted to only substructures that have been found in drug-like molecules.

RECAP (REtrosynthetic Combinatorial Analysis Procedure) exemplifies one approach where 'connection rules' are derived from organic reactions (Lewell et al., 1998). RECAP rules consist of eleven bond cleavage rules (Figure 1.7a). The authors suggest several applications of RECAP, including library design where the rules are applied to databases of biologically active molecules to identify so-called *privileged* motifs and structures. The same set of rules can be applied to reconstruct custom libraries that contain the privileged motifs. Since their introduction, RECAP rules have been implemented in a number of *de novo* design tools (Fechner and Schneider, 2006; Dean et al., 2006). Privileged structures are those that appear frequently in a set of active compounds for a given target family and are therefore associated with biological

activity (Evans et al., 1988). The concept has been popular for many years in library design.

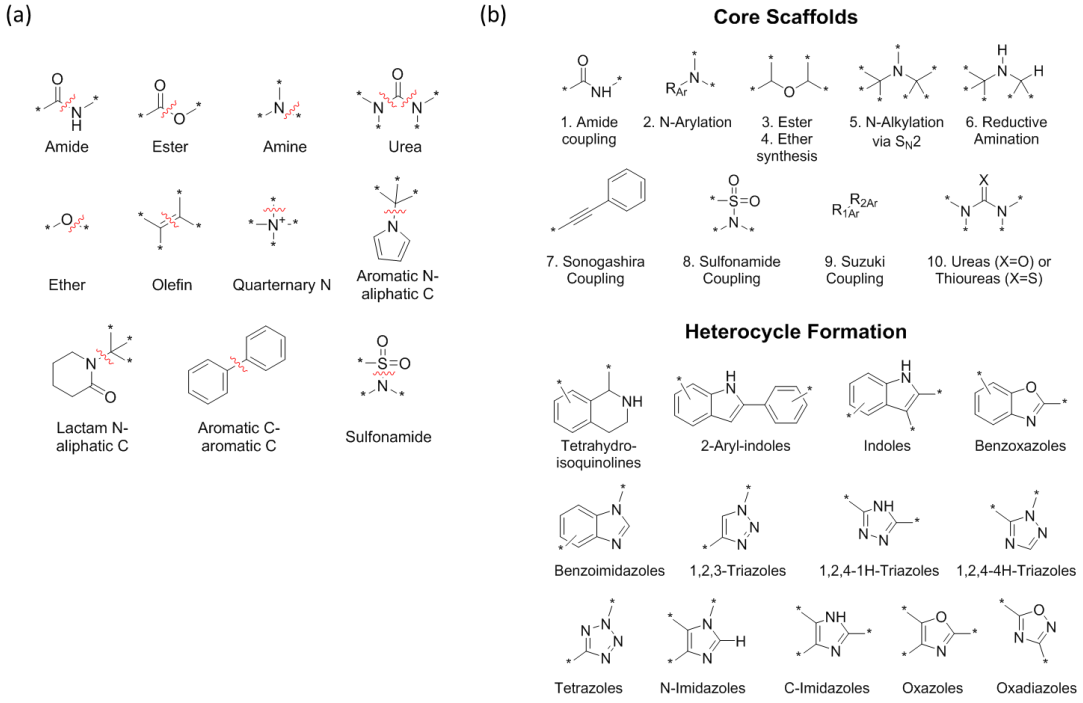


Figure 1.7. (a) The eleven disconnection rules of RECAP. Figure is adapted from (Lewell et al., 1998). The red wavy lines represent the bond that is broken. (b) Core scaffold and heterocycle substructures used to create poised fragment library. Core scaffolds are derived from the most commonly used reactions used by medicinal chemists as defined by Roughley *et al.* (Roughley and Jordan, 2011). Heterocyclic substructures originate from Hartenfeller *et al.* (Hartenfeller et al., 2011). Figure is adapted from (Cox et al., 2016).

One way of achieving synthetic tractability is limiting the number of reactions by which the building blocks can react and only screening compounds that contain the selected synthetic disconnections. One example of this approach was the creation of a 'poised' fragment library by Cox *et al.* which led to the discovery of the first reported inhibitors of the atypical bromodomain, PHIP(2) (Cox et al., 2016). They defined poised fragments as those that can be synthesised using robust and general reactions; thus, in order to create the poised fragment chemical space, they only experimentally screened fragments that contained a certain substructure (Figure 1.7b). The substructures were derived from the ten most used reactions in drug discovery, as defined by Roughley *et al.* (Roughley and Jordan, 2011), in addition to twelve heterocycle forming reactions as

defined by Hartenfeller *et al.* (Hartenfeller et al., 2011) and their own oxazole reaction (Spencer et al., 2012; Joerger et al., 2015). These reactions will be referred to as 'poised reactions' in Chapter 2.

However, the concept of reaction vectors (RVs) by Patel et al. promises a wider scope of reaction types and should reveal more chemically feasible space (Patel et al., 2009). Reaction vectors are knowledge-based representations of organic transformations that encode the topological changes that take place in a chemical transformation, alongside the environment in which it occurs. These transformations can then be applied to unseen starting materials for *de novo* design (Hristozov et al., 2011). The knowledge-based attribute can ensure that every reaction in the database has precedent in the literature and thus the resulting molecules should have higher likelihood of synthetic tractability. Validation of this *de novo* design tool was shown by reproducing known products in the knowledge base, in addition to reproducing known synthetic routes. However, one limitation of RVs is that they can only be applied in the "forward" sense – currently they have no retrosynthetic ability. Moreover, reaction vectors may address the first challenge in *de novo* design by providing greater access to synthetically feasible chemical space with a high degree of confidence but the next stage of candidate scoring and selection becomes ever more challenging. The application of such widely applicable reaction vectors in combination with a large knowledge-base leads to the inevitable combinatorial explosion in *de novo* design.

Recently, Ghiandoni *et al.* have reported a data-driven reaction classification model (Ghiandoni et al., 2019). Their model can classify any input reaction into one of the 336 reaction classes that they made and their proposed four-level hierarchical classification system allows investigation of the results at different levels. Such a reaction classification tool has potential to combat the combinatorial explosion that results from

reaction enumeration, as it can sort the results by the class of reactions involved; thus it has high usability with medicinal chemists.

A closely related topic to synthetic feasibility in *de novo* compound generation, is reaction prediction, which has recently received much interest, especially in data driven approaches such as machine-learning systems (Engkvist et al., 2018). Elements of reaction prediction include: (i) given reagents and reactants, what are the major product(s) and corresponding yield(s)? (ii) What are the best reaction conditions? (iii) Given a molecule, what is the best retrosynthetic route?

Recently Coley *et al.* used a database of hundreds of thousands of reactions from granted US patents to train a graph-convolutional neural network for reaction outcome prediction *i.e.* predicting the products of a reaction given the reactants, reagents and solvents (Coley et al., 2019a). They were able to correctly predict the major product for over 85% of cases and the model performed as well as expert medicinal chemists. Furthermore, they then reported integration of their method into automated synthesis that uses a robotic flow chemistry platform and their integrated artificial intelligence (AI) techniques are able to perform retrosynthesis, recommend reaction conditions and evaluate the best synthetic forward route. Although the platform requires some human intervention, it is nevertheless a major milestone in the field of automated synthesis and shows the potential of automated synthesis fuelled by AI (Coley et al., 2019b).

1.4 Fragment-Based Drug Discovery

The process of hit identification is crucial as it can determine the overall likelihood of success of a drug discovery project. The pharmaceutical industry has traditionally relied on automation and robotics to perform high-throughput screening (HTS) to quickly

identify small molecule leads. This method screens large libraries of compounds (10^5 - 10^6). However, one of the key challenges is ensuring diversity and quality in the screened library as its size is minimal compared to that of chemical space. Fragment-based drug discovery (FBDD) has emerged as a complementary and alternative approach to HTS and offers several advantages over HTS, such as a higher coverage of chemical space, since chemical space grows exponentially with the count of heavy atoms (Mashalidis et al., 2013; Murray and Rees, 2009).

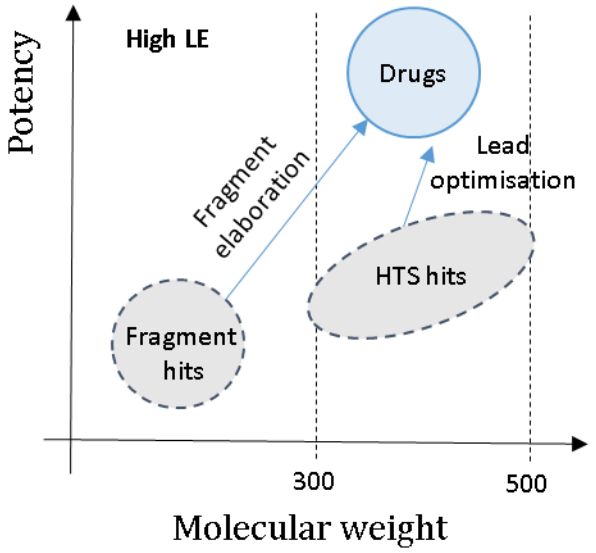


Figure 1.8. Optimisation of fragment hits versus optimisation of HTS hits. Fragment hits typically have lower molecular weights and lower potency than High Throughput Screening (HTS) hits. Optimisation of fragment hits involves elaboration to molecules of increased molecular weight, whereas optimisation of HTS hits involves a more local search that involves smaller changes in molecular weight. The upper limit of 300 for molecular weight is one of the Rule of Three (Congreve et al., 2003) and is shown by one of the dashed lines. The other dashed line at molecular weight 500 represents the rule in Lipinski Rule of Five (Lipinski et al., 1997). Figure is adapted from (Scott et al., 2012).

Fragments are small organic molecules with relatively weak inhibition constants (100 μM - 10 mM) and typically follow the rule of three (Congreve et al., 2003): molecular weight < 300 Da, number of hydrogen-bond donors ≤ 3, number of hydrogen-bond acceptors ≤ 3 and cLogP ≤ 3. After a fragment library is screened, the fragment hits are then elaborated by synthetic chemistry and 'grown' to give molecules with higher molecular weight, and ideally, greater affinity and selectivity (Figure 1.8). However,

one of the drawbacks in fragment screening is that identification of hits requires relatively sensitive experimental techniques that are able to detect weak binding.

Fragment optimisation is typically performed before any fragment growth strategy is applied. It involves improving the binding affinity or activity and drug-like properties by introducing minor structural changes, whilst maintaining an almost constant molecular weight (Figure 1.9). This strategy stems from the probability that the 'best' fragment being present in the originally screened fragment library is low.

Many methods exist to systematically explore chemical space around this fragment-hit, such as similarity search or substructural searching. However an alternative approach was recently proposed by Hall *et al.*, who recommended finding analogues to fragment-hits by construction of a graph database, which they named *The Fragment Network* (Hall et al., 2017).

They treat each molecule as a set of rings, linkers and substituents. The first node in the network is the fragment-hit itself, then new nodes are generated by removing each ring, linker and substituent and these nodes are connected to the parent node by an edge. More compounds, for example from commercial suppliers, undergo the same treatment. Each node contains information about the contained molecule *e.g.* SMILES, number of heavy atoms and number of ring atoms. Each edge contains information about the bonds that are made or broken between the molecules of the connected nodes. Once the network has been built, it can be used to query for nodes that are a certain number of edges away from the query molecule/node. The results are then grouped according to the type of transformation or edge and then sorted according to the frequency of this transformation derived from an internal compound database. Thus, the tool is

particularly attractive as it is chemically intuitive and resembles what medicinal chemists would do.

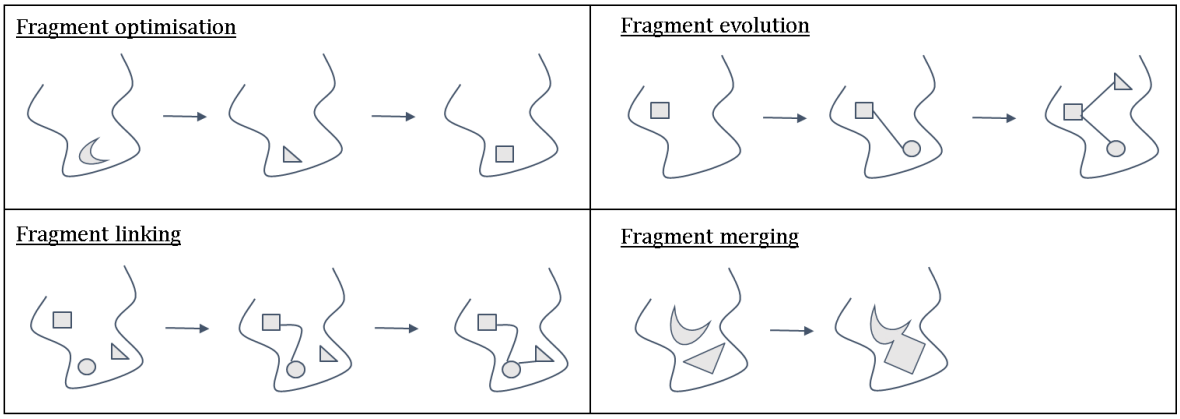


Figure 1.9. Illustrations of the different fragment optimisation and growing strategies.

After fragment optimisation, the hit-to-lead process can progress, which can be divided into three main approaches: fragment linking, fragment growing, and fragment merging (Figure 1.9). Fragment linking is when two or more fragments that bind to different parts of the binding pocket are joined to obtain a molecule with a higher affinity. Difficulties with fragment-linking include knowing whether the binding site can simultaneously accommodate two fragments and finding the ideal chemical linker that joins the two fragments, without introducing too much strain, so that the original fragment-derived parts in the linked molecule can still form the same interactions within the binding pocket. Fragment growing is the most widely-used strategy and involves the addition of functional groups that bind to additional parts of the target protein. Fragment merging combines features from several fragment hits that bind in approximately the same region of the binding pocket into one. In this work, we are interested in fragment growing strategies due to their compatibility with synthetic elaborations.

1.5 Project Aims

Drug discovery remains a challenging and lengthy process, with the failure of drug candidates often occurring late in the pipeline due to poor pharmacokinetics, lack of efficacy, or toxicity. Hence there is a need for better methods to design initial screening libraries and choose the most promising molecules with this in mind. Hit-to-lead development is often driven by the subjective decisions of medicinal chemists, often biased by what is quickest and most reliable to synthesise. The intention of this project was to develop more objective, less biased tools to explore the vastness of chemical space and to suggest equally acceptable yet novel molecules that are easy to synthesise. In particular, I focus on developing tools for the situation following an X-ray fragment screening campaign, where there is currently no agreed best method for proposing and prioritising fragment elaborations, which maximises the structural information gained from the fragment hits.

In Chapter 2, I describe a workflow that I designed that objectively proposes follow-up candidates to a fragment-hit obtained from an X-ray fragment screening campaign, such as those conducted with the XChem facility at the I04-1 beamline at Diamond Light Source (Section 1.2.4.1). The workflow involves retrosynthetic decomposition of a fragment-hit by poised reactions and then forward enumeration by poised reactions. These novel derivatives are then docked into the original fragment's protein structure thus generating potential ligand binding modes in the protein binding site. Elaborated compounds are chosen based on the conservation of binding pose observed across previously obtained X-ray protein-fragment data. I applied the workflow in a prospective study to suggest follow-up compounds for NUDT7 and experimentally validated them using a semi-automated synthesis protocol I developed and X-ray

crystallography. As it is currently unclear how best to propose and prioritise elaborated candidate fragments given the structural information obtained from a previous fragment screening campaign, I aimed to use this pragmatic approach, to highlight the challenges in the prioritisation of elaborated candidates following a fragment screening campaign and therefore addressed some of the discovered problems in the rest of my work.

One such problem was the use of RMSD for measuring the conservation of binding mode of elaborated fragments. This forms the focus of Chapter 3, where I proposed an open-source combined shape and chemical feature overlap score to use as an alternative to RMSD for such situations. The score, which we named SuCOS, was compared to COS that was introduced by Malhotra and Karinicolas. SuCOS showed very good correlation to COS, which uses the commercial software ROCS that is developed by OpenEye. I explored the strengths and weaknesses of RMSD, PLIF similarity, and SuCOS on a dataset of X-ray crystal structures of paired larger and smaller molecules bound to the same protein. My redocking and cross-docking studies reveal advantages of SuCOS over RMSD and PLIF similarity. When redocking, SuCOS produces fewer false positives and false negatives than RMSD and PLIF similarity; and in cross-docking, SuCOS is better at differentiating experimentally-observed binding modes of an elaborated molecule given the pose of its non-elaborated counterpart, including cases where there is no exact substructural match with the smaller ligand.

In light of these results in Chapter 4, I extended the investigation of SuCOS to its use in virtual screening and classification of actives from decoys, using the DUD-E set. I compared the native AutoDock Vina score to rescoring with SuCOS on the same docked poses of the DUD-E dataset and found that rescoring with SuCOS was better on average at discriminating actives over decoys. The SuCOS calculation compares the binding pose of a single reference and a single query molecule; however, when there

are multiple reference molecules, as is often the case for the results after a fragment screen, it is not clear how best to combine the multiple SuCOS values for each query molecule into one. Hence, I investigated two data fusion rules: maximum and cumulative. This investigation was performed on four diverse protein targets. However, I found no consistently superior data fusion method. Furthermore, inspection of the actives for each target, CAH2 and TRYP1, revealed that the actives showed a large proportion had anchoring groups and substructures vital for binding. Thus, it is arguable that for these targets, using a single reference molecule that contains the group which provides the anchoring interaction is a better method than using a set of fragments, which show a diverse range of binding poses, as the references.

In Chapter 5, I investigated using Bayesian optimisation in ligand-based and structure-based virtual screening. Following on from the results of Chapter 2, application of compound generation strategies such as reaction enumeration, may generate too many candidate follow-up molecules, where the synthesis and biological testing of all is not feasible. Bayesian optimisation provides an alternative method by iterative sampling with a goal to find the optimal value of an objective function within the fewest number of steps. Therefore, this method could fit in well with the design-test-make-analyse cycle. I investigated how different molecular representations and different kernels affect the performance of the Bayesian optimisation. For ligand-based VS, I used two validation datasets: MMP-12 inhibitors and anti-malaria compounds, and found that Morgan fingerprint with the Tanimoto kernel had the best performance amongst the investigated methods. For structure-based VS, I investigated two different structure-based descriptors: vectorised RDKit pharmacophoric feature maps, and protein-ligand interaction fingerprints (PLIFs) derived from Arpeggio. For validation of the vectorised RDKit pharmacophoric feature maps, four different target datasets from the PDBbind

database were used, and for validation of the PLIFs, a set of CAH2 binders was used. However, the results from the structure-based validation showed no advantage to using either of the two investigated structure-based descriptors over the 2D fingerprint methods.

Finally in Chapter 6, I describe my conclusions and future work that may be undertaken as a result of the work I have presented in this thesis.

Chapter 2 Prospective Study – Designing Follow-Up Compounds for NUDT7

2.1 Introduction

In this chapter, I outline my workflow for proposing and prioritizing candidate follow-up compounds after an initial fragment screening campaign. The workflow is pragmatic and involves reaction enumeration, protein-ligand docking and computational filtering for candidates that show a predicted conserved binding pose with respect to original fragment hit. I applied this workflow to NUDT7, a target of interest to the Structural Genomics Consortium. I generated ~2k candidate amides, and prioritised 105 to synthesise, based on four hypotheses I proposed, all focused on potential interactions formed within the binding site. I programmed an automated liquid handler to facilitate the synthesis. As a result, I soaked 78 crude reaction mixtures, deemed most promising by LCMS, into crystals of NUDT7 and obtained 5 novel crystal structures.

Through the application of my workflow to a prospective study, I aimed highlight the challenges in the prioritisation of elaborated candidates and therefore address some of the discovered problems in the rest of my work. Currently it is unclear how best to design a follow-up elaborated fragment library, given the structural information in a previous fragment soaking campaign. A carefully designed hypothesis driven follow-up library could conclude in the frequent observation of certain protein-ligand interactions

in the experimental structures, which may indicate that they are crucial for protein-ligand binding. With this in mind, I implemented hypothesis driven design with the aim to address the issue of how best to design a follow-up library in order to gain the most amount of structural information from its experimental screening.

2.1.1 NUDT7

The Nudix hydrolase family are hydrolytic enzymes that cleave nucleoside diphosphates linked to any moiety, x (Bessman et al., 1996; Daniels et al., 2015). The reaction produces nucleoside monophosphate (NMP) and the moiety linked to a phosphate group, X-P. They are found in all classes of organisms, including eukaryotes, bacteria and archaea; and eliminate potentially toxic nucleotide metabolites from the cell and regulate the concentrations and availability of many different nucleotide substrates, cofactors and signalling molecules (Gasmi and McLennan, 2001; McLennan, 2006). The catalytic amino acids are found within the 'Nudix box', which is a 23-amino acid conserved motif, GX_5EX_5[UA]$XREX_2EEXGU$, where U is a hydrophobic residue and X is any amino acid. This motif forms a short helix (Figure 2.1). Peroxisomal coenzyme A disphosphatase NUDT7 (or 'NUDT7' for short) is one of the 24 members of the Nudix hydrolase family and has become a focus of my collaborators at the Structural Genomics Consortium. NUDT7 is involved in the regulation of peroxisomal Acetyl-CoA levels. In 2017, the X-ray crystal structure of apo Human Peroxisomal coenzyme A diphosphatase NUDT7 was deposited (PDB ID: 5T3P).

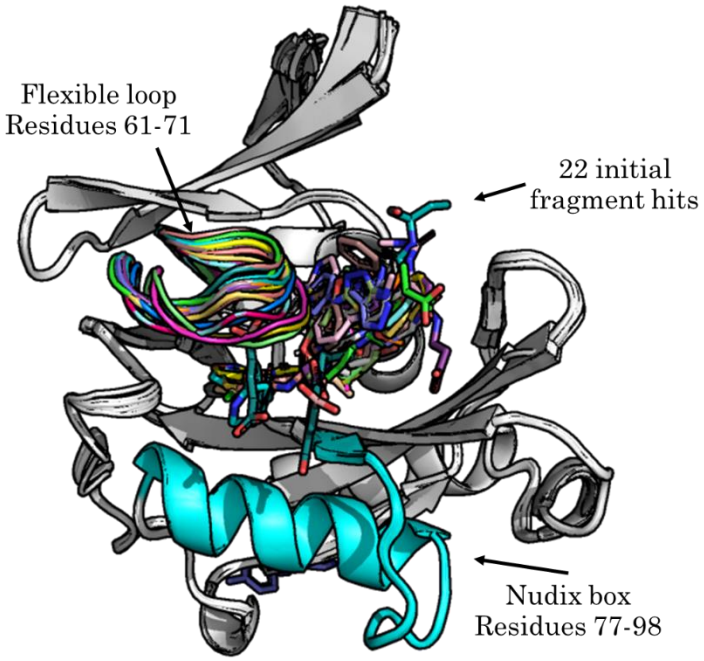


Figure 2.1. Initial fragment hits from screening NUDT7 with DSPL fragment library. NUDT7 was screened using high-throughput X-ray crystallography with the DSPL fragment library, which resulted in the initial 22 fragment hits. The structures of the 22 fragment hits are overlaid and shown. The Nudix box is highlighted by the cyan cartoon and is where the catalytic amino acids are found. The flexible loop is highlighted with the different colours representing the conformation of the loop in each structure. The fragment hits are shown in stick representation and the different colours show each fragment hit. All 22 fragment hits are seen to occupy the same binding site except one which can be seen in the background in dark blue. This figure was produced using PyMOL (Schrödinger, LLC.).

Researchers at the SGC used high-throughput X-ray crystallography to screen NUDT7 with the Diamond-SGC Poised Library (DSPL), which contains 776 fragments (Cox et al., 2016; Velupillai et al., 2018). This resulted in 22 initial fragment hit crystal structures; 21 of the fragments were found to bind in the same main binding site (Figure 2.1). In the aligned structures, a flexible loop (residues 61-71) can be seen adjacent to the binding site.

A first round of 140 follow-up compounds was synthesised by the Brennan group and soaked into NUDT7 crystals; this resulted in a further 24 fragment hit crystal structures which are shown in Figure 2.2a (Velupillai et al., 2018). Again, from the structural data, the flexible loop can be seen within the main binding site (residues 61-71). There is also a cysteine residue in the active site (CYS73) which may be interesting for the

development of covalent inhibitors. Many of the 24 hits have aromatic rings that share a conserved binding mode between ARG61 and GLU97.

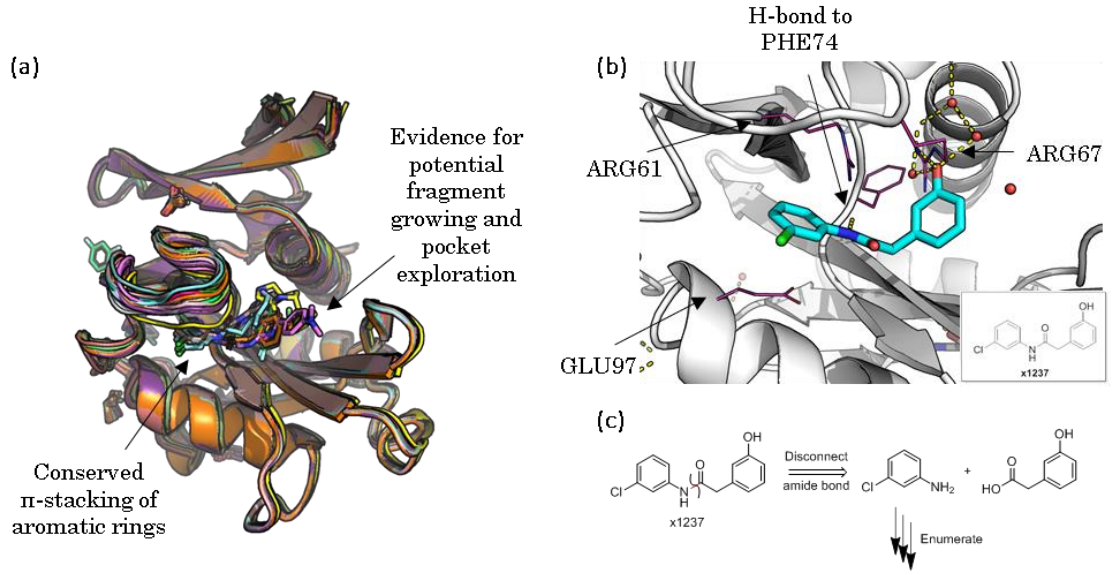


Figure 2.2. Second generation hits for NUDT7 and my follow-up strategy for one hit. (a) Soaking of the 140 first generation compounds resulted in 24 hits that are shown. (b) One of these was compound x1237 which is shown by the cyan sticks (PDB ID: 5QH1), (Velupillai et al., 2018). The dashed yellow lines represent polar interactions made with the binding pocket, one of these was a H-bond with PHE74 which is shown by the magenta lines. (c) In this chapter I have proposed compounds to follow-up from hit x1237. The x1237 amide bond (shown in red) was disconnected and the 3-chloroaniline elaborated with the forward poised amide reaction. (a) and (b) were produced using PyMOL (Schrödinger, LLC.).

2.1.2 Prior State of the Art and Chapter Aims

In this chapter, I proposed compounds to make based on one of the follow-up hits, x1237 (PDB ID: 5QH1), which has interactions that are common with the rest of the hits (PLIF heatmap shown in Appendix Figure A.3). The fragment x1237 makes a hydrogen bond to PHE74 with its amide nitrogen and the chloroaniline ring has stacking interactions with ARG61 and GLU97.

To propose compounds, I designed a workflow, described in the next section, that combines open source cheminformatics and computational approaches in structural biology, to propose elaborated molecules based on an initial X-ray crystal structure of a fragment hit, along with the available reagents and reactions necessary to make them.

These novel derivatives are docked into the fragment-protein structure and potential ligand binding modes are generated in the protein binding site of interest. The elaborations are suggested by retrosynthetic decomposition of the fragment hit by poised reactions (Cox et al., 2016) and then forward enumeration by poised reactions. Because these *in silico* reactions sample the repertoire of known synthetic chemistry, I thus ensure the search space covers synthesisable molecules. I chose elaborated compounds based on the conservation of binding pose with respect to the parent fragment hit and used numerous other filters such as the similarity of protein-ligand interaction fingerprints to all of the known experimentally-observed fragment hits as well as factoring in the cost of the reagents.

Many *in silico* workflows have been proposed to prioritised hit-to-lead candidates in structure-based drug discovery and many of them have involved enumeration, docking, scoring and filtering (Lionta et al., 2014). They all differ in how they perform each of these steps and there has been no 'one-size-fits-all' approach that works for all targets, data types and objectives. However, the approach I present in this chapter has been tailored towards the high-throughput protein soaking and screening facilities at Diamond Light Source. The approach aimed to address the question of how to best to design a follow-up elaborated fragment library that utilises the information obtained from the previous fragment soaking campaign whilst maximising the possible structural information gained from designing a hypothesis driven library to screen. From this prospective study, I also aimed to highlight the challenges in the prioritisation of elaborated candidates and therefore address some of the discovered problems in the rest of my work.

This prospective work also aimed to contribute to the ongoing research effort at the Structural Genomics Consortium to find potent inhibitors and binders for the NUDT7

target of interest. By proposing elaborated fragments to screen and generating novel protein-ligand crystal structures, I would add new structural information regarding ligand binding modes for this target.

2.2 Methods

2.2.1 Workflow for *in silico* Elaboration of Fragment Hits and Docking

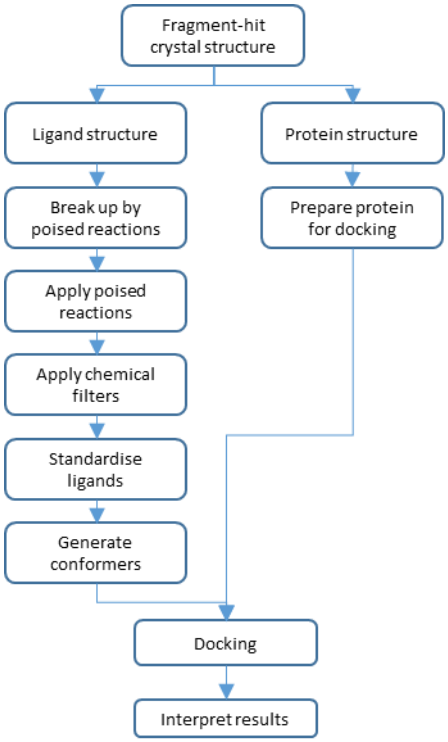


Figure 2.3 Flow chart showing the overview of the workflow to propose fragment-hit follow-ups.

I developed a workflow (Figure 2.3) that takes a crystallographic fragment hit obtained from XChem (Collins et al., 2018) and decomposes it using poised reactions (explained in Section 1.3.2) into its corresponding poised components, or ‘synthons'. Forward poised reactions are then applied to one of the poised synthons in order to search chemical space for potential molecules that medicinal chemists could rapidly

synthesise. The proposed molecules are standardised and protonated at a specified pH. Conformers are generated, which are then docked into the parent fragment's protein crystal structure to generate potential binding poses. Ligand efficiencies are then calculated and the presence or absence of protein-ligand interactions can prioritise what is made next. In the following section, I discuss in detail each step in the algorithm.

The existence of open source software and the continuous efforts towards its development encourages a collaborative environment within the research community. Open source software is accessible to all and is ready to be built upon. I have therefore focused on using mainly open source tools. My workflow used KNIME (version 3.7.0) (Berthold et al., 2008), Python (version 2.7) (Rossum, 1995), RDKit (version 2017.03.1) and Open Babel (O'Boyle et al., 2011) , while the docking component used AutoDock Vina (Trott and Olson, 2010) and AutoDockTools 4 (Morris et al., 2009). PyMOL (Schrödinger, LLC.) was used to visualize the results and generate images.

2.2.2 Reactions and Reagents

Cox *et al.* designed 'poised' fragment libraries by retrosynthetically deconstructing fragment hits into poised synthons, searching catalogues for similar synthons and then applying the poised reactions in one-step chemistry to provide fragment analogues for screening (Cox et al., 2016) (also discussed in Section 1.3.2 and shown in Figure 1.7b). Poised reactions provide highly feasible potential follow-up candidates that can be synthesised in a rapid and parallel manner.

Like Cox *et al*., the first part of my workflow uses poised reactions to deconstruct the fragment-hit into poised synthons, followed by forward poised enumeration to generate novel follow-up candidates. This methodology was implemented in a KNIME

workflow (Berthold et al., 2008) which I adapted from the XPoise workflow originally developed by Cox (Cox, 2016).

The workflow also requires a database of reagents that can be used to perform *in silico* reactions with the generated poised synthons. In order to mimic the medicinal chemist, the reagent database should be similar to what the chemist would normally use. I used the MolPort building block catalogue as one of the reagent databases, which contained 319,974 molecules (MolPort Building Blocks Database, accessed 1 December 2017). MolPort is a compound ordering service that has compound databases that are updated monthly with each compound's current availability and approximate price ranges. The choice of MolPort as the reagent database was arbitrary and other commercial compound ordering services such as Enamine could have been used. I also used the Brennan Laboratory (PEB) reagent database, which consisted of 3,931 reagents.

2.2.3 Protein Preparation for Docking

The NUDT7 protein was protonated using PDB2PQR (Dolinsky et al., 2004) with the AMBER forcefield and the PROPKA algorithm (Olsson et al., 2011; Søndergaard et al., 2011) to assign protonation states at pH 7. This pH was appropriate as NUDT7 helps to regulate CoA and acetyl-CoA levels in the peroxisome, where the pH is 6.9 to 7.1.The resulting PQR file, which contains the protein's atomic coordinates, partial atomic charges and van der Waals radii, was then converted into the PDBQT format for docking using the MGLTools script *prepare_receptor4.py* with all default parameters, which adds Gasteiger charges, merges non-polar hydrogens and outputs a PDBQT file containing coordinates, partial charges and AutoDock 4 atom types.

2.2.4 Ligand Preparation for Docking

The ligands were read in in SMILES format and were standardised by MolVS (Version 0.0.9) using the *standardize_smiles* function, and Open Babel (O'Boyle et al., 2011) was used to protonate the SMILES at pH 7. The SMILES strings were read into RDKit (version 2017.03.1) to generate RDKit *Mol* objects, then RDKit's *AddHs* was used to add explicit hydrogens to the *Mol* objects and finally RDKit's *ConstrainedEmbed* was used to generate conformers. The *ConstrainedEmbed* function was chosen as it generates a conformer where part of it is constrained to have particular coordinates as a user provided 'core' molecule. This was done with the assumption that the candidate elaborated fragments have a conserved binding mode with respect to the parent fragment. For the case of this chapter, the core molecule was identified by taking the maximum common substructure (MCS) between the parent fragment hit, x1237, and the candidate molecule, which was calculated using RDKit's *FindMCS* with *completeRingsOnly* and *ringMatchesRingOnly* set to *True*.

If the molecule has no amides or aliphatic rings then I generated only one conformer, else ten conformers were generated and clustered, keeping only those which are greater than 0.5 Å from the conformer with the lowest energy. This was done to create structural diversity in the conformers, as the AutoDock Vina docking automatically treats amides and aliphatic rings as rigid during flexible ligand docking (Section 1.2.4.2). The choice of ten conformers and 0.5 Å clustering threshold was arbitrary but the generation of more conformers would have increased the computation time taken to dock all and the use of too few conformers might mean that alternative conformations of the amides and aliphatic rings were not explored. The conformers were outputted as SDFs that were then converted to SYBYL MOL2 format using Open Babel so that bond

order information was retained. The ligand MOL2 file was then processed into the PDBQT using MGLTools script *prepare_ligand4.py* with all default parameters (Morris et al., 2009).

2.2.5 Docking

All docking studies in Chapters 2 and 3 were performed using AutoDock Vina (Trott and Olson, 2010) (version 1.2.1) using a docking box size of 22 Å on each side. The center of the docking box was chosen to be the centroid of the crystallographic ligand, x1237. AutoDock Vina generates up to nine diverse poses for each docking by default and predicts a binding affinity for each.

For each docking run, AutoDock Vina outputs the docking poses in a PDBQT file that also contains the AutoDock Vina predicted affinities. The PDBQT file was converted into a SDF file using Open Babel, that was then used for RMSD calculation. For PLIF calculation, the PDBQT file was converted into a PDB file using Open Babel and combined with the original PDB of the fragment-hit protein, without the original ligand.

2.2.6 Calculation of RMSD

A maximum common substructure RMSD (MCS-RMSD) cutoff was applied to the resulting docking poses to filter out candidates that did not give rise to any docking pose with the conserved binding pose *i.e.* for each candidate if no docking poses were found within the MCS-RMSD cutoff with respect to the parent fragment hit, x1237, then the virtual molecule can be disregarded.

For each MCS-RMSD calculation between the parent fragment hit, x1237, and a docked pose, an MCS first needs to be identified between the two molecules before the

RMSD between corresponding atoms can be calculated. Like the conformer generation step, the MCS structure between a reference (parent fragment hit) and a query (docked pose) structure was determined by the *FindMCS* function in RDKit, with both options *completeRingsOnly* and *ringMatchesRingOnly* set to *True*. It should be noted that there are several other algorithms that can compute the MCS, each differing in their approach (Englert and Kovács 2015; Kawabata 2011; Raymond, Gardiner, and Willett 2002; Duesbury, Holliday, and Willett 2018); I focused on the MCS functionality provided by RDKit as it is open source. The RMSD calculation takes into account symmetry if present in a molecule, such as a *para*- substituted phenyl ring. However, it does not take into account multiple substructure matches, for example if there are multiple MCSs present in a molecule, it will only match one of them.

2.2.7 Calculation of Protein-Ligand Interaction Fingerprints (PLIFs)

Protein-ligand interaction fingerprints (PLIFs) have been previously reported in various forms, using different definitions for their interactions (Marcou and Rognan, 2007; Deng et al., 2004; Radifar et al., 2013; Drwal et al., 2018). All aim to transform intermolecular protein-ligand interactions into a 1D bit array that represents the presence or absence of specific interactions between a part of a ligand and specific residues in the protein.

In this study, PLIFs were calculated using Arpeggio (Jubb et al., 2017). PDB files were preprocessed using the *clean_pdb.py* Python script from (Jubb, 2014), which tries to resolve common errors from PDBs that are raised when BioPython or OpenBabel, which are used by Arpeggio, is used to read in the structures. The errors it tries to resolve include dealing with multiple models by taking the first model, dealing with multiple occupancies by taking the highest occupancy and removing alternate locations,

and converting selenomethionines to methionines. I included 12 of Arpeggio's 15 interactions types: covalent, vdW, hydrogen bond, halogen bond, ionic, aromatic, hydrophobic, carbonyl, polar, metal, weak hydrogen bond and weak polar. Proximal interactions, steric clash and vdW clash interaction types were excluded. The reasoning for this was that proximal interactions, which are interactions within 5 Å but greater than the vdW interaction distance, were seen frequently and hence overwhelmed other interaction types. If multiple interactions of the same type were made to a specific protein residue then it was only recorded once. Only interactions between the ligand and the protein residues, and between the ligand and any metal ions were considered. Bridging interactions with water were not considered. However, it should be recognised that these waters should ideally be considered as they are important mediators for interactions in many protein-ligand binding sites, but automatic identification of such bridging waters is still currently challenging (Sridhar et al., 2017).

In order to quantify the proportion of protein-ligand interactions that are recapitulated in a candidate query pose with respect to the parent fragment crystal pose, I measured the similarity between the two PLIFs using the Tversky coefficient, hereafter referred to as TvPLIF, with weights $\alpha = 1$ and $\beta = 0$ for the reference and query structure respectively and was calculated using the RDKit function *DataStructs.TverskySimilarity*. In this chapter, the reference was the crystal structure of NUDT7-x1237 and the query was the docked candidate complexed with the protein conformation of the NUDT7-x1237 crystal structure.

2.2.8 Experimental procedure for amide coupling using the Opentrons liquid-handling robot

In order to create a semi-automated experimental procedure, I used the liquid-handling robot, Opentrons OT-One, to perform the following amide forming reaction:

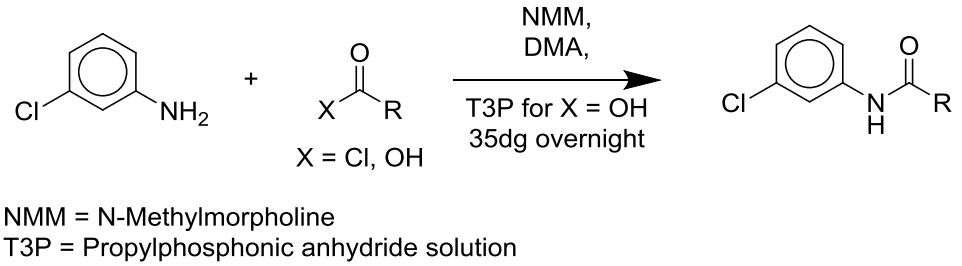


Figure 2.4. Reaction scheme for coupling and acylation reactions.

Two 96-well plates were used: one for the coupling reactions with the carboxylic acids and the other for the acylation reactions with the acyl chlorides.

I wrote Python scripts (Appendix A.1) for:

1. making stock solutions for the solid reagents;
2. mixing of reagents to initiate the reaction;
3. taking aliquots of the reaction mixture for quality control;
4. taking aliquots to be used for soaking using the crude reaction mixture.

For the carboxylic acids which were solids, 30 mg of each were manually weighed into each 96 well pot. For those that were liquids, 30 μL was manually measured out. Subsequently, all were diluted to 0.8 M solutions in DMA using the Opentrons, corresponding to step (1). Due to their volatility, the stock solutions of the acyl chlorides were manually made up to 0.8 M solutions in a 96 plate just before the reactants were mixed together.

For step (2) different reaction procedures were used for the acylation and coupling reactions and are detailed as follows, both being performed with the Opentrons robot:

Acylation: to each reaction well, amine stock solution (75 mL, 0.06 mmol, 1.0 eq) was dispensed, followed by NME (16.5 mL, 0.15 mmol, 2.5 eq) then the acyl chloride stock solution (82.5 mL, 0.066 mmol, 1.1 eq).

Coupling: to each reaction well, amine stock solution (75 mL , 0.06 mmol, 1.0 eq) was dispensed, followed by NME (16.5 mL, 0.15 mmol, 2.5 eq) then T3P (42.9 mL of 50% solution in ethyl acetate, 0.072 mmol, 1.2 eq), then the acid stock solution (78.75 mL, 0.063 mmol, 1.05 eq).

Both coupling reactions were left in a shaker overnight at 35°C and aliquots taken the following morning for mass spectrometry to monitor the progress of the reaction (step 3). Aliquots were also dispensed into a 384 plate ready for soaking of the crude reaction mixtures into NUDT7 crystals (step 4). Both aliquots for LCMS and for crude reaction soaking were dispensed with the Opentrons.

2.2.9 Soaking crude reaction mixtures into NUDT7 crystals

Crude reaction mixtures were directly soaked into NUDT7 crystals using the LabCyte Echo liquid handler at a concentration of 20% (v/v) (calculated from the initial drop volume). The plates were resealed and the compounds left to soak for one hour at room temperature before the crystals were mounted in nylon loops and immediately flash frozen in liquid nitrogen.

2.2.10 Data Collection and Structure Solution

All datasets were collected on beamline I04-1 at Diamond Light Source. Initial electron density maps were calculated with DIMPLE (Wojdyr et al., 2013) and inspected with COOT (Emsley et al.). PanDDA (Pearce et al., 2017) was used for hit identification.

2.3 Results and Discussion

2.3.1 Choosing Follow-Up Compounds to x1237

The workflow was applied to fragment-hit x1237 (Figure 2.2c).The amide bond was disconnected and the 3-chloroaniline moiety kept constant due to its conserved interactions seen across multiple fragment hits, and the fact that structural data from previous fragment hits showed potential for exploration of the binding pocket on the other side (Figure 2.2a). Note that this would also displace water molecules from the binding site, which might be entropically favoured (Figure 2.2b).

The resulting poised 3-chloroaniline fragment was subjected to forward poised amide enumeration. Two reagent databases were used: the MolPort building block catalogue (MolPort Building Blocks Database, accessed 1 December 2017) and the PEB reagent database, which contained 312,974 and 3,931 compounds respectively (Figure 2.5). Both databases were filtered using an adapted version of the “Amides (1A)” synthon building block filter from the XPoise workflow (Table 2-1), developed by Cox, 2016. This building block filter identified possible carboxylic acids and acyl halides that could react. It also contains additional filters, such as filtering molecules with amine groups and a REOS (Rapid Elimination Of Swill) filter which aims to eliminate potentially problematic compounds (Walters et al., 1998). For the MolPort database, this resulted

in 34,847 carboxylic acids and 1,609 acyl chlorides, while for the PEB database this gave 338 carboxylic acids and 79 acyl chlorides.

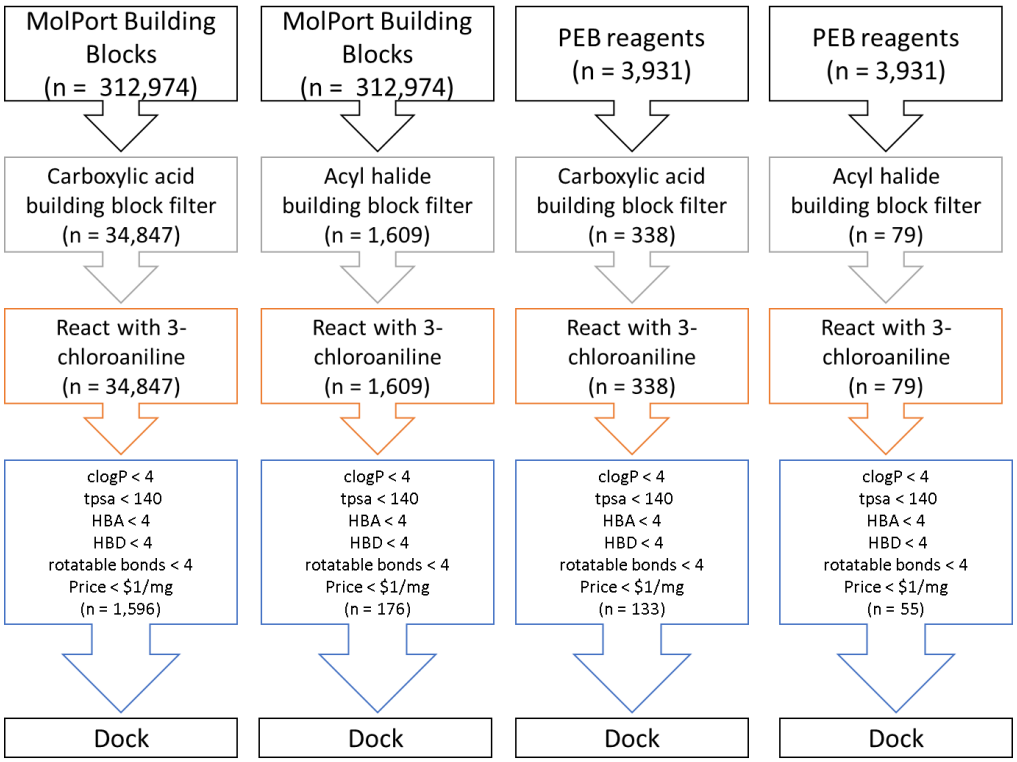


Figure 2.5. *In silico* filtering of the MolPort and PEB reagent databases that led to the proposed amide follow-ups of fragment-hit x1237 for docking to the protein of NUDT7-x1237 crystal structure. Each arrow shows the number of molecules (n) at each stage.

These acids and acyl halides were then 'reacted' with 3-chloroaniline using the RDKit Two Component Reaction node with the reaction SMARTS described in Table 2-1 to produce the product amides. The *RDKit Descriptor Calculation* KNIME node was then used to calculate physicochemical, topological, and other descriptors for each amide. These were then used to filter according to the following rules: cLogP < 4; tPSA < 140; number of hydrogen-bond donors < 4; number of hydrogen-bond acceptors < 4 and number of rotatable bonds < 4. There was no filter for the molecular weight; however, the molecular weight distribution of the candidate amides ranged from 169 to 398, with a median of 291, which corresponded to a median molecule weight gain of 164 with respect to 4-chloroaniline (Supplementary Figure A.1). These filtering rules are

reminiscent of the rule of three and rule of five; my intention was to create filters for the intermediate stage between the two. The molecular size of the elaborated fragment follow-ups are likely to increase with respect to the original fragment hit; however, a gradual increase in molecular size corresponds to a larger coverage of chemical space and gives more opportunities to optimise intermediate molecules in the hit-to-lead process.

I also filtered according to price < $1/mg, and no Pan Assay INterference compoundS (PAINS) (Baell and Holloway, 2010). By matching the substructural features described by Baell and Holloway that frequently generate false positives or "PAINS", it is possible to eliminate compounds that might display non-specific binding across multiple targets in high-throughput biochemical screens. Saubern *et al.* published a list of such structural features to help identify such problematic structures and the SMARTS strings from this list were used to filter out compounds with matching substructure (Saubern et al., 2011). This gave 1,596 and 176 product amides from the MolPort carboxylic acids and acyl chlorides, respectively, and 133 and 55 amides from the PEB carboxylic acids and acyl chlorides respectively.

At the time this study was performed, no deduplication step was carried out until the end of the workflow to ensure there was no overlap between the MolPort and PEB library. However, upon review of these two libraries, there was considerable overlap: 62/133 (47%) of the PEB amides made from carboxylic acids and all 55 (100%) of the PEB amides made from acyl halides were found to be present in the corresponding MolPort library. If this study is to be repeated in the future, duplicates should be removed at the start of the workflow.

XPoise Step	Description	Filter Value	SMARTS
Building block filters	Find acid chlorides	≥ 1	[Cl,Br,I]C=O
	Find carboxylic acids	≥ 1	[H]OC=O
	Remove primary amines	0	[#7H2]
	Remove secondary amines	0	[NH1] (keep [#7H1][C,S]=O)
	Keep secondary (sulfon)amides	≥ 1	C(=O)NC(=O)
	Remove hydrazines and hydroxylamines	0	NN N-[OH]
	REOS filter	N/A	N/A
Reaction SMARTS	For acid chlroides	N/A	[Cl,Br,I][C:2]=[O:3].[#7:1]>>[#7:1][C:2]=[O:3]
	For carboxylic acids	N/A	[OH1][C:2]=[O:3].[#7:1]>>[#7:1][C:2]=[O:3]
Product Filters	cLogP	< 4	N/A
	TPSA	< 4	N/A
	# H-bond donors	< 4	N/A
	# H-bond acceptors	< 4	N/A
	# of rotatable bonds	< 4	N/A
	Price	< $1/mg	N/A
	Filter out PAINS	N/A	See Saubern et al. (Saubern et al., 2011)

Table 2-1. SMARTS strings used in the building block filter to filter out the acyl halides and carboxylic acids used in the virtual coupling with 3-chloroaniline. Reaction SMARTS for forward poised amide enumeration with 3-chloroaniline. Filters specifying the physicochemical requirements for the amide products. This table is adapted from Cox, 2016.

The resulting product amides were docked into the protein conformation of the NUDT7-x1237 crystal structure using the protocol described in Section 2.2.5. MCS-RMSDs and TvPLIF scores were calculated for each docked pose of each product amide with respect to the crystal structure of x1237. The docked poses were filtered to select only those that made an H-bond to PHE74 and furthermore, those with an MCS-RMSD < 2 Å.

Currently, it is unclear how best to pick follow-up compounds to make, while using the most structural information from a fragment soaking campaign. Four hypotheses were

proposed based on the evidence obtained from the existing X-ray crystal structures (Figure 2.6):

(a) formation of an hydrogen-bond or polar interaction with CYS73 is required for binding;

(b) interaction with greasy pocket (any interaction with all four residues TYR41, VAL43, ILE122 and MET192) is required for binding;

(c) choosing candidates that show diversity in PLIFs ensures that at least some candidates make the protein-ligand interactions required for binding. To create the PLIF diversity, the candidates were clustered and those that had the best Pareto rank in each cluster (the objectives for the Pareto rank are discussed later) were chosen;

(d) choosing candidates that show diversity in PLIFs by clustering on PLIF, but picking the worst Pareto Rank from each PLIF cluster (the objectives for the Pareto rank are discussed later). This hypothesis was to act as a control to (c) to see if the Pareto ranking produced any enrichment in binders.

Hypotheses (a) and (b) are based on specific interactions and the decision to explore these particular interactions was based on interactions in previously observed NUDT7-fragment X-ray crystal structures. Choosing which particular interaction to explore was somewhat arbitrary. Hypothesis (c) was designed to choose compounds that sample a diverse range of interactions within the pocket. Moreover, hypothesis (c) prioritised candidates based on multiple objectives (discussed in more detail later) that involve prioritising candidates that show protein-ligand interactions that have already been seen in the other fragment hit crystal structures, whilst minimising reagent cost. Hypothesis (d) was chosen so as to act as a control with respect to the third hypothesis, to see if Pareto ranking resulted in more candidates that bind.

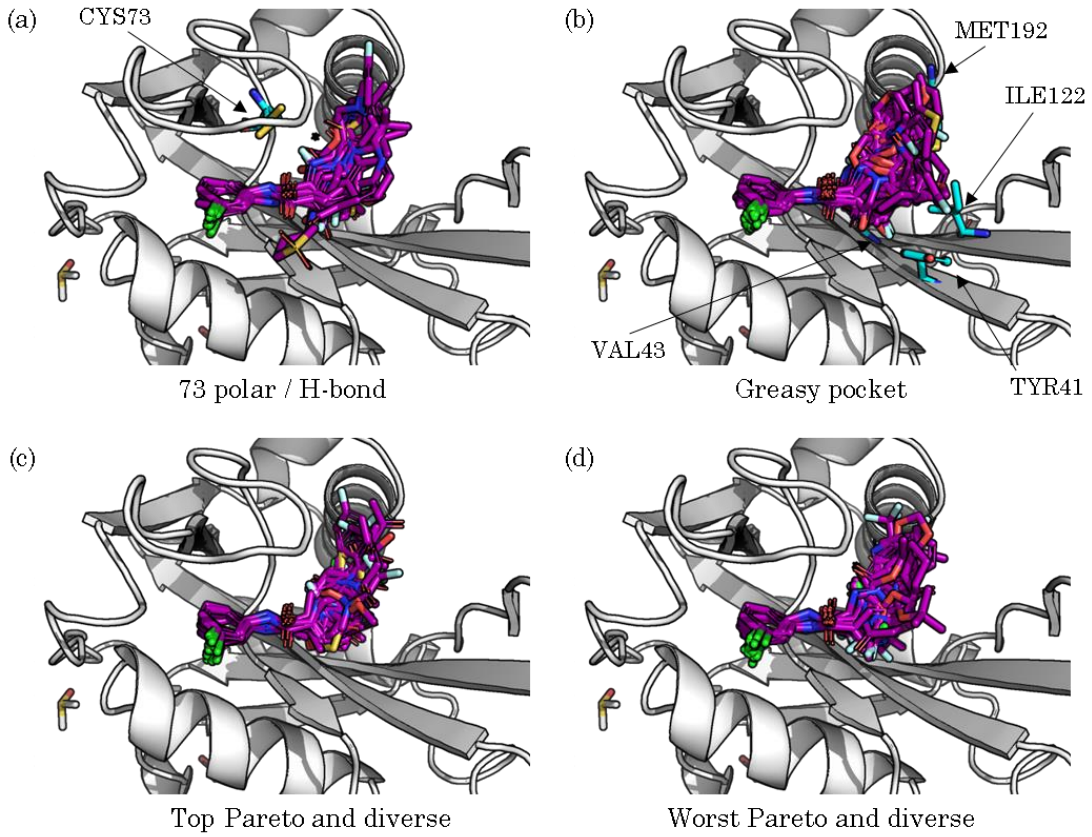


Figure 2.6. Four hypotheses were made to help choose which follow-up compounds to make: (a) Formation of an H-bond or polar interaction with CYS73 (cyan sticks) is required for binding; (b) Interaction with the greasy pocket *i.e.* any interaction with residues TYR41, VAL43, ILE122 and MET192 (cyan sticks), is important for binding; (c) Choosing candidates with diverse PLIFs and best Pareto rank will sample the binding pocket and hence generate at least some candidates that make the protein-ligand interactions essential for binding; (d) Choosing candidates with diverse PLIF and worst Pareto ranking, for same reasons as (c) but tests whether Pareto ranking produces better candidates that are more likely to bind. This figure was produced using PyMOL (Schrödinger, LLC.).

The docked poses were filtered according to these four hypotheses. As the aim was to prioritise candidates for synthesis and consequent crystal soaking, the plan was to propose no more than 200 compounds, as this was deemed a manageable number that I could handle; hence, this gave a maximum of 50 compounds per hypothesis.

For hypotheses (a) and (b), if the number of compounds exceeded 50 for either hypothesis and for each reagent database, then the Morgan Fingerprint with radius 2, using all defaults and *nbits* = 1024, was calculated for each compound using RDKit and the RDKit's Diversity Picker KNIME node was used to select a diverse set of 50 compounds.

For the two other hypotheses, (c) top Pareto and diverse, and (d) worst Pareto and diverse, diversity in PLIFs was obtained by clustering the poses into 50 groups using the *k*-means KNIME node, with $k = 50$ according to their PLIFs. For hypothesis (c), the best Pareto ranked pose was kept from each cluster, while for hypothesis (d), the control, the worst Pareto ranked pose was kept. For the worst Pareto ranked, low cost was also desired.

The Pareto score for each pose was calculated using the following criteria: (i) high ligand efficiency *i.e.* the AutoDock Vina predicted affinity divided by the number of heavy atoms in the ligand; (ii) high TvPLIF similarity to the PLIF of the parent fragment-hit x1237; (iii) high TvPLIF to the PLIFs of all other 1st and 2nd generation fragment-hits (Supplementary Figure A.3), calculated as a sum of all TvPLIFs, and; (iv) low cost (for the MolPort reagents). The rationale behind the second and third objectives in the Pareto rank, (ii) and (iii), was to favour candidates that could recapitulate protein-ligand interactions seen in the previous fragment screening campaigns (Supplementary Figure A.3). As the interactions have been previously seen, they could be important for binding.

Lastly, duplicates were removed from the final list of compounds and the number of compounds chosen for each hypothesis and from each reagent database is summarised in Table 2-2. From the PEB reagents, 70 amides were selected, while from the MolPort database, 103 amides were chosen. I used Enamine as the compound supplier, and out of the 103 MolPort compounds, 48 were available for purchase from Enamine but only 42 of these were purchased, as the remaining 6 were too expensive. Despite the price filter I applied in the workflow, these expensive compounds emerged as the prices were quoted for MolPort and not Enamine and also there was variation in compound prices between the time the MolPort database was downloaded to when the compounds were

ordered. From the PEB reagents, 63 out of the 70 reagents were found, while the rest were deprecated. This resulted in 105 unique candidate amides that were to be made (Appendix A.2).

Reagent Database	Hypothesis				Total
	73 polar / H-bond	Greasy pocket	Best Pareto and diverse	Worst Pareto and diverse	
MolPort	15	33	34	21	103
Enamine (subset of MolPort)	10	14	15	9	48 (42 purchased)
PEB Reagents	4	40	6	20	70 (63 obtained)

Table 2-2. Summary of the follow-up compounds of x1237 chosen for synthesis. The compounds are split by hypothesis and where the acyl chloride/carboxylic acid reagent originates. A total of 118 compounds (48 Enamine and 70 PEB compounds) were chosen. Out of these, 42 of the Enamine compounds were purchased and 63 PEB reagents were obtained which resulted in 105 candidate compounds that were to be made.

2.3.2 Synthesis of the Follow-Up Compounds

I synthesised the 105 follow-up amides using the Opentrons OT-One liquid-handling robot (Ma et al., 2016; Densmore et al., 2017). The Opentrons machine has protocols for controlling its operation that are coded in Python. The machine was originally designed to automate biological protocols, but the Python API allows the user to build customised scripts so I tailored mine to my chemistry reaction protocols (see Section 2.2.8 and Appendix A.1). It also enabled the rapid set up of reactions that could be completed in one afternoon.

The 105 amides were the products of reacting 80 acids and 25 acyl chlorides (see Appendix A.2 Table A-1 and Table A-2). A slightly different reaction protocol and reaction condition was used for the carboxylic acids compared to the acyl chlorides; for the former I used the coupling reaction protocol, and for the latter the acylation protocol was used (see Section 2.2.8 for reaction protocols). All reaction mixtures were left

overnight at 35°C before aliquots were taken for liquid chromatography mass spectrometry (LCMS) analysis to assess the completion of the reactions.

LCMS is an analytical technique to identify chemical components within a mixture. During liquid chromatography, samples are dissolved in a mobile phase which is then passed over a solid phase. The different components in the sample have varying levels of interaction between the solid and liquid phase; hence, their retention times will differ. After liquid chromatography, the separated components can be analysed by mass spectrometry, which measures the mass-to-charge ratio of ions in the injected ionised sample.

The LCMS results indicated that the acylation reactions appeared to work better *i.e.* go to completion, than the coupling reactions; the amide product could be clearly seen as the major peak with no visible peak for the starting material for only 20 LCMS traces (25%) of the coupling reactions; whereas this was true for 20 LCMS traces (80%) of the acylation reactions. Starting materials were clearly seen in the LCMS traces of 28 coupling reactions (35%), whereas this was true for just one LCMS trace of the acylation reactions. For the carboxylic acids, products were not seen in the LCMS trace for 8 reactions (10%) and for the acyl chlorides, products were not seen in the LCMS trace for 2 reactions (8%).

Based on the quality of their LCMS traces, a total of 78 crude (unpurified) reaction mixtures were chosen, to be soaked into crystals of NUDT7. These 78 crude reaction mixtures comprised 23 acylation crudes and 55 coupling crudes. It is worth noting that soaking of crude reaction mixtures in contrast to purified products, introduced another factor of uncertainty in the results. The choice of soaking with crudes was due to ongoing research at Diamond Light Source that was investigating the effect of soaking

crude reaction mixtures as opposed to purified product into protein crystals. Provided with more time, I would repeat this experiment by soaking the crystals with purified amide products.

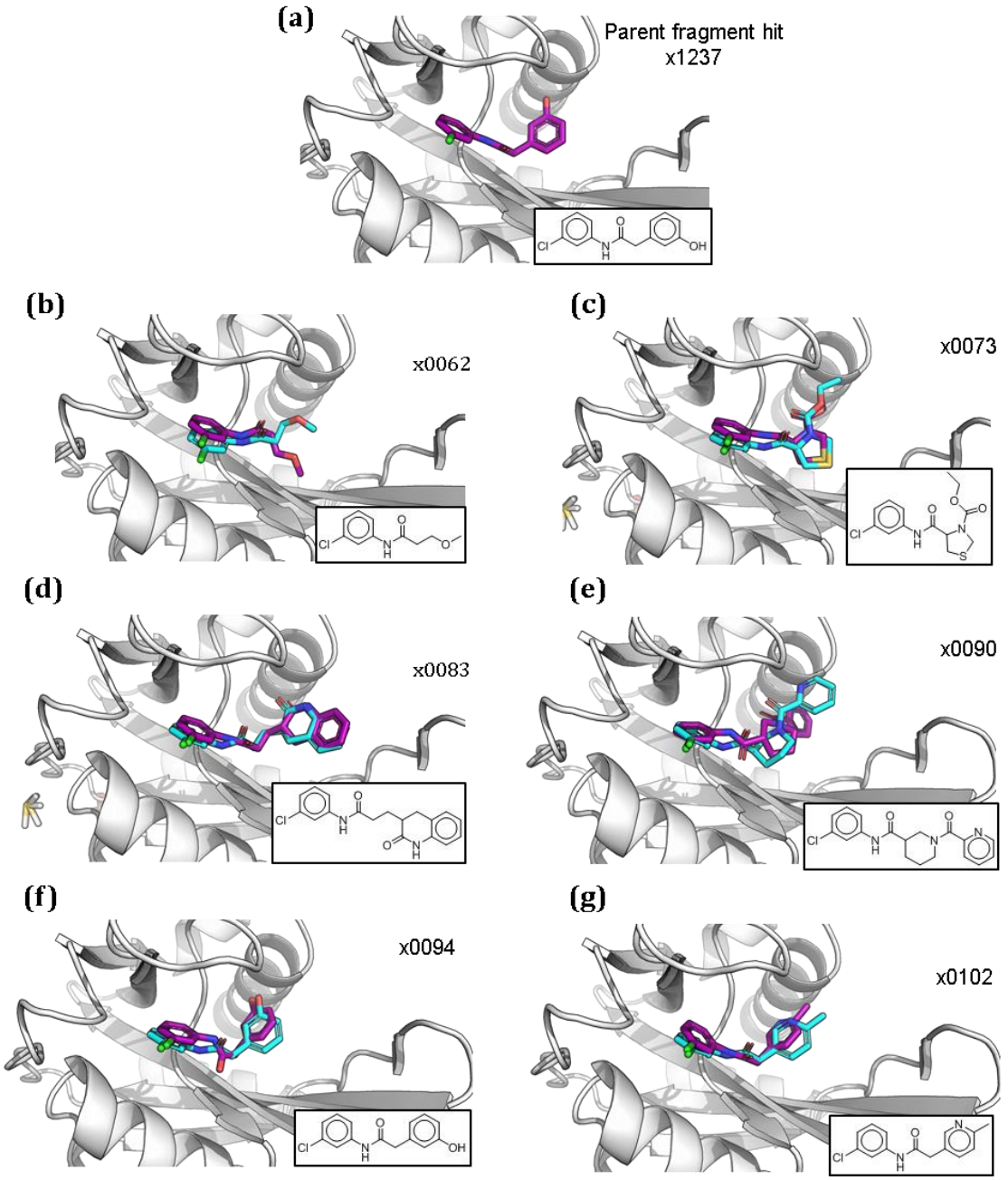


Figure 2.7. 78 crude reaction mixtures were soaked into NUDT7 crystals which resulted in 6 crystal structures. (b) – (g) All 6 structures show a very similar binding mode with respect to their shared 3-chloroaniline substructure. For each product, the X-ray crystal pose of the amide hit is shown in purple sticks, while the predicted docked pose is shown in cyan sticks. The NUDT7 secondary structure is shown in white cartoon representation. The hit IDs and 2D structures of the hits are also shown in the corner of each panel. Note that compounds shown in (a) and (f) are identical, *i.e.* the original hit, x1237 in (a), and the product of a carboxylic acid, x0094 in (f); which is a reassuring control. This figure was produced using PyMOL (Schrödinger, LLC.).

	Crystal Name	Resolution / Å	Reagent	Molecular weight of amide product	LCMS trace comment	Hypothesis	Crystal pose to docking pose RMSD / Å
(a)	†**x1237**	1.65	-	-	-	-	-
(b)	**x0062**	1.86	AcCl	214	Good LCMS	Best Pareto and diverse	1.6
(c)	**x0073**	2.01	AcOH	315	Starting material and product seen.	73 polar / H-bond	1.2
(d)	**x0083**	2.06	AcOH	329	Starting material and product seen.	Greasy pocket	0.53
(e)	**x0090**	2.33	AcOH	344	Starting material and product seen.	Worst Pareto and diverse	2.4
(f)	***x0094**	2.46	AcOH	262	Good LCMS	Greasy pocket	1.0 (1.0 from x1237)
(g)	**x0102**	2.00	AcOH	261	Starting material and product seen.	Greasy pocket	0.93

†x1237 is the parent hit
*x0094 is the same compound as the parent hit x1237

Table 2-3. Summary of the results from the crude reaction screen, showing the resolution of each X-ray crystal structure, the acyl reagent for which the hit came from, the molecular weight of the amide product, the quality of the LCMS trace, the hypothesis of why the candidate was chosen and the RMSD between the crystal and docked pose. It is interesting to note that the original hit was selected by the most successful hypothesis, 'greasy pocket'. The letters (b)-(g) correspond to the structures shown in Figure 2.7.

The soaking was performed following the procedure described in Section 2.2.9. Following data collection and structure solution (see Section 2.2.10), 6 X-ray crystal structures were identified to contain amide fragment hits, with resolutions ranging from 1.86 Å to 2.46 Å. It should be noted that one of these (x0094) is the same as the parent fragment x1237 (Figure 2.7) – a reassuring control result.

These 6 amide product hits show a very similar binding mode of the 3-chloroaniline moiety which agrees with the binding mode of the original parent hit, x1237, and also with those of the previous fragment screen (see Figure 2.2a). Amide product x0094, which is the same compound as the parent fragment hit x1237, has an RMSD of 1.0 Å from the pose of the parent fragment hit x1237; this is reasonable given that the resolution of the structures are 2.46 Å and 1.65 Å. All but x0090 had successful predicted docking poses, according to the criterion that RMSD should be less than 2 Å with respect to the crystal pose. One reason for the unsuccessful pose prediction of

x0090 could be due to the movement of the side chains in the binding site, in particular MET192, which was not modelled in protein-ligand docking (Appendix Figure A.2).

Starting materials were clearly seen in the LCMS traces of the crude reaction mixture of amide products x0073, x0083, x0090, and x0102 (but only very low amount was seen; see Appendix A.3 for LCMS traces), yet the amide product was observed in the X-ray structures. This contrasts to the starting material observed for three structures (discussed below and shown in Figure 2.8 and Table 2-4). Not much research has been done into the effect of soaking of crude reaction mixtures into protein crystal structures versus purified compound (Renaud et al., 2016). Previous research by Murray *et al.* has focused on the effect of crude reaction mixtures on the dissociation rate constant k_d (or off-rate k_{off}). The biophysical screen was not protein crystallography but surface plasmon resonance (SPR) and they reported that the dissociation rate constant, k_d, for a ligand binding to a protein can be accurately determined from crude unpurified reaction mixtures (Murray et al., 2014). Ongoing research at Diamond Light Source is investigating the effect of soaking crude reaction mixtures versus purified product into protein crystals, but preliminary results suggest that like SPR, the data from X-ray crystallographic screening is mostly unaffected by the use of crude reaction mixture versus purified product.

These six hits come from all four hypotheses, with the most (three products) being found by the "greasy pocket" hypothesis, which also corresponded to the hypothesis of the original fragment hit x1237. They all show novel interactions and exploration of the binding pocket. Compound x0090 is interesting as its structure is larger than the previous fragment-hits; the compound is comprised of three rings and thus explores this NUDT7 pocket more extensively than the fragment hits seen previously.

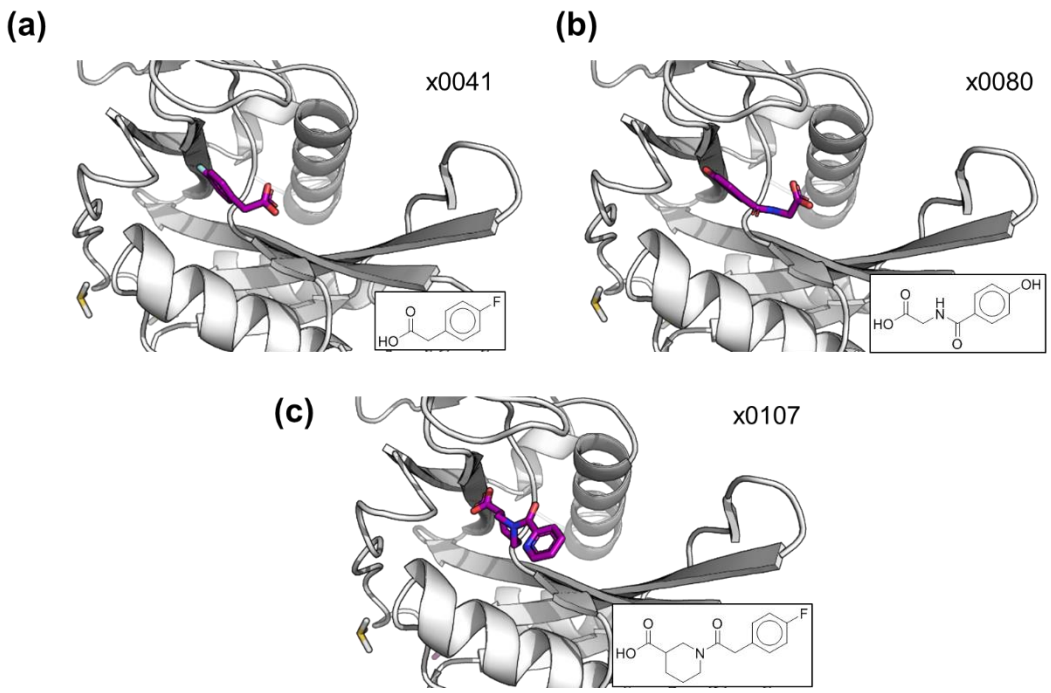


Figure 2.8. (a) – (c) Starting materials were seen in three X-ray crystal structures. This figure was produced using PyMOL (Schrödinger, LLC.).

	Crystal Name	Resolution / Å	Reagent	LCMS trace comment
(a)	**x0041**	2.14	AcCl	Good LCMS
(b)	**x0080**	1.89	AcOH	Starting material and product seen.
(c)	**x0107**	2.08	AcOH	Starting material and product seen.

Table 2-4. Details of the structures containing starting materials shown in Figure 2.8.

Interestingly, starting material was seen for three structures (Figure 2.8). For x0041, the reagent was an acyl chloride but due to its hydrolysis, the corresponding carboxylic acid was seen in the X-ray crystal structure (Figure 2.8a). Also, no conservation of binding mode with respect to the parent fragment-hit was observed for these three structures. For all three, the corresponding LCMS traces show little or no starting material (Appendix A.4). This contrasts to the six amide product X-ray crystal structures where starting material was seen in four of the six LCMS traces of the corresponding crudes discussed previously. From these results, it is still inconclusive how soaking with crude reaction mixture affects the ligand bound crystal structure seen *i.e.* it is possible that if these crude mixtures had been purified, the corresponding product may have been seen in the NUDT7 crystal structures. Moreover, there could be false negatives amongst the

rest of the crude reaction mixtures, hence, given more time, this experiment should be performed with purified products, to eliminate this uncertainty regarding soaking with crudes.

2.4 Conclusions

In this chapter, I have presented results from a prospective study that used my fragment elaboration protocol to propose follow-up compounds to an X-ray crystallographic fragment hit of Human peroxisomal coenzyme A diphosphatase NUDT7. Poised reactions were used to enumerate and generate candidate compounds, specifically amides, which were docked to generate theoretical poses for each. Filters based on computed molecular properties were used to eliminate incompatible functional groups, including rules such as the number of rotatable bonds and cLogP. Candidate compounds were selected using four hypotheses and how much structural information could be gained from each – using the results of docking, two hypotheses were based on predicted interactions made within the binding site, the third hypothesis aimed to pick candidates that made a diverse set of interactions within the binding pocket, but were also highly ranked in terms of ligand efficiency, and that made interactions that had been seen in previous fragment screens of NUDT7. The final hypothesis acted as a control to the third hypothesis, by picking the worst ranked in terms of ligand efficiency and interactions already seen.

Ultimately, 105 amide crude reactions were performed using semi-automated synthesis with a programmable liquid handler, and LCMS was used to assess the quality of the reactions. Of these, 78 of the crude reaction mixtures were chosen to be soaked into crystals of NUDT7. Subsequent X-ray crystallography found six crystal structures with

the product amide bound to NUDT7: five of these are novel hits and the other was the same as the parent fragment hit x1237, which represented a reassuring control result.

This represents a hit-rate of 8% (6/78) which is comparable to the hit rate reported by Hartshorn *et al.* who observed a hit-rate of 0.5 to 10% when they investigated the screening of fragment libraries using high throughput X-ray crystallography on five different proteins (Hartshorn et al., 2005). However, one should be cautious in comparing the hit-rate I obtained versus their reported hit-rate, as the situations are not equivalent. For example, this experiment was not the first fragment screening campaign for NUDT7, and the library I designed was based on an established binding chemotype; hence, one may expect the hit-rate to be higher than the typical 0.5-10%. On the other hand, the soaking was performed with crudes and not purified compounds; hence, this may have had a detrimental effect on the hit-rate.

The binding mode of the common substructure of these six hits is very similar to the original parent fragment, and the predicted docked binding pose was accurate apart from one. Starting material was seen in three crystal structures and hence it remains unclear what the effects of soaking crude reaction mixture versus purified product is on the result of a fragment screen. Given more time, this experiment should be repeated with purified products to eliminate the uncertainty associated with using crude mixtures *i.e.* my results could contain possible false negatives, where product would have been observed if purified product was used to soak.

In terms of how much information was gained from the four hypotheses, the experiment is inconclusive. Each of the five new fragment hits provided new information about potential areas to grow within the binding pocket. It is possible that more information could have been gained from just using the third hypothesis to pick compounds that

make a diverse set of interactions within the binding site. Further experiments will help to understand the best way of prioritising compounds to make.

It is also worth noting that no repeat experiments were performed. Operation of the Opentrons was not perfect and often there were errors in the liquid handling. For example, if a liquid was too viscous then bubbles may have formed within the pipette tip, so an inaccurate volume might be transferred. Errors were also reported by the LabCyte Echo liquid handler during the transfer of crude reaction mixtures to the NUDT7 crystals, so for some wells the protein crystal would not have been soaked with compound – this could have resulted in a false negative result, where a hit X-ray crystal structure could have been seen if the protein had been correctly soaked. Furthermore, there were some failures in the mounting of crystals. For all of these listed errors, multiple repeats of the experiment, multiple repeats of each reaction within a single experiment, and multiple soaking and mounting of the same crude reaction would overcome this and help quantify the reproducibility of this protocol.

This prospective study used RMSD to measure the conservation of binding mode between the docked poses of proposed follow-up candidates and the parent fragment hit x1237. However, through inspection of some of the results, I recognised that RMSD may be inappropriate to use when either molecule is pseudosymmetric; hence, this provided motivation for the next chapter, where I explored an open-source alternative to RMSD for measuring the conservation of binding modes between fragments and their elaborated counterparts.

Chapter 3 Comparison of Similarity of Binding Mode Measures for Elaborated Fragments

The studies described in this chapter for the comparison of similarity of binding mode measures for elaborated fragments has been published in the preprint server ChemRxiv (Leung et al., 2019).

In this chapter I have implemented SuCOS, an open source alternative to a combined overlap score, COS, which was devised by Malhotra and Karanicolas to measure the conservation of binding mode of pairs of smaller and larger ligands bound to the same protein. The advantage of SuCOS over COS is that it is implemented using the open source toolkit RDKit whereas COS uses the commercial software ROCS. To my knowledge, no previous study has directly compared three measures – RMSD, PLIF similarity and combined shape and chemical feature overlap – for the quantification of conservation of binding mode for 3D structures of fragments and their elaborated counterparts. Hence, the studies presented in this chapter looks at the strengths and weaknesses of RMSD, PLIF similarity, and SuCOS on a dataset of X-ray crystal structures of paired elaborated larger and smaller molecules bound to the same protein.

3.1 Introduction

In the elaboration of a fragment hit, it is assumed that the binding mode of the original fragment hit in the elaborated fragment molecule will be structurally conserved. Malhotra and Karanicolas assembled a dataset of 297 ligand pairs from the PDBbind database to validate this hypothesis, and found that it was true in 86% of the cases (Malhotra and Karanicolas, 2017). They quantified the degree of binding mode conservation by devising a combined overlap score, COS, that considers steric overlap and the overlap of chemical features such as hydrogen bond donors and acceptors. COS is an asymmetric metric, as it considers the proportion of overlap of the query molecule with the reference molecule, but not the other way around. COS ranges from 0, where there is no overlap of the elaborated molecule's volume and chemical features onto the non-elaborated counterpart's, to 1, where there is complete overlap. Malhotra and Karanicolas used a COS cutoff that ranged from 0.4 to 0.55, depending on the chemical substructure match between the reference and the query: if a pair had a COS score less than this, the pair was deemed not to have a conserved binding mode.

Drwal *et al.* published a related analysis that investigated the similarity of binding modes for fragments compared to even smaller crystallization additives (Drwal et al., 2017). They used shape and chemical feature overlap to determine pose similarity, but Protein Ligand Interaction Fingerprints (PLIFs) to determine binding mode similarity. A PLIF similarity threshold of 0.6 or greater was deemed to be a conserved binding mode.

Numerous studies have also shown that the use of binding mode information can lead to greater docking success by selecting the correct binding pose (Marcou and Rognan, 2007; Verdonk et al., 2016; Anighoro and Bajorath, 2016b; Desaphy et al., 2013; Fu

and Meiler, 2018; Jain, 2003; Nicholls et al., 2010). Marcou and Rognan used PLIFs to rescore docking poses and picked poses based on those with the highest PLIF similarity to the PLIF of the reference protein-ligand complex (Marcou and Rognan, 2007). Shape similarity has also been shown to be a successful metric in prioritizing the correct pose (Anighoro and Bajorath, 2016b; Kumar and Zhang, 2016b).

Despite the existence of these metrics, a root mean square deviation, or RMSD, cutoff of 2.0 Å is still the most widely used measure for assessing whether or not a docking is successful (Onodera et al., 2007; Plewczynski et al., 2011; Warren et al., 2006). The RMSD is calculated by measuring the positional deviation, or distance, between equivalent atoms in the reference and query molecules. In fragment-based elaboration, the reference is the fragment hit, and the query is the elaborated fragment. Exactly which atoms are used in the RMSD calculation can vary: sometimes it is a simplistic one-to-one mapping of atoms in the compared molecules, and sometimes it takes chemical symmetry into account, such as the 3-fold rotational equivalence of tertiary-butyls (Allen and Rizzo, 2014).

If RMSD is calculated between dissimilar molecules, such as fragments and their elaborated counterparts, the corresponding atom-to-atom mapping must be determined. This is commonly done by computing the maximum common substructure, or MCS (Fu and Meiler, 2018; Drwal et al., 2018). Several MCS algorithms exist each differing in their approach (Englert and Kovács, 2015; Kawabata, 2011; Raymond et al., 2002; Duesbury et al., 2018). For example, distinctions can be made between connected and disconnected MCSs, where the latter can contain multiple substructures. For the measurement of RMSD and defining the 3D atom-to-atom mapping, it is not clear which is the best approach to use.

Sometimes, a strict match of atoms does not exist, but it is still reasonable to map ring atoms in one molecule to non-ring atoms in another, or atoms in aliphatic rings to those in aromatic rings. Further complications arise when MCS is used for virtual screening libraries of fragment follow-ups, as they can include bioisosteres, pseudosymmetric small molecules and small changes in the chemical scaffold of that fragment, thus reducing the MCS to less than the original fragment hit.

Although the term pseudosymmetry has been previously used in the context of molecules (Reddy et al., 2007; Malhotra and Karanicolas, 2017), to the best of my knowledge there is currently no formal definition. Therefore for clarity, I refer to pseudosymmetry in a molecule when the molecule resembles having symmetry but due to small differences between two substructure groups it is not symmetric *e.g.* atom insertion or atom replacement.

Although manual inspection of compound overlap by experienced structure-based compound designers is effective in qualitatively evaluating compound overlay, these confounding factors make algorithmic comparison of $10^3 – 10^7$ molecules challenging. Nevertheless, there are several examples of where RMSD has been used for comparing non-identical molecules such as in virtual screening programs (Bian et al., 2017; Zaliani et al., 2009; Durrant et al., 2009; Murray et al., 2012; Ichihara et al., 2014).

3.1.1 Prior State of the Art and Chapter Aims

Currently it is unclear what metric is best at quantifying the degree of binding mode conservation, especially in cases where a smaller ligand and its elaborated counterpart are compared. Previous studies have discussed the pitfalls of certain metrics and hence use multiple metrics alongside one another (Temml et al., 2014). To the best of my

knowledge, no study has focused on the direct comparison of these metrics to quantify conservation of binding mode of elaborated molecules and their non-elaborated counterparts. I utilised the Malhotra and Karanicolas ligand pair dataset to investigate three metrics: RMSD, PLIF similarity, and a new metric called SuCOS, a combined shape-chemical feature-based metric I have developed. From this study, I aimed to assess which is the best method for binding mode comparison of elaborated molecules by looking at the strengths and weaknesses of each metric. In this chapter, I differentiate between the calculation of the RMSD between identical molecules, and the RMSD between elaborated molecules with their non-elaborated counterparts. For the former, I shall use the notation 'All-RMSD' to indicate that all atoms in the reference and query molecules were used *i.e.* in Part II for comparing poses of identical molecules; while for the latter, I use the notation 'MCS-RMSD' to indicate that an MCS was first identified in both molecules to define the corresponding pairs of atoms for the RMSD calculation *i.e.* in Part I and Part III for comparing poses of smaller and larger molecules. Furthermore, I have used two measures for PLIF similarity – Tversky and Tanimoto which are denoted TvPLIF and TnPLIF respectively.

3.2 Methods

In this chapter, I used the dataset curated by Malhotra and Karanicolas, hereafter referred to as the MK dataset, that consists of 297 ligand pairs from the PDBbind database, where each pair consisted of a smaller and larger ligand solved in complex with the same protein partner (Malhotra and Karanicolas, 2017). The larger ligand could have, but not necessarily, arisen through synthetic elaborations of the smaller ligand.

Notation	Description
L_S^X	Crystal structure of smaller ligand
L_L^X	Crystal structure of larger ligand
L_L^A	Aligned structure of larger ligand, after aligning larger ligand's cognate protein onto the protein from the smaller ligand's complex
P_S^X	Protein in the smaller ligand's crystal structure
P_L^X	Protein in the larger ligand's crystal structure
$P_SL_S^X$	Protein-ligand complex for the smaller ligand's crystal structure
$P_LL_L^X$	Protein-ligand complex for the larger ligand's crystal structure
$P_LL_L^A$	Larger ligand's protein complex that has been aligned onto the smaller ligand's protein complex
$P_SL_S^{D,i}$	The i^{th} docked pose of the smaller ligand into the P_S^X
$P_LL_L^{D,i}$	The i^{th} docked pose of the larger ligand into the P_L^X
Metric(L_S^X, L_X^A)	The metric between L_S^X and L_L^A; *e.g.* RMSD(L_S^X, L_X^A) is the RMSD between L_S^X and L_L^A.

Table 3-1. Notation used to describe the structures and metrics used in this chapter.

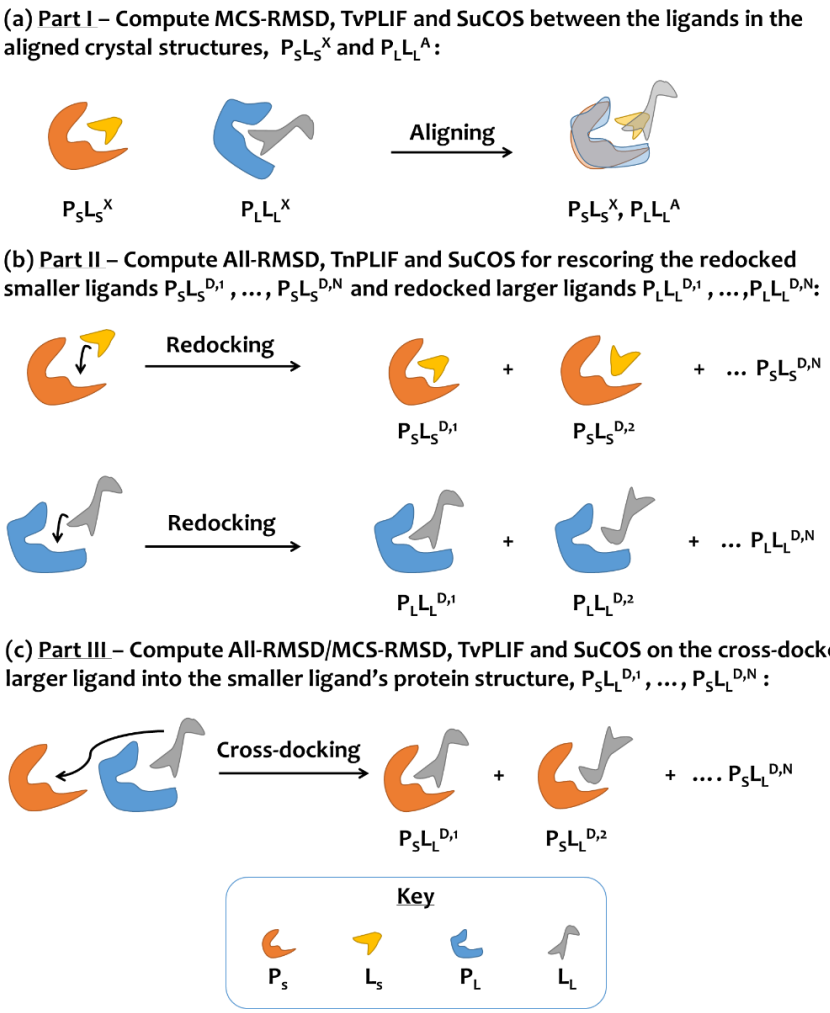


Figure 3.1. Overview of study performed in this chapter. (a) Computation of metrics, MCS-RMSD, TvPLIF, and SuCOS, to score the conservation of binding mode for the ligands in the aligned crystal structures, $P_SL_S^X$ and $P_LL_L^A$, in the MK dataset. (b) Perform redocking studies to generate poses and score them with the metrics All-RMSD, TnPLIF and SuCOS. (c) Cross-docking the larger ligand, L_L, into the smaller ligand's protein crystal structure, P_S^X, to simulate a realistic virtual screening effort and assessing which metric performs best at picking out the correct pose. In this Chapter, studies (a), (b) and (c) correspond to Sections 3.3.1, 3.3.2 and 3.3.3 respectively.

Here, I perform a direct comparison of the three metrics on (i) Malhotra and Karanicolas' dataset of paired larger and smaller molecules bound to the same protein (Figure 3.1a); (ii) redocking each ligand to its respective protein (Figure 3.1b); and (iii) cross-docking of the larger molecule into the smaller molecule's cognate protein structure (Figure 3.1c). I use this dataset to simulate elaboration efforts and situations where virtual screening may have used binding pose similarity to compare virtual molecules to a fragment hit and decide which elaborated molecule to make next.

3.2.1 Downloading and Filtering of the Malhohtra and Karanicolas Ligand Pair Set

The PDB codes for the ligand pair dataset are available from the supporting information of the Malhotra and Karanicolas study of binding mode changing during chemical elaboration (Malhotra and Karanicolas, 2017). The dataset contains 297 ligand pairs. However, only 284 ligand pairs were used in this study. Thirteen pairs were excluded, of which 11 pairs have the ligand positioned in between the two chains in at least one of the PDB structures (3vp2/3voz, 3vp4/3uo9, 3voz/3uo9, 3deh/3dek, 3zsy/3zso, 3n7a/3n86, 3zsz/3zso, 3zt1/3zso, 3iaf/3iae, 2pqz/2qnn, 2y54/2y58). Deprecation of the PDB entry 3v0y and absence of 1yp9 from the PDB/UniProt mapping server (Martin, 2005) led to the exclusion of 3v0y/2qzr and 3ati/1yp9 respectively. The PDBs for the remaining ligand pairs were downloaded from the RCSB PDB Web site (Berman et al., 2000) as the biological assemblies.

3.2.2 Preparation of the Malhotra and Karanicolas Ligand Pairs

For each PDB structure, only the first chain that contains the small molecule was used. The first model was used for each structure and alternative states were removed using

the *removealt* function in PyMOL (Schrödinger, LLC.) (version 2.1.0). The larger ligand's crystal structure, $P_L L_L^X$, was aligned to the smaller ligand's crystal structure, $P_S L_S^X$, using PyMOL's *align* function using all atoms, which I refer to as:

$$P_S L_S^X, P_L L_L^A = align(P_S L_S^X,\ P_L L_L^X) \qquad (3.1)$$

where $P_L L_L^A$ denotes the larger ligand's structure after alignment. The ligands were chosen by the three letter code according to the Supplementary Information of the study by Malhotra and Karanicolas (Malhotra and Karanicolas, 2017). For each pair, $P_S L_S^X$ and $P_L L_L^A$, if multiple ligands exist in either structure then only one ligand was used for each. These ligand pairs were chosen by having the shortest distance between their centroids, computed by RDKit's *computecentroid* function. For example, if $P_S L_S^X$ has multiple ligands but $P_L L_L^A$ only has one, then the smaller ligand, L_S, with the smallest distance to the larger, L_L, is chosen. If multiple ligands are present in both, then all pairwise distances between all centroids were calculated and the two closest were chosen.

The smaller and larger PDB structures were checked for consistent numbering of residues between each ligand pair. Residues in pairs with inconsistent numbering were renumbered according to the UniProt numbering scheme using the UniProtKB/SwissProt database (Martin, 2005).

3.2.3 Protein Preparation for Docking

For the preparation of proteins for docking, the same procedure as described in Section 2.2.3, was performed.

3.2.4 Ligand Preparation

The ligands were downloaded without hydrogens in the SDF file format from the RCSB PDB Web site and their respective SMILES obtained from the datafield of the SDF file. The procedure as stated in Section 2.2.4 for SMILES standardisation and protonation was then applied.

For the conformer generation step, a similar procedure to the one described in the previous Chapter, Section 2.2.4, was used. Again, the RDKit *ConstrainedEmbed* function was used, but different 'core' molecules were used as the source of constraint in *ConstrainedEmbed*. For the redocking study (Section 3.3.2), L_S^X and L_L^X were used as the reference cores for the constrained conformer generation of L_S and L_L respectively. Hence only one conformer was generated which corresponds to the protonated crystal pose. For the cross-dockings of the L_L (Section 3.3.3), the reference core was the MCS to L_S^X which was identified by RDKit's *FindMCS* with *completeRingsOnly* and *ringMatchesRingOnly* set to *True*. Again, if there were no amides or aliphatic rings present in the molecule then only one conformer was generated, else ten conformers were generated and clustered, keeping only those which are greater than 0.5 Å from the lowest energy conformer. This was done to create diversity in conformers as the docking treats amides and aliphatic rings as rigid. The conformers were outputted as SDFs that were then converted to MOL2 format using Open Babel in order to retain bond order information. The ligand MOL2 file was then processed into the PDBQT using MGLTools script *prepare_ligand4.py* with all default parameters (Morris et al., 2009).

3.2.5 Docking

In this chapter, I performed two different types of docking – redocking (Figure 3.1b) and cross-docking (Figure 3.1c). Redocking is the process of taking a ligand from its structure and docking it back into the same structure. It is typically used before a structure-based virtual screening campaign to validate whether the docking method can successfully recapitulate the experimentally-observed binding pose, and if so, what the best parameters are.

Cross-docking is the process of taking a series of ligand-protein complexes and docking each ligand into every receptor. Cross-docking aims to address the conformational flexibility of the protein to improve docking performance by use of multiple protein structures. In this study, I refer to cross-docking to the specific case of docking the larger ligand into the smaller ligand's protein crystal structure.

The same docking procedure as stated in Section 2.2.5 was used for both redocking (Section 3.3.2) and cross-docking (Section 3.3.3) studies. For the redocking study, the center was chosen to be the centroid of the crystallographic ligand, either L_S^X or L_L^X depending on the redocking. For the cross-docking study, the box center was set to the centroid of the smaller crystallographic ligand, L_S^X.

Like in Section 2.2.5, for each docking run, AutoDock Vina outputs the docking poses in a PDBQT file which contains the AutoDock Vina predicted affinities (Aff^{Vina}). The PDBQT file was converted into a SDF file using Open Babel that was then used for RMSD and SuCOS calculation. For PLIF calculation, the PDBQT file was converted into a PDB file using Open Babel and combined with the either the original PDB of the smaller protein, P_S, or the larger protein, P_L, without the original ligand.

3.2.6 Calculation of RMSD

In this chapter, I differentiate RMSD calculations into All-RMSD and MCS-RMSD. All-RMSD refers to an RMSD calculation that uses all atoms in the reference and query structures *i.e.* for poses of identical molecules. MCS-RMSD refers to firstly identifying an MCS in both reference and query molecules and using the corresponding pairs of atoms for the RMSD calculation. Both All-RMSD and MCS-RMSD calculations were calculated using RDKit in Python (RDKit, Version 2018.03.1, 2018) and the procedure was described in the previous Chapter, Section 2.2.6.

In Section 3.3.1, MCS-RMSD was used between the X-ray poses of the smaller ligand, L_S^X, and the larger ligand, L_L^X. In Section 3.3.2, All-RMSD was used between the docked ligand, L_S^D or L_L^D, and its respectively crystal ligand pose, L_S^X or L_L^X. In Section 3.3.3, MCS-RMSD was used between the docked pose of the larger ligand, L_L^D, and the smaller ligand crystal pose, L_S^X, and All-RMSD was used between the docked pose of the larger ligand, L_L^D, and the larger ligand's crystal pose, L_L^X.

3.2.7 Calculation of Protein Ligand Interaction Fingerprints (PLIFs)

The method for PLIF calculation was described previously in the previous Chapter, Section 2.2.7. However, in this chapter, I used not only the Tversky index as the measure of similarity between two PLIFs but also the Tanimoto coefficient. They are denoted TvPLIF, for Tversky, and TnPLIF, for Tanimoto. Calculations of TvPLIF, like in the previous Chapter, were computed with weights $\alpha = 1$ and $\beta = 0$ corresponding to the smaller and larger ligand respectively (Section 3.3.1) or in the case of the cross-docking (Section 3.3.3), for the crystal pose, L_L^X, and docked pose, L_L^D, respectively. For

calculations of TnPLIF (Section 3.3.2), the RDKit function *DataStructs.TanimotoSimilarity* was used to calculate the Tanimoto coefficient.

3.2.8 Calculation of SuCOS

SuCOS is a metric inspired by Malhotra and Karanicolas' combined overlap score, COS (Malhotra and Karanicolas, 2017):

$$COS = 0.5\frac{O_{ls}}{O_{ss}} + 0.5\frac{C_{ls}}{C_{ss}} \quad (3.2)$$

where O_{ls} and O_{ss} represents the volume overlap of the larger ligand with the smaller ligand and the volume overlap of the smaller ligand with itself respectively. Similarly C_{ls} and C_{ss} represents the same but for 'color' overlap instead of volume overlap. The volume and color overlap are given equivalent weights. They used ROCS software (ROCS, version 3.2.0.3, 2015) to calculate these overlaps on the smaller and larger ligands in the pre-aligned crystal structures. As mentioned in the introduction (Section 1.2.3.4), the ROCS algorithm uses Gaussian spheres to represent the atoms or 'color' features and the intersection of the spheres represents the overlap between the two molecules. The color features, or chemical feature types, included hydrogen-bond acceptors, hydrogen-bond donors, anions, cations and aromatic rings. As mentioned in the introduction to this chapter, the values of COS range from 0 to 1, where a score of 0 means there is no overlap of the larger with the smaller ligand and a score of 1 means that the larger overlaps fully with the smaller in terms of volume and color overlap.

SuCOS is based on COS and is similarly a 3D overlap score that is composed of half chemical feature overlap and half shape overlap. It is an open source alternative to COS

and I do not claim it to be superior in any way. It utilises two RDKit functions, *ScoreFeats* for chemical feature overlap and *ShapeProtrudeDist* for shape overlap:

$$SuCOS = 0.5\ (ScoreFeats) + 0.5\ (1 - ShapeProtrudeDist) \quad (3.3)$$

The proportion of the reference molecule's volume that is covered by the query molecule is calculated by *(1 – ShapeProtrudeDist)* with the option *allowReordering* set to *False*. This is an asymmetric shape overlap metric, so if the query molecule completely covers the reference molecule in volume, then the score from *(1 – ShapeProtrudeDist)* will be 1 regardless of how much larger the reference molecule is compared to the query molecule. *ShapeProtrudeDist* computes the shape protrusion between the two molecules based on their predefined alignment and uses a grid to encode the molecular shapes. This contrasts with how COS calculates the shape overlap, which uses ROCS that represents the atoms as Gaussian spheres. The score from *(1 – ShapeProtrudeDist)* ranges from 0 (no volume overlap) to 1 (complete volume overlap).

SuCOS's chemical feature overlap score is calculated using all eight of RDKit's pharmacophoric feature types, *i.e.* : *Donor*, *Acceptor*, *NegIonizable*, *PosIonizable*, *ZnBinder*, *Aromatic*, *Hydrophobe*, and *LumpedHydrophobe*. For each pharmacophoric feature type, RDKit has a list of SMARTS patterns defined in the *BaseFeatures.fdef* file to identify the chemical features in a molecule. The *FeatMap* function creates a so-called "feature map" for the reference molecule; I used the default settings, with a Gaussian potential centred on each feature that uses a sigma factor of 1 Å, and only counts overlaps within 2.5 Å. The *ScoreFeats* function scores the query molecule's set of pharmacophoric features against the feature map of the reference molecule. The score is normalized by the number of features in the smaller molecule, making it an

asymmetric score. RDKit has three ways of scoring feature maps: *All*, *Closest* and *Best*. When scoring by *All*, every feature in the query molecule is compared with every matching feature in the reference feature map. When scoring by *Closest*, however, each feature in the query molecule is compared with only the nearest matching feature in the reference feature map. Finally, when scoring by *Best*, each feature in the query is compared with every feature in the reference feature map, and the highest score found is used. In this work, I used the *All* scoring mode. This can occasionally output scores greater than 1; therefore, the outputted feature map score is clamped to a maximum of 1, since the region of interest is the classification boundary and not at this upper end of the range. It should be noted that SuCOS can also use the *Closest* and *Best* modes; however, only the *All* mode was used.

As both the shape and chemical feature components of SuCOS are asymmetric, SuCOS is thus asymmetric, *i.e.* it depends only on how well the query overlaps with the reference, and, unlike RMSD, is independent of any size differences between the two.

There are several differences between SuCOS and COS. For example, their lists of chemical feature types differ and also how the shape and chemical features are encoded. With regards to the different feature types, Malhotra and Karanicolas describe using five 'color' feature types, whereas SuCOS uses eight of RDKit's pharmacophoric feature types. Moreover, the substructures that are associated with each chemical feature are different. Another difference already mentioned is the way RDKit's *ShapeProtrudeDist* encodes the 3D molecule using a grid with a default spacing of 0.5 Å, whereas ROCS shape uses Gaussian spheres. Both RDKit's FeatMap and ROCS's color uses Gaussian spheres to represent the chemical features but whilst the default parameters for RDKit are clear (Gaussian σ=1 and cutoff radius=2.5 Å) the corresponding parameters that were used in COS are not clear. Thus these differences

between SuCOS and COS can account for the differences later observed (Section 3.3.1, Figure 3.3).

In Section 3.3.1, I compare the structure of each larger ligand (reference) to that of its corresponding smaller ligand (query). In Section 3.3.2 and Section 3.3.3, I compare a docked pose to a reference crystal pose, and for this, the reference molecule and query molecule was the crystal pose and docking pose respectively.

3.3 Results and Discussion

3.3.1 Part I: Comparison of MCS-RMSD, TvPLIF and SuCOS Between the Ligands in the Aligned Crystal Structures of the Malhotra and Karanicolas Ligand Pair Set

For the studies performed in this chapter, I used a dataset of 284 ligand pairs, which is a subset of the 297 ligand pairs that Malhotra and Karanicolas published (see Section 3.2.1 for details). The set of 284 ligand pairs comprises of 485 unique PDB IDs and the distribution of their resolutions are shown in Figure 3.2. 11% (51/485) of the structures have a resolution over 2.5 Å and < 1% (4/485) have a resolution over 3 Å. For these poor resolution structures, there is a larger uncertainty in the exact geometry of the ligand in the electron density; however, the studies presented in this chapter are concerned with the comparison of three different measures for quantifying binding mode similarity, where each comparison uses the same ligand pose, hence inclusion of these poor resolution structures should not affect the results. Moreover, in the previous study from Malhotra and Karanicolas', they were concerned with measuring how often elaborated ligands change their binding modes and the reason when they did change. In

their study, they considered factors such as poor resolution and poor R_{free} and concluded that exclusion of such pairs did not change their results (*i.e.* those with higher R_{free} values were not found to have a statistically significant higher frequency of changed binding mode).

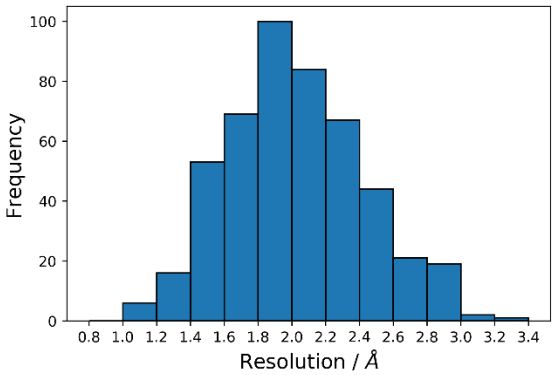


Figure 3.2. The distribution of the resolutions of the 485 PDB structures used in Chapter 3.

Malhotra and Karanicolas evaluated the ligand overlap of the smaller and larger ligands by computing the COS score. In this section (Part I), I calculated the MCS-RMSD and TvPLIF for each pair of this dataset, in order to compare these against Malhotra and Karanicolas' COS values (Figure 3.1a). For each pair, I also calculated values using SuCOS, my open source alternative to COS, which is introduced below.

The COS metric uses commercial software ROCS (version 3.2.0.3) to compute the overlap of volume and chemical features. As mentioned in Section 3.2.8, the volume overlap is the intersection volume of the two molecules, normalised by the smaller molecule. Chemical feature overlap refers to the spatial overlap of pharmacophoric features such as hydrogen bond donors, hydrogen acceptors, aromatic groups *etc.* and likewise it is computed by taking the intersection of the chemical features of the two molecules and normalizing by the chemical features of the smaller.

I devised an open-source alternative to COS, namely SuCOS, which uses RDKit functions *ShapeProtrudeDist* and *ScoreFeats* to compute the volume and chemical feature overlap respectively (see Section 3.2.8). I compared the performance of SuCOS to COS by computing SuCOS for each pair of the filtered Malhotra and Karanicolas ligand pair set and compared it to the corresponding COS score. SuCOS achieved a good correlation with COS, having a Pearson correlation coefficient, $R_P = 0.93$, Spearman correlation coefficient, $\rho = 0.94$ and Kendall's tau, $\tau = 0.79$ (Figure 3.3a).

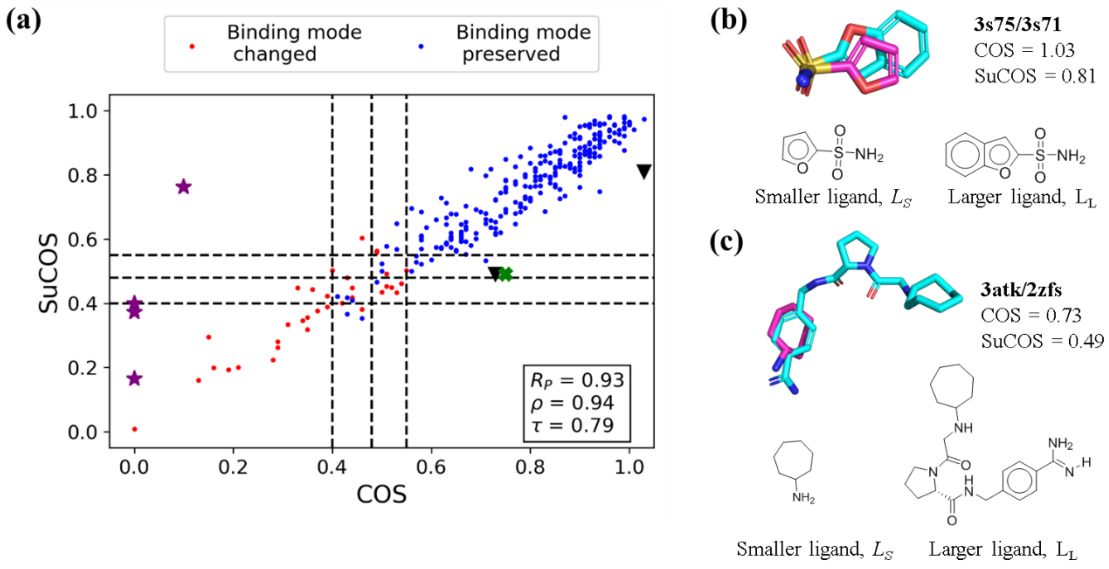


Figure 3.3 SuCOS is a good open-source alternative to the COS metric used by Malhotra and Karanicolas. (a) SuCOS was calculated for every ligand pair in the filtered Malhotra and Karanicolas set. The COS values for each ligand pair were obtained from the Supporting Information of the Malhotra and Karanicolas study. Good correlation (Pearson correlation coefficient, $R_P = 0.93$, Spearman correlation coefficient, $\rho = 0.94$ and Kendall's tau, $\tau = 0.79$) shows that SuCOS is an excellent open-source alternative and one that I shall use in this study. Dotted lines show the three different COS cutoff levels Malhotra and Karanicolas used to define the conservation of binding mode. The corresponding three cutoff levels are also shown on the SuCOS axis. As defined by Malhotra and Karanicolas, the conserved binding modes are shown by the blue points, while the unconserved modes are shown by the red points. The four outliers in purple stars and green cross are discussed in the text. The two black triangles represent pairs 3s75/3s71 and 3atk/2zfs, which are the pairs with largest differences in COS and SuCOS values. These are also discussed in the text and are shown in (b) and (c) respectively, where the smaller and larger ligand is shown with magenta and cyan carbons respectively.

The vertical dotted lines show the three different COS cutoffs Malhotra and Karanicolas used to determine whether an elaborated molecule had changed binding mode. The different cutoffs accounted for pairs with differing chemical substructure scores. Upon inspection of the outliers, four of the points have low COS but high SuCOS (PDB IDs

of smaller/larger pairs: 2hdq/1l2s, 3adt/3ads, 4e49/3f8e and 3adu/3ads, shown by the purple stars in Figure 3.3a); all these structures have multiple smaller and/or larger ligands bound to the protein and it is possible that COS was computed for the one with poorer overlap. It is also possible that another ligand pair (PDB IDs: 1o6i/1w1p) has the smaller and larger ligand swapped (shown by green cross in Figure 3.3a) as the reported volume overlap score is 1.06 but the ligand corresponding to 1o6i is the larger of the two.

As both values for COS and SuCOS rely on the ligands being pre-aligned, the slight differences in values may be attributed to the different methods used to align the protein pairs: Malhotra and Karinicolas used *TM-align* (Zhang and Skolnick, 2005) to align the pairs of smaller and larger complexes, whereas I used PyMOL's *align* function (see Methods Section 3.2.2). To investigate this, I used *TM-align* to align the pairs of the filtered Malhotra and Karanicolas ligand pair set and recalculated the SuCOS values; however, a very similar correlation was obtained when compared to the corresponding COS values (change in R_P, ρ and $\tau < 0.01$, see Figure 3.4). Therefore, differences between COS and SuCOS values must be predominantly due to algorithm differences in how shape and chemical feature overlap is calculated by the RDKit functions in comparison to the ROCS implementation which in turn may explain the two ligand pairs with the largest differences in COS and SuCOS (Figure 3.3b and Figure 3.3c). From here on, I used SuCOS to calculate the combined shape and chemical feature overlap.

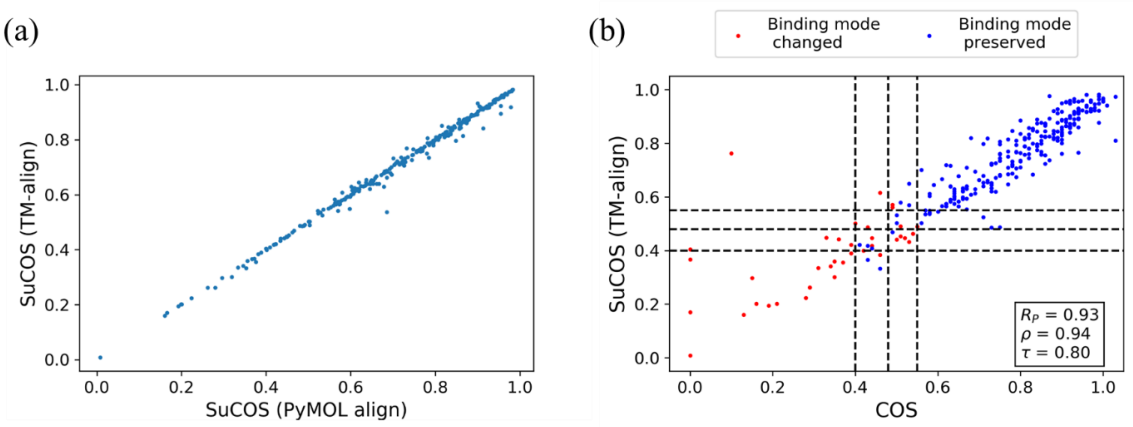


Figure 3.4. (a) Comparing the differences in SuCOS values when using *TM-align* to align the pairs of the filtered Malhotra and Karanicolas ligand pair set versus PyMOL's *align* function. (b) The SuCOS values from the TM-aligned ligands are plotted against their corresponding COS values obtained from the Supporting Information of the Malhotra and Karanicolas study. The difference in R_p, ρ and τ values are all less than 0.01 when compared to their corresponding values in the plot where the complexes were aligned using PyMOL's *align* function (Figure 3.3).

Next, to determine the correlation between MCS-RMSD and SuCOS, the two metrics were calculated on each of the ligand pairs. The scatter plot shows the expected trend that MCS-RMSD decreases with SuCOS score, with R_P = -0.66, ρ = -0.82 and τ = -0.63 (Figure 3.5a). The widely used 2 Å RMSD cutoff (Bursulaya et al., 2003) is also shown as a dotted line. Points located in the top-right and bottom-left can be regarded as false negatives (FNs) (high RMSD, high SuCOS) and false positives (FPs) (low MCS-RMSD, low SuCOS) respectively and the proportion of these points are 12% and 2% respectively. Although, the FNs and FPs imply that the metric on the x axis is taken as the ground truth, the use of these percentages were intended to highlight the cases where there are discrepancies between the two metrics and the relative proportion of these cases.

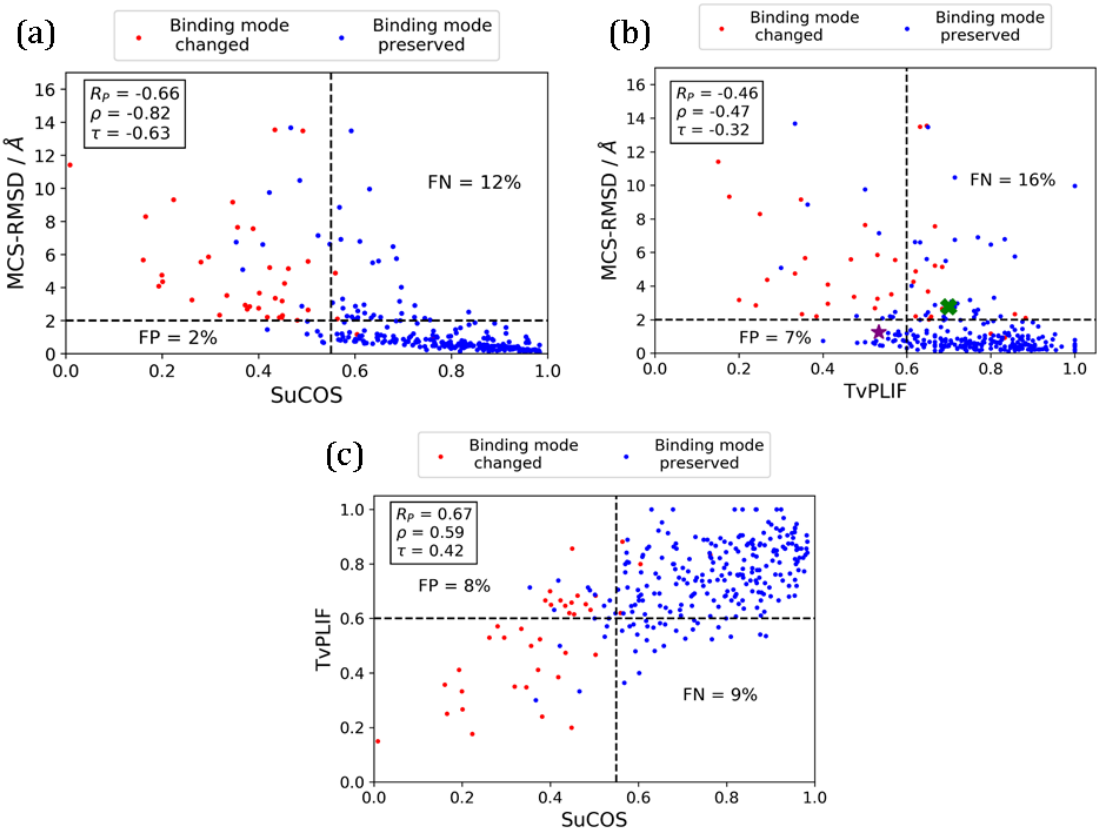
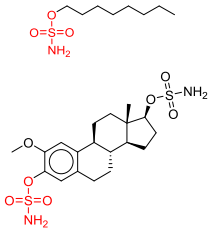
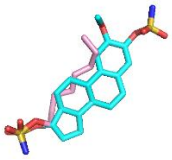
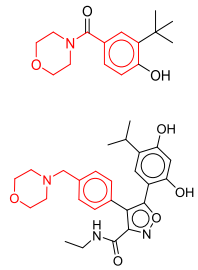
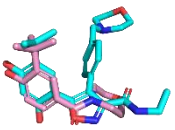
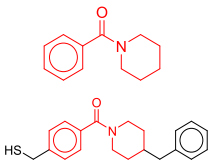
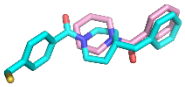
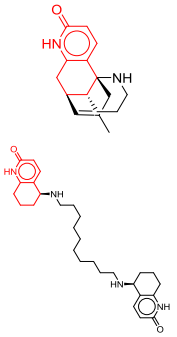
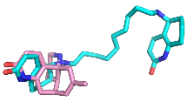
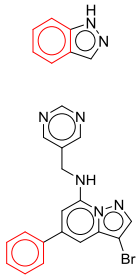
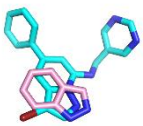


Figure 3.5. Comparison of three conservation of binding mode metrics on the Malhotra and Karanicolas ligand pair set. If I adopt the commonly used cutoffs to define a conserved binding mode (MCS-RMSD < 2 Å, TvPLIF > 0.6, and COS > 0.55), I can inspect and count the number of discrepancies between the metrics. As I showed in Figure 3.3, COS and SuCOS are highly correlated and therefore adopted the same threshold for SuCOS. The values for the Pearson correlation coefficient R_P, the Spearman correlation coefficient, ρ, and the Kendall's tau, τ, are shown in each plot. (a) Scatter plot of MCS-RMSD against SuCOS. Points located in the upper-right part are considered as false negative (FN) points (MCS-RMSD > 2 Å and SuCOS > 0.55), and points located in the lower-left part are considered as false positive (FP) points (MCS-RMSD < 2 Å and SuCOS < 0.55). (b) Scatter plot of MCS-RMSD against TvPLIF. Points located in the upper-right part are considered as FNs (MCS-RMSD > 2 Å and TvPLIF > 0.6), and points located in the lower-left part are considered as FPs (MCS-RMSD < 2 Å and TvPLIF < 0.6). The green and purple stars represent the example FN given in Figure 3.6 and example FP given in Figure 3.7 respectively. (c) Scatter plot of TvPLIF against SuCOS. Points located in the lower-right part are considered as FNs (TvPLIF < 0.6 and SuCOS > 0.55), while points located in the upper-left are FPs (TvPLIF > 0.6 and SuCOS < 0.55).

Visual inspection of the ligand pairs corresponding to the FNs gives rise to some of the examples given in Table 3-2. These cases highlight the pitfalls of using atom-to-atom matching, like that in RDKit's MCS algorithm, to compute RMSD: if the molecules have pseudosymmetry, multiple substructure matches or substructures which have similar chemical properties.

PDB ID	2D Ligand representation	3D Representation		Comment
Smaller: 3ibi **Larger: 2gd8**			MCS-RMSD = 13.5 Å TvPLIF = 0.65 SuCOS = 0.59	MCS does not compare aliphatic chain against ring system.
Smaller: 2xht **Larger: 2vci**			MCS-RMSD = 6.5 Å TvPLIF = 0.80 SuCOS = 0.68	Good overlap of chemical groups with similar chemical properties (bioisostere) but MCS-RMSD calculation compares non-overlapping substructure due to better substructure match.
Smaller: 4eh4 **Larger: 3iw7**			MCS-RMSD = 6.9 Å TvPLIF = 0.77 SuCOS = 0.57	Pseudosymmetric molecule but good overlap of alicylic and aromatic rings.
Smaller: 1gpn* **Larger: 1h22***			MCS-RMSD = 13.7 Å TvPLIF = 0.33 SuCOS = 0.47	Multiple substructure matches in larger molecule
Smaller: 2vta* **Larger: 2r3k***			MCS-RMSD = 5.2 Å TvPLIF = 0.68 SuCOS = 0.46	Elaboration involves heteroatom changes to core. MCS would result in comparing the benzene ring

*These pairs are not FNs but nevertheless represent cases where the RMSD is misleading.

Table 3-2. Examples of cases where maximum common substructure RMSD is inappropriate for comparing poses of elaborated molecules with fragment hits. The PDB IDs of the smaller and larger structures are shown next to their corresponding 2D structures, with the common substructures used to compute the MCS-RMSD highlighted in red. The 3D representation shows a protein-based overlay of the fragment hit and larger ligand's crystal structures, generated by aligning the larger ligand's protein structure to that of the smaller ligand using the *align* function in PyMOL.(Schrödinger, LLC.) The MCS-RMSD, TvPLIF and SuCOS values for the aligned smaller and larger ligands are also shown, as well as an explanation of why MCS--RMSD is inappropriate.

It could be argued that this is an implementation error of the MCS algorithm; in the example of pseudosymmetry I included in Table 3-2, the smaller ligand, of 4eh4, contains a phenyl group linked via a carbonyl group to the nitrogen of a piperidine ring; the larger ligand, of 3iw7, also contains these same functional groups, but its piperidine has an extra para-substituent of a methylene linked to a second phenyl group. The aligned protein crystal structures show a structural overlap of the phenyl of the smaller ligand to the second phenyl of the larger ligand in the co-aligned binding pockets; the carbonyl of the smaller ligand structurally overlaps the methylene linker, and not the carbonyl linker; the piperidine ring roughly overlaps structurally, but is rotated ~180º. This means that the MCS algorithm correctly matched the exact maximum common substructure (carbonyl to carbonyl); but in this case, MCS would need to be modified to be able to detect chemical similarity between almost-identical substructures (carbonyl to methylene) and rotated rings. Thus, it is hard to argue that this is a bug in the implementation of MCS. SuCOS, however, is able to detect such chemical similarity and pseudo-rotations. Hence, for ligands of 4eh4 and 3iw7, MCS-RMSD (6.9 Å) suggests there is a big difference in binding modes, but SuCOS (0.57) and TvPLIF (0.77) detect similar modes.

Therefore, a non-substructure matching alternative such as SuCOS may be more appropriate to use when comparing poses of elaborated molecules against their non-elaborated counterparts.

Plotting MCS-RMSD against TvPLIF shows a weak negative correlation (Figure 3.5b, R_P = -0.46, ρ = -0.47 and τ = -0.32). The vertical line at TvPLIF = 0.6 is the proportion of interactions that Marcou and Rognan found that must be maintained to correspond to the 2 Å RMSD cutoff (Marcou and Rognan, 2007). Again, several points are situated in the top-right corner and bottom-left corner. These correspond to the FNs and FPs

respectively and make up 16% and 7% of the points respectively. 59% of the FNs in Figure 3.5b are also FNs in Figure 3.5a. Manual inspection of the remaining FNs show indeed a changed binding mode but the ligand manages to retain many of the original interactions. For example in pair 3adt/3ads, the larger ligand has changed binding mode but 70% of interactions are retained (Figure 3.6). Through processing of the Arpeggio output, only the interaction type and the residue number are kept, but no information is kept about which atom of the ligand is responsible for the interaction. Hence by this definition, the same interaction can be maintained with a different ligand atom which can accommodate some movement in the ligand.

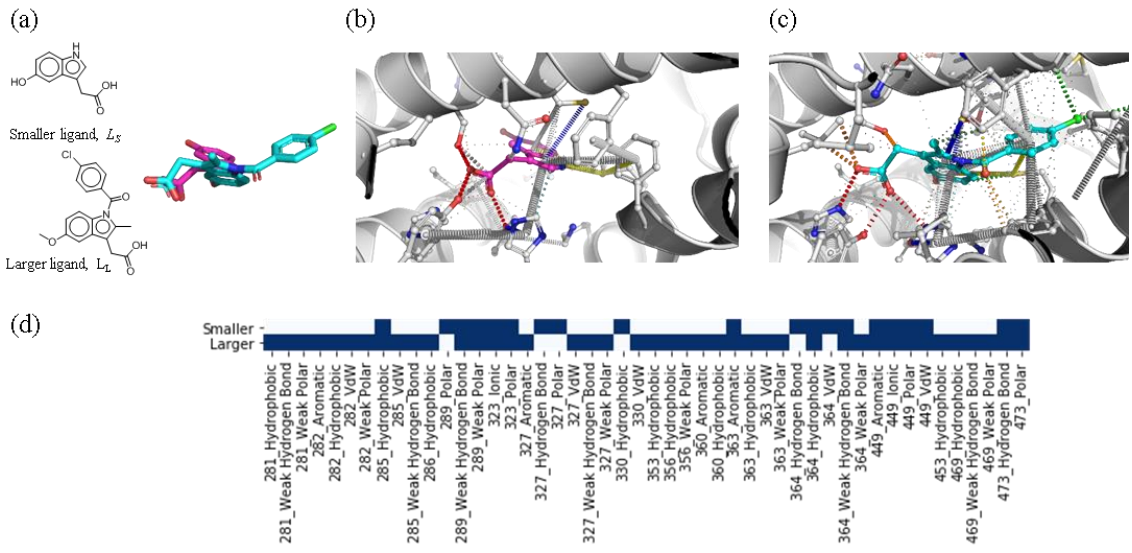


Figure 3.6. Example of a false negative in Figure 3.5b, where MCS-RMSD is high but TvPLIF is high. Aligned structures of 3adt/3ads have ligands with different binding modes yet many of the protein-ligand interactions are maintained by using different atoms within the ligand. The smaller and larger ligand are shown in pick and cyan sticks respectively. MCS-RMSD = 2.8 Å, TvPLIF = 0.70, SuCOS = 0.40.

Examination of the FP cases shows indeed very similar binding modes of the pairs, however the lower than expected TvPLIF may be explained by movement in the protein binding pocket and the strict definitions of distance and geometry for a particular interaction. Using 1ce5/1g3c as an example, there is good overlap of the smaller and larger ligand (MCS-RMSD = 1.2 Å, TvPLIF = 0.53, SuCOS = 0.81), however due to

conformational change in the protein binding site and slight differences in distances and angles, the TvPLIF is lower than expected (Figure 3.7). Furthermore, many of the interactions present in the smaller complex, but absent from the larger complex, are weak interactions. When computing PLIF similarity, all interaction types are given equal importance, but this can give a lower than expected TvPLIF, as seen in this example and many other FPs.

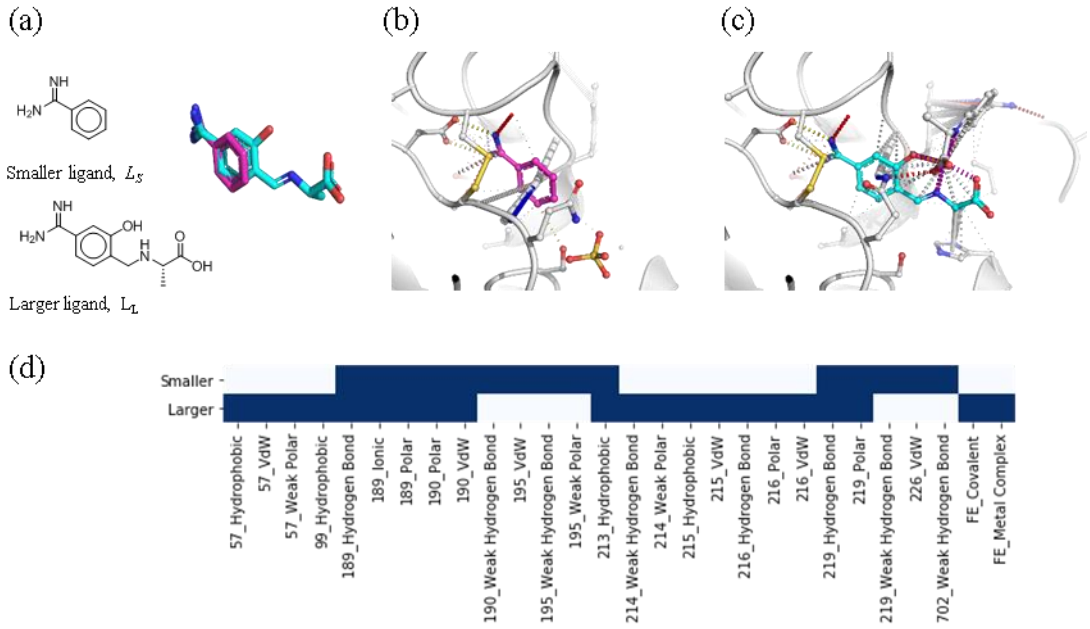


Figure 3.7. Example of a false positive in Figure 3.5b, where MCS-RMSD is low but TvPLIF is low. Aligned structures of 1ce5/1g3c have good overlap of the smaller (pick sticks) and larger ligand (cyan sticks) but TvPLIF is maybe lower than expected. MCS-RMSD = 1.2 Å, TvPLIF = 0.53 and SuCOS = 0.81.

The plot of TvPLIF against SuCOS shows a slightly better correlation (Figure 3.5c, R_P = 0.67, ρ = 0.59 and τ = 0.42). The points located in the top-left (SuCOS < 0.55, TvPLIF > 0.6) and bottom-right (SuCOS > 0.55, TvPLIF < 0.6) of the plot are labelled as FPs and FNs, which make up 8% and 9% of the points respectively. Manual inspection of some of the FNs can explain the lower than expected TvPLIFs. For example, in three of the structures missing residues were found in the PDB file (3uok, 3hoz, 1t48). Unlike RMSD and SuCOS which are ligand-centric metrics, PLIF is also

dependent on protein structure, hence care must be taken that the quality of the protein structure is adequate.

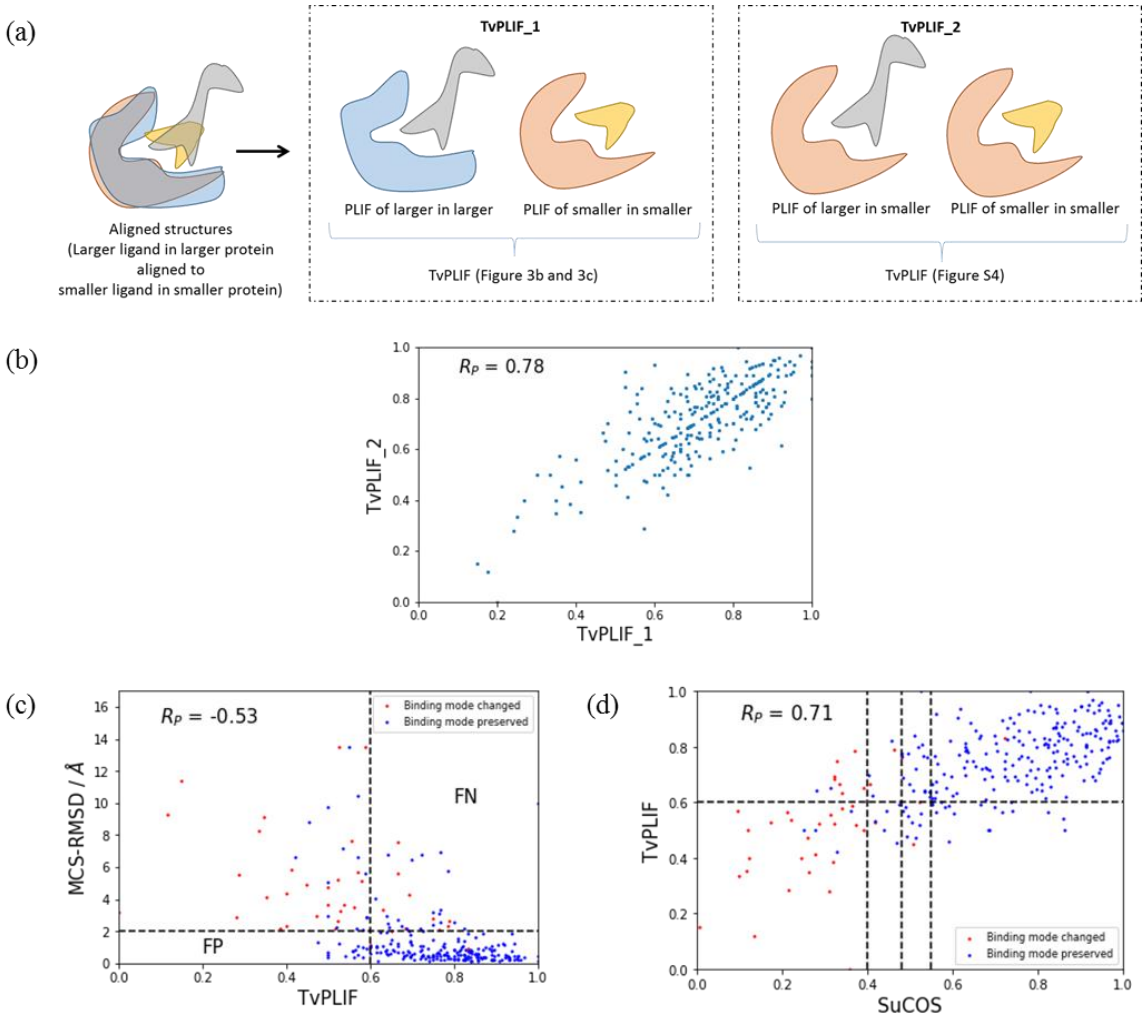


Figure 3.8. The TvPLIF between the smaller crystal structure and the larger crystal structure has some contribution from the change in conformation of the ligand binding site. Even if the ligands overlap perfectly in the aligned structures, their TvPLIF may not necessarily be 1 (all interactions may not be conserved) due to movement of the protein. (a) To eliminate this noise, the same protein conformation (the smaller protein) was used for both the larger and the smaller ligand. The TvPLIF was recomputed for all ligand pairs and scatter plots corresponding to Figure 3.5b and c were regenerated. (b) TvPLIF_1 corresponds to the TvPLIF calculated with the crystal structures of smaller and larger ligand. TvPLIF_2 corresponds to the TvPLIF calculated with the protein conformation of the smaller ligand for both smaller and larger ligands. The differences in TvPLIF due to different protein conformation is reflected in $R_P < 1$. (c) Better R_P was found for MCS-RMSD vs TvPLIF plot (-0.53 versus -0.46). (d) Better R_P was found for TvPLIF vs SuCOS plot (0.71 versus 0.67).

Furthermore, as the PLIF depends on the conformation of the protein binding side residues, there should be some noise that is a result of protein conformation differences between ligand pairs in Figure 3.5b and Figure 3.5c. To investigate this variable, the

larger ligands of the aligned structure, L_L^A, were combined with the smaller proteins structure, P_S^X, and the PLIFs computed on $P_S^X L_L^A$. For Figure 3.5b and Figure 3.5c, R_P was improved to -0.53 and 0.71 respectively (Figure 3.8). Indeed PLIF similarity is able to capture information about which interactions are kept or lost across multiple crystal structures of ligands bound to the same protein that ligand centric metrics such as RMSD and SuCOS cannot. However, if only one protein conformation is used *e.g.* in the redocking or docking numerous virtual ligands into the same protein, then information regarding the ligand pose should be captured using a ligand centric metric.

3.3.2 Part II: Using All-RMSD, TnPLIF and SuCOS to Rescore the Redockings of the Malhotra and Karanicolas Ligand Pair Set

Typically during a docking campaign, multiple poses are outputted for each docking run and the pose with the best docking score is usually chosen. RMSD is frequently the default metric for evaluating pose prediction, and the 2 Å cutoff is still widely used: any pose within 2 Å of the crystal ligand pose is deemed a 'successful' docking. As shape overlap and PLIF similarity have also been used to measure the conservation of binding mode, I also investigated their use in evaluation of docking success, including computing the rates of FPs and of FNs.

In Section 3.3.1, an asymmetric Tversky coefficient of shared protein-ligand interactions, TvPLIF, was used to compare binding modes of elaborated molecules against their smaller, non-elaborated counterparts. This was done to prioritise the interactions known to be made by the smaller ligand. However, the Tanimoto coefficient, TnPLIF, of shared interactions has been used in numerous studies to evaluate redocking (Anighoro and Bajorath, 2016a; Liu et al., 2017). TnPLIF measures

the similarity between two poses of the same molecule, typically its docked pose and its crystal structure.

In this section, each of the smaller ligands, L_S, was redocked into its cognate protein crystal structure, P_S^X, which produced a number of docked poses, $P_S L_S^{D,i}$, where i is the i^{th} pose produced for that docking. Similarly, the larger ligands, L_L, were redocked into their own cognate protein crystal structures, P_L^X:

$$P_S L_S^{D,i} = Dock\,(L_S\,,\, P_S^X) \tag{3.4}$$

$$P_L L_L^{D,i} = Dock(L_L\,,\, P_L^X) \tag{3.5}$$

where i is the i^{th} docked pose.

As the MK dataset was made non-redundant only in terms of PDB pairs and not by single PDB IDs, I extracted the set of unique PDB entries in the MK dataset, to give 485 unique PDB IDs for redocking using AutoDock Vina. From this, 436 ligands were successfully redocked into their cognate proteins. The remaining 49 failed because of various errors, including the protein crystal structure having incomplete loops; the ligand having unusual atom types, such as boron; or the docked ligand SDF being unreadable by RDKit. For each ligand, up to 9 poses were generated (see Section 3.2.5). In total, 3793 poses were produced and for each pose All-RMSD, TnPLIF and SuCOS were computed with reference to its crystal structure pose. Each pose also had an associated Vina score, Aff^{Vina}. Each metric was then used to rank the poses and select a single pose, *i.e.* the pose with the lowest RMSD, $Pose(RMSD_{best})$; that with the highest TnPLIF, $Pose(TnPLIF_{best})$; that with the highest SuCOS, $Pose(SuCOS_{best})$; and that with the lowest Aff^{Vina}, $Pose(Aff_{best}^{Vina})$:

$$Pose(RMSD_{best}) = Min(\,RMSD(L_S^{D,1}, L_S^X), RMSD(L_S^{D,2}, L_S^X), \ldots\ RMSD(L_S^{D,N}, L_S^X)\,) \tag{3.6}$$

$$Pose(TnPLIF_{best}) = Max(\, TnPLIF(L_S^{D,1}, L_S^X), TnPLIF(L_S^{D,2}, L_S^X), \ldots \; TnPLIF(L_S^{D,N}, L_S^X)\,) \quad (3.7)$$

$$Pose(SuCOS_{best}) = Max(\, SuCOS(L_S^{D,1}, L_S^X), SuCOS(L_S^{D,2}, L_S^X), \ldots \; SuCOS(L_S^{D,N}, L_S^X)\,) \quad (3.8)$$

$$Pose(Aff_{best}^{Vina}) = Min(\, Aff^{Vina}(L_S^{D,1}), Aff^{Vina}(L_S^{D,2}), \ldots \; Aff^{Vina}(L_S^{D,N})\,) \quad (3.9)$$

where N is the total number of docked poses produced for that ligand. If multiple poses had the same best score, then only the first occurring pose was kept for that metric.

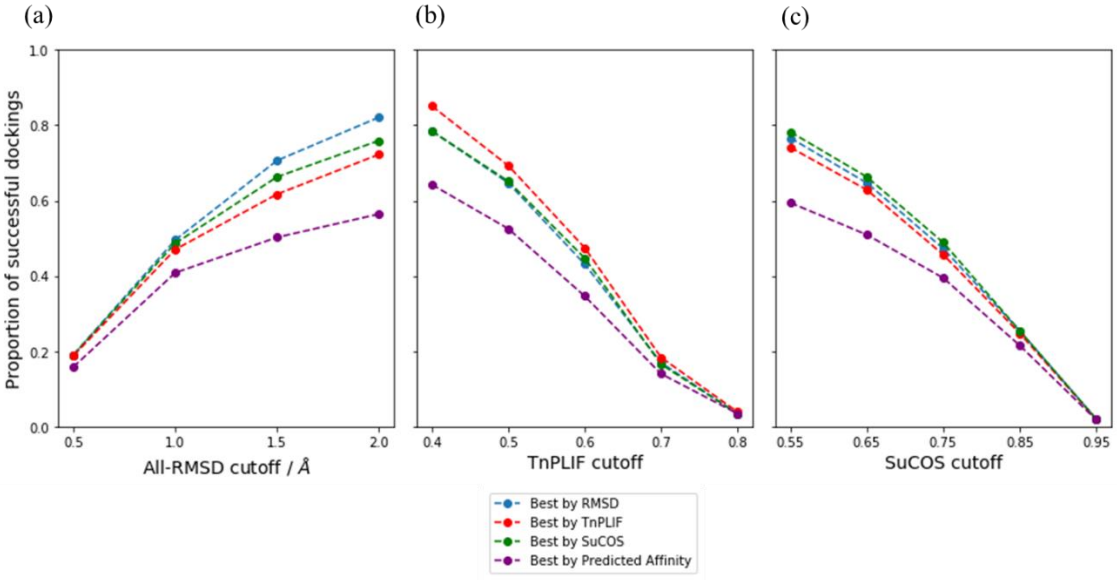


Figure 3.9. Choosing by affinity consistently performs the worst at picking the docking pose which most closely resembles the crystal ligand pose. Each ligand in the Malhotra Karanicolas ligand pair set was redocked back into its cognate protein. Ranking the poses by each metric – All-RMSD, TnPLIF, SuCOS, and Vina Predicted Affinity – leads to differing success rates for each metric. The proportion of successful redockings is plotted against various cutoffs shown on the *x* axis, namely (a) All-RMSD, (b) TnPLIF, and (c) SuCOS.

I looked at the effect of varying the cutoff for All-RMSD, TnPLIF, and SuCOS on the proportion of dockings that matched the crystallographic binding mode (Figure 3.9). Using the standard 2 Å cutoff, 358 of the 436 redockings (82%) generated at least one 'successful' pose. Ranking the docked poses using the other metrics, TnPLIF, SuCOS, and Aff^{Vina}, I found 315 (72%), 331 (76%), and 246 (56%) of the poses were within 2 Å of the crystallographic binding mode, respectively (Figure 3.9a). It is worth noting that the success rate when ranking docked poses by the native Vina score, 56%, was less

than the reported 78% success rate (Trott and Olson, 2010). There could multiple reasons for this including differences in dataset composition *e.g.* differences in the distribution of molecular sizes.

In addition to All-RMSD, the success of pose prediction can be defined by SuCOS or TnPLIF (Figure 3.9 (b) and (c) respectively). Ranking poses by Aff^{Vina} consistently performed the worst at pose prediction. The difficulty of accurate affinity prediction has already been well established and is supported by the recent study of Ramírez and Caballero (Ramírez and Caballero, 2018). Furthermore, the standard error of the AutoDock Vina scoring function is 2-3 kcal/mol (Morris et al., 2009), so if there is little variation between the affinity of the poses, it will perform poorly in picking the 'correct' pose. Rescoring by TnPLIFs and SuCOS leads to greater pose prediction success as seen in previous studies (Drwal et al., 2017; Desaphy et al., 2013; Marcou and Rognan, 2007; Verdonk et al., 2016; Kumar and Zhang, 2016b; Anighoro and Bajorath, 2016b).

All-RMSD and SuCOS performed similarly across all three methods of defining success, which suggests that SuCOS is a good non-substructure matching alternative to All-RMSD. The 2 Å All-RMSD cutoff corresponds to approximately SuCOS = 0.55 and TnPLIF = 0.4 (Figure 3.9). This TnPLIF value is lower than the 0.6 TnPLIF cutoff found by Marcou and Rognan (Marcou and Rognan, 2007). I used Arpeggio to calculate the PLIFs, which is based on an expanded definition of Marcou and Rognan's interaction types and contains additional interaction types. For example, Arpeggio has polar and weak polar interaction types, whereas Marcou and Rognan's in-house method did not. This algorithmic difference, in addition to differences in datasets may explain the different cutoffs found. Despite the difference in TnPLIF cutoffs, I chose to use the

cutoff of 0.6 as it was the value previously used in the literature (Marcou and Rognan, 2007); however, the analysis could also be repeated for the 0.4 threshold.

Where TnPLIF disagrees with all other metrics, it is again worth noting that PLIFs are highly sensitive to distance and direction. Taking the redocking of ligand of 1o39 as an example (Figure 3.10), another disadvantage of ranking by TnPLIF is made clear as penalties are introduced when a docked pose makes more interactions than the crystallographic binding mode. Furthermore, the ligand binds to the surface of the protein and is partially solvent exposed. This part of the ligand that does not interact with any residues is not captured in the PLIF. Unlike RMSD and SuCOS, the nature of the binding site can influence how much of the ligand pose is captured.

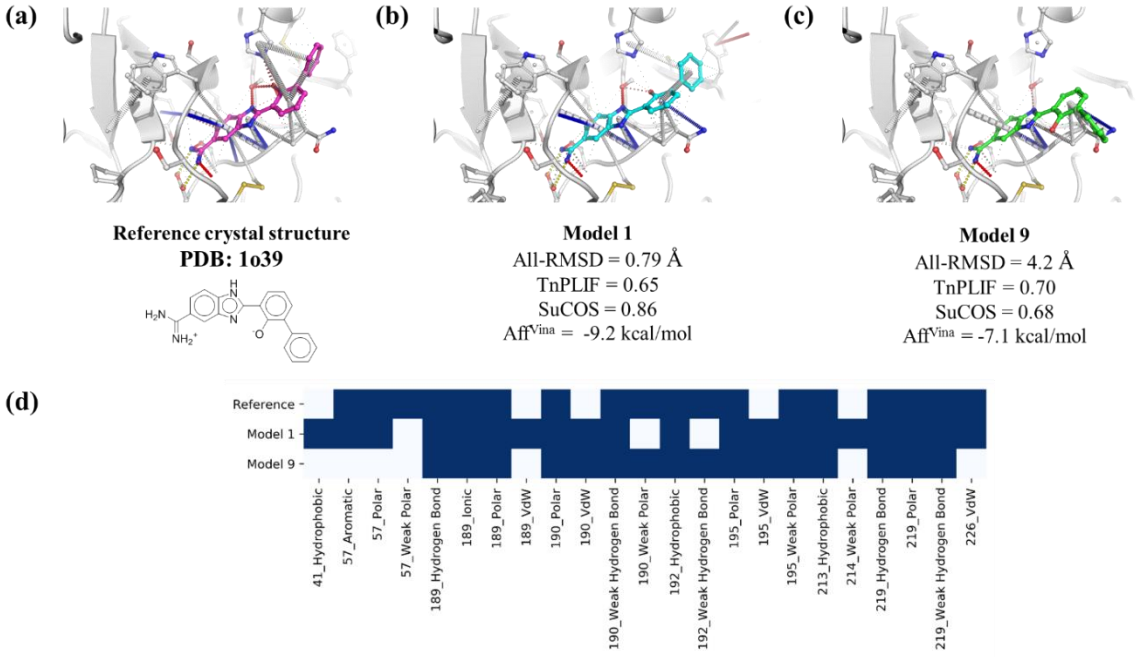


Figure 3.10. Redocking ligand of 1o39: example of a disadvantage of using TnPLIF when redocking. Both model 1 and model 9 have very similar TnPLIF scores; however, model 1 is very similar to the crystal structure pose and model 9 is quite different. (a) The crystallographic binding mode of the ligand is shown in pink sticks. (b) Model 1 is shown in cyan sticks and is ranked best by All-RMSD, SuCOS and Aff^{Vina}. The pose overlaps almost exactly with the crystal structure of the ligand but it forms several additional interactions by just slight differences in orientation of the ligand. (c) Model 9 is shown in green sticks and is ranked best by TnPLIF. It makes fewer interactions than are present in the crystal structure, but the terminal phenyl ring is incorrectly pointing out of the binding pocket. (d) PLIF interaction heatmap of the reference, Model 1 and Model 9. The presence of an interaction is shown by a blue square. The notation for the PLIFs along the *x*-axis is the residue number and the interaction type: *e.g,* 41_Hydrophobic refers to a hydrophobic interaction with residue 41.

In the case of redocking, the reference and query molecule are identical, so All-RMSD is used. When comparing molecules and their elaborated counterparts, RMSD depends on defining the pairs of corresponding atoms to be used in the calculation of the positional deviations.

Some of the problems shown in the examples where RMSD is inappropriate to use for elaborated molecules in Section 3.3.1 (Table 3-2) may also arise when comparing identical molecules. Therefore, all poses for all redockings were visually inspected and each redocking was manually classified as "successful" (had at least one pose that closely matched the crystallographic binding mode), or "unsuccessful" (had no poses that closely matched the crystallographic binding mode). This classification is indeed somewhat subjective, however, as each metric may contain FPs and FNs, it was done so that the strengths and weaknesses of metric can be understood.

Thus, for each metric the number of TPs (*i.e.,* the metric correctly classified a successful docking), TNs (*i.e.,* the metric correctly classified an unsuccessful docking), FPs (*i.e.,* incorrectly classified a successful docking) and FNs (*i.e.,* incorrectly classified an unsuccessful docking) was recorded (Table 3-3).

Criterion	**TP**	**TN**	**FP**	**FN**
All-RMSD < 2 Å	336	70	22	8
TnPLIF > 0.60	211	91	1	133
SuCOS > 0.55	338	88	4	6

Table 3-3. Visual inspection of all the poses from the redockings shows that all metrics – All-RMSD, TnPLIF and SuCOS – give false positives and false negatives. The table gives the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each metric and criterion.

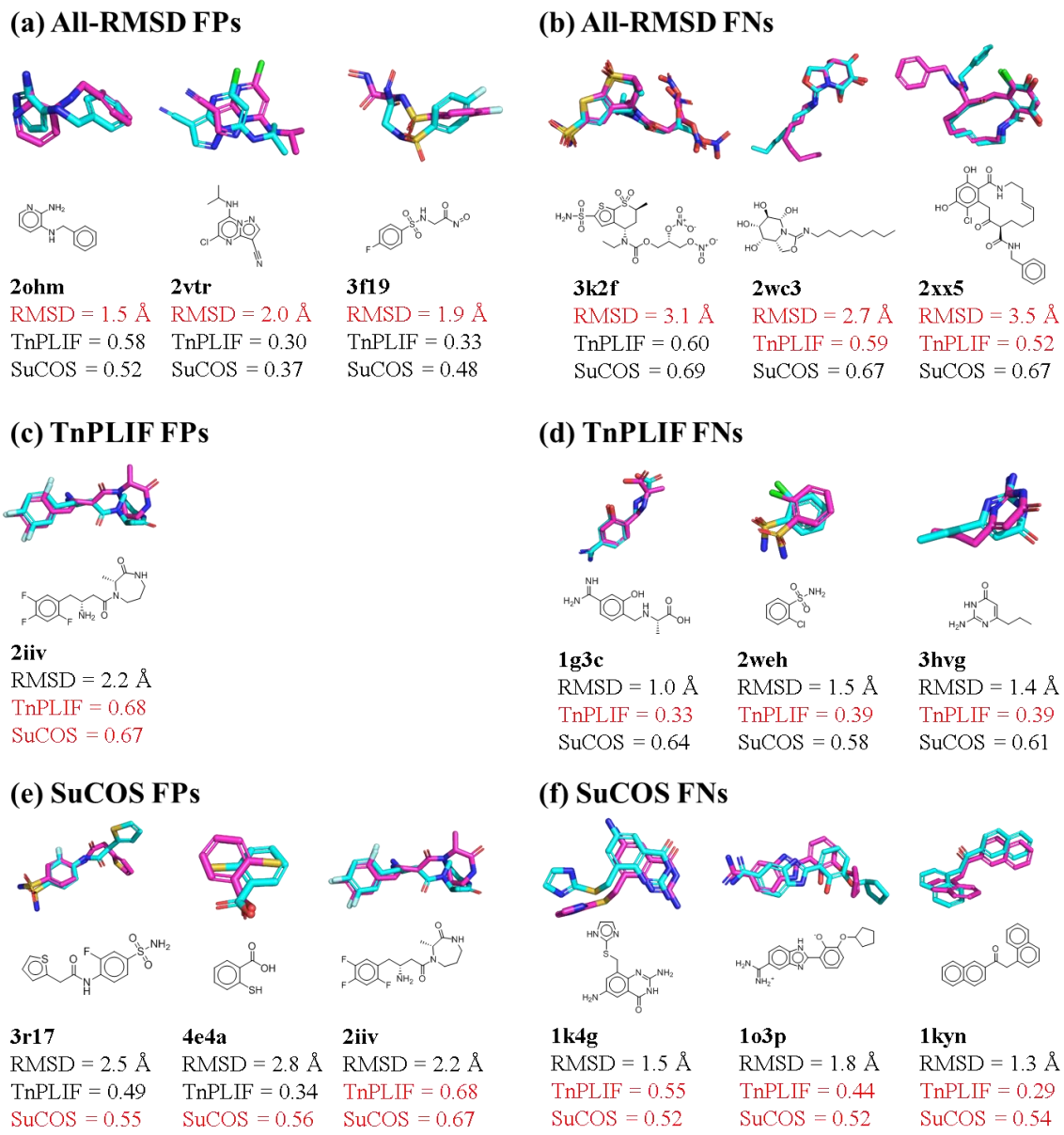


Figure 3.11. (a) – (f) Three examples of FPs and FNs for each metric are shown. The crystal ligand is shown in cyan and the docked pose in pink throughout. The PDB code is shown under each structure, together with the All-RMSD, TnPLIF and SuCOS values for the docked pose against the crystal ligand pose. The values highlighted in red are the false values.

Visual inspection of the FPs and FNs for All-RMSD again highlighted some of its weaknesses. For example, there were numerous cases of FPs where the molecule is small. The RMSD metric is size dependent (Hawkins and Nicholls, 2012; Hawkins et al., 2008) and using the 2 Å cutoff for smaller ligands may be too large. Normalization of RMSD by molecular size, or using size-appropriate cutoffs for different sized molecules may overcome this, yet the 2 Å cutoff is still widely used. For All-RMSD

FNs, there were cases where there is good overlap of the cores of the crystal ligand and docked pose but the side chain has changed conformation (see 2wc3 and 2xx5 in Figure 3.11b). This dramatically increases the All-RMSD value, as discussed by Hawkins *et al.* (Hawkins and Nicholls, 2012) but SuCOS is much less affected. Pseudosymmetry led to a high All-RMSD for 3k2f despite the good overlap of the docked and crystal ligand pose.

The relatively low number of TPs and high number of FNs for TnPLIF can be attributed to the cutoff of TnPLIF = 0.6 being stricter and not equivalent to the 2 Å RMSD cutoff as discussed before. Indeed, reducing this cutoff to TnPLIF = 0.4 decreases the number of FNs and FPs to 7 and 37 respectively.

Several of the SuCOS FPs resemble the All-RMSD FNs in that their cores overlap but the rest of the molecule differs (*e.g.,* 3r17 and 2iiv, Figure 3.11e). For the six SuCOS FNs, several have staggered rings when comparing the docked pose to the crystal pose (1k4g, 1o3p and 1kyn, Figure 3.11f). For each example, there is a good contribution of shape overlap but not feature overlap to SuCOS (1k4g: shape = 0.68, features = 0.36; 1o3p: shape = 0.69, features = 0.35; 1kyn: shape = 0.70, features = 0.38). For shape overlap, the type of atom that overlaps is not considered, so for staggered heteroatomic rings, a relatively high shape overlap can be maintained. However, for shape plus feature overlap, there also needs to be exact matches of each feature type, so for staggered heterocycles the overlap of features can be poor. This explains why these staggered ring poses have relatively poor SuCOS scores. However, one way of potentially overcoming this would be to increase the weight of the shape overlap component and decrease the weight of the chemical feature overlap for molecules which are largely composed of heteroatomic rings. Alternatively, it could be addressed by increasing the default cutoff distance for matching features from 2.5 Å.

3.3.3 Part III: Comparison of All-RMSD/MCS-RMSD, TvPLIF and SuCOS on the Cross-Docked Larger Ligand into the Smaller Ligand's Protein Structure of the Malhotra and Karanicolas Ligand Pair Set

I refer to the scenario where a ligand is docked into the same protein but with a different protein conformation as "cross-docking". Here, I docked the larger ligand, L_L, into its paired smaller ligand's protein crystal structure, P_S^X (Figure 3.1c). This simulates a virtual screening effort investigating potential fragment-hit follow-ups by docking them into the fragment-hit's protein structure. Each docking produced a number of docked poses, $PsL_L^{D,i}$, where i is the i^{th} docked pose produced for that cross-docking:

$$PsL_L^{D,i} = Dock(L_L, P_S^X) \tag{3.10}$$

Of the total 284 larger ligands, 242 were successfully docked into the smaller ligand's protein structure, due to various errors as discussed in Section 3.3.2 such as the protein crystal structure having incomplete loops, or the ligand having unusual atom types. As described earlier, after generating up to 10 conformers for each larger ligand, each conformer was docked using AutoDock Vina giving a total of 11,879 poses for the whole set, with an average of ~49 poses for each ligand. The MCS-RMSD was calculated for each pose of the larger, elaborated ligand comparing it to its corresponding smaller ligand, L_S^X, while All-RMSD was computed when comparing to itself. TvPLIF and SuCOS were also calculated for each docking pose with respect to both L_S^X and L_L^A.

The distributions of all metrics were all closer to L_S^X than to L_L^A (Figure 3.12). This is not unexpected, as comparing a larger ligand with its paired smaller ligand requires

only part of the larger ligand to be similar. For example, for RMSD, only the maximum common substructure needs to overlap, while the rest of elaborated portion has no restrictions. Comparing the larger ligand with its crystal pose requires the whole structure to match.

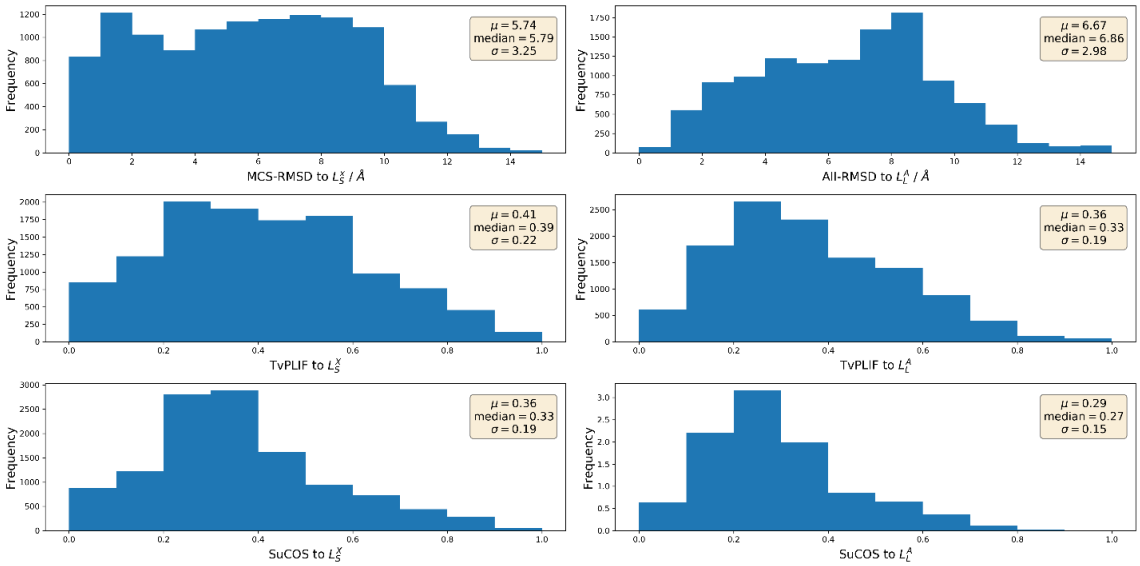


Figure 3.12. Calculation of the metrics for all cross-docking poses shows that all the distributions are closer to the smaller ligand, $L_S{}^X$, than to the larger ligand, $L_L{}^A$. RMSD, TvPLIF and SuCOS were calculated for each cross docking pose with respect to its corresponding smaller, L_S^X, and larger ligands, L_L^A, which were aligned in the initial study. The results are shown for the 242 cross-docked ligands and a total of 11,843 poses. Each histogram has a legend that shows the mean (μ), median and standard deviation (σ).

I investigated whether picking poses using different metrics affected cross-docking success. For each of the 242 cross-dockings, one pose was kept for each metric. I compared the poses of the larger ligand with the crystal pose of the smaller ligand and retained the following: the one with the lowest MCS-RMSD, Eq. 3.10; the highest TvPLIF, Eq. 3.11; and the highest SuCOS, Eq. 3.12. The pose with the best AutoDock Vina affinity was also kept (Eq. 3.13). The results are summarized in Table 3-4.

$$Pose(RMSD_{best\_to\_smaller}) = Min(\, RMSD(L_L^{D,1}, L_S^X), RMSD(L_S^{D,2}, L_S^X), \ldots \; RMSD(L_S^{D,i}, L_S^X)\,) \quad (3.11)$$

$$Pose(TvPLIF_{best\_to\_smaller}) = Max(\, TvPLIF(L_S^{D,1}, L_S^X), TvPLIF(L_S^{D,2}, L_S^X), \ldots \; TvPLIF(L_S^{D,i}, L_S^X)\,) \quad (3.12)$$

$$Pose(SuCOS_{best\_to\_smaller}) = Max(\, SuCOS(L_S^{D,1}, L_S^X), SuCOS(L_S^{D,2}, L_S^X), \ldots \; SuCOS(L_S^{D,i}, L_S^X)\,) \quad (3.13)$$

$$Pose(Aff_{best}^{Vina}) = Min(\, Aff^{Vina}(L_S^{D,1}), Aff^{Vina}(L_S^{D,2}), \ldots \; Aff^{Vina}(L_S^{D,i})\,) \quad (3.14)$$

Criterion	Number of poses	Number with All-RMSD < 2 Å to L_L^A (/242)	Number with TvPLIF > 0.6 to L_L^A (/242)	Number with SuCOS > 0.55 to L_L^A (/242)
$\boldsymbol{Pose(\text{RMSD}_{best\ to\ L_S^X})}$	242	81	104	100
$\boldsymbol{Pose(TvPLIF_{best\ to\ L_S^X})}$	242	60	116	82
$\boldsymbol{Pose(SuCOS_{best\ to\ L_S^X})}$	242	76	109	106
$\boldsymbol{Pose(Aff_{best}^{Vina})}$	242	62	81	78
MCS-RMSD < 2 Å to L_S^X	2,050	119	106	114
TvPLIF > 0.6 to L_S^X	2,338	111	121	113
SuCOS > 0.55 to L_S^X	1,949	114	111	117
Keeping all	11,879	140	153	134

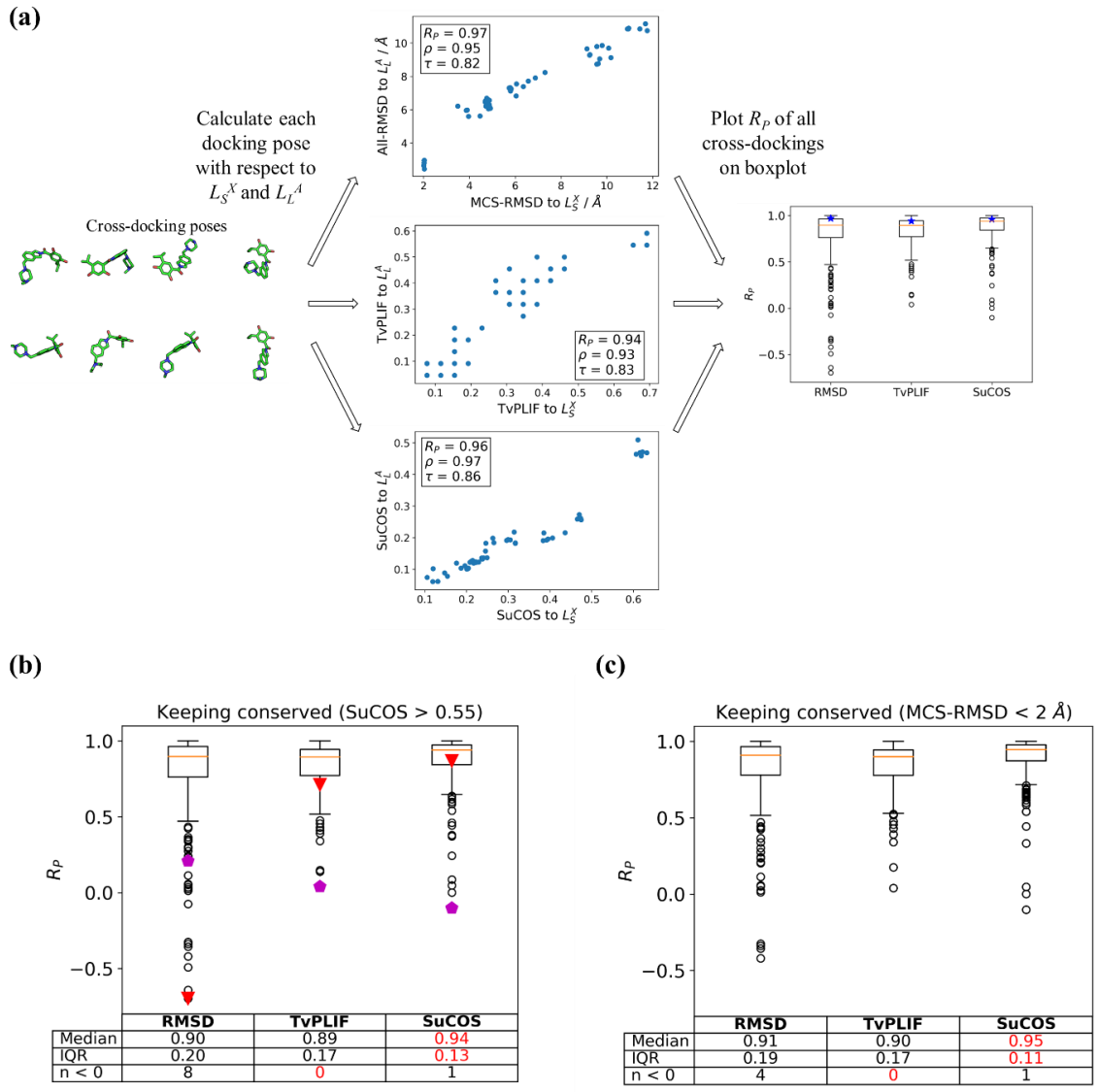
Table 3-4. Summary of cross-docking the larger ligand, L_L, from the MK dataset into the protein from the complex containing its paired smaller ligand, P_S^X. Instead of using the native docking score, Aff^{Vina}, of the larger ligand from the docking, L_L^D to select a docked pose, it is possible to rank all of the docked poses of the larger ligand, L_L, by computing the MCS-RMSD, TvPLIF, and SuCOS against the smaller ligand's crystal structure, L_S^X. The success rates were computed against L_L^A. The criteria used to define a successful docking were All-RMSD < 2 Å, SuCOS > 0.55 and TvPLIF > 0.6. The success rates if all poses within a cutoff with respect to L_S^X are kept are also shown. The maximum success rate is also shown for each metric, if all poses are kept. Choosing one pose from each cross-docking leads to a much lower success rate across all metrics considered. A much higher recovery rate of the crystal structure of the larger ligand can be achieved if all poses within a given threshold are kept.

Using the definition of success as All-RMSD < 2 Å with respect to L_L^A, choosing the best pose by MCS-RMSD with respect to L_S^X achieved a success of 33% (81 cross-dockings) (Table 3-4). Similarly, choosing best by TvPLIF and SuCOS with respect to the L_S^X gives a success of 25% (60 cross-dockings) and 31% (76 cross-dockings) respectively. Interestingly, choosing best by Aff^{Vina}, gives a slightly better pose prediction than ranking by TvPLIF, with 26% success (62 cross-dockings). However, it should be noted that if all poses are kept, then 58% (140 cross-dockings) achieved at least one successful pose (Table 3-4).

Using this information, keeping all poses that satisfy a cutoff with respect to the smaller ligand pose may lead to greater pose prediction than keeping only the best pose. Hence, the following cutoffs were used to keep all poses that satisfy that cutoff: for MCS-RMSD, < 2 Å, for TvPLIF, > 0.6, for SuCOS, > 0.55. These criteria retained 17% (2,050), 20% (2,338) and 16% (1,949) poses respectively. With these cutoffs, the success rates for MCS-RMSD, TvPLIF and SuCOS were 49% (119), 46% (111) and 47% (114) respectively.

This suggests that in a virtual screen where only the structural information of a smaller non-elaborated ligand is known, using only MCS-RMSD, TvPLIF or SuCOS to score against the smaller ligand to keep just one pose, may filter out poses which are actually closer to the larger ligand crystal pose. Alternatively, keeping all poses within a threshold of the smaller ligand crystal pose will give a better success of pose prediction.

Next, for each metric I considered the correlation of every docked pose with respect to the crystal structure of the smaller ligand, L_S^X, and the aligned crystal structure of the larger ligand, L_L^A (Figure 3.13a). If a docking pose scores well with the smaller ligand crystal pose, then it should also score well with the larger ligand crystal pose and *vice versa*, provided the smaller ligand and its elaborated counterpart have a conserved binding mode — which was true in 86% of the cases studied by Malhotra and Karinicolas (Malhotra and Karanicolas, 2017). Therefore, the better the correlation, the better the metric should do at differentiating good poses.



(b)

	RMSD	TvPLIF	SuCOS
Median	0.90	0.89	0.94
IQR	0.20	0.17	0.13
n < 0	8	0	1

(c)

	RMSD	TvPLIF	SuCOS
Median	0.91	0.90	0.95
IQR	0.19	0.17	0.11
n < 0	4	0	1

Figure 3.13. SuCOS has the best correlations between docking poses of a larger ligand to its respective crystal pose and to its smaller counterpart crystal pose. (a) Schematic showing how boxplots (b) and (c) were created. For each cross-docking pose, the RMSD, TvPLIF, and SuCOS values were calculated with respect to the smaller and larger crystal ligand pose. These scores can be plotted on a scatter plot, with each point on the plots representing a single pose. The Pearson correlation coefficients, R_P, were then calculated for each metric for each cross-docking, comparing the smaller ligand and the larger, elaborated ligand. The blue stars on the boxplots represent the Pearson correlation coefficients obtained for the example cross-docking shown. (b) Using SuCOS > 0.55 to define a conserved binding mode, the cross-docking pairs were filtered so only crystal ligands that showed a conservation of binding mode were kept. The collated Pearson correlation coefficients for each metric are shown on the boxplot. Four Pearson correlation coefficients were not included as there were fewer than nine outputted docking poses. The median, interquartile range (IQR) and number of negative Pearson correlation coefficients are shown in the table below the boxplot. The red triangles denote the RMSD outlier example shown in Figure 3.16. The magenta pentagons denote the SuCOS outlier example shown in Figure 3.15. (c) Defining the conservation of binding mode with SuCOS does not bias the results. If the crystal ligand pairs are filtered according to MCS-RMSD (crystal ligand pairs with MCS-RMSD < 2 Å kept), SuCOS still performs the best in terms of highest median and lowest IQR. The best values are highlighted in red.

Using SuCOS > 0.55 to define a conserved binding mode, the MK dataset was filtered to include only those elaborated ligands with a conserved binding mode. For each cross-docking, the All-RMSD to L_L^A was plotted against the MCS-RMSD to L_S^X for all the poses of that cross-docking. The distributions of Pearson correlation coefficients were plotted as boxplots and similar boxplots were drawn for TvPLIF and SuCOS (Figure 3.13b).

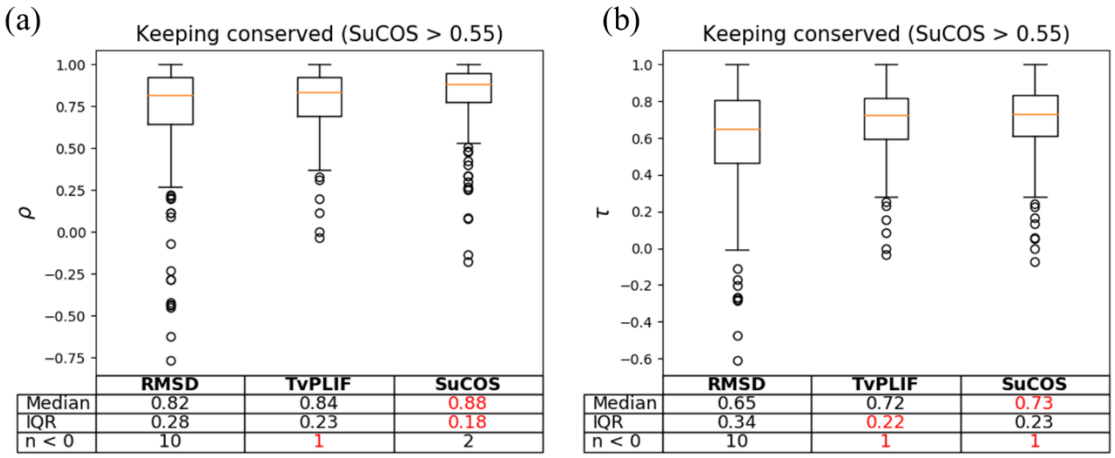


Figure 3.14. For each cross-docking pose, the RMSD, TvPLIF, and SuCOS values were calculated with respect to the smaller and larger crystal ligand pose. These scores can be plotted on a scatter plot, with each point on the plots representing a single pose. Using SuCOS > 0.55 to define a conserved binding mode, the cross-docking pairs were filtered so only crystal ligands that showed a conservation of binding mode were kept. Boxplots for each metric are shown for the collated (a) Spearman's ρ and (b) Kendall's τ. Four correlation coefficients were not included as there were fewer than nine outputted docking poses. The median, interquartile range (IQR) and number of negative correlation coefficients are shown in the table below each boxplot.

A Pearson correlation coefficient of one indicates there is perfect correlation between the cross-docking poses and scoring to both the smaller ligand crystal pose and the larger ligand crystal pose. In most cases for all three metrics — RMSD, TvPLIF, and SuCOS — this correlation is near to one. In terms of median and interquartile range (IQR), however, SuCOS performed the best, with the highest median (0.94) and lowest IQR (0.13). SuCOS also performed similarly well when the median correlations were calculated for Spearman's ρ, and Kendall's τ (Figure 3.14).

In order to avoid bias by using SuCOS to define the conservation of binding mode, I also performed the same analysis but filtered the MK dataset selecting only those poses with a conserved binding mode using the criterion MCS-RMSD < 2 Å (Figure 3.13c). SuCOS still performed better than RMSD and TvPLIF, in terms of both highest median and lowest IQR (0.95 and 0.11, respectively).

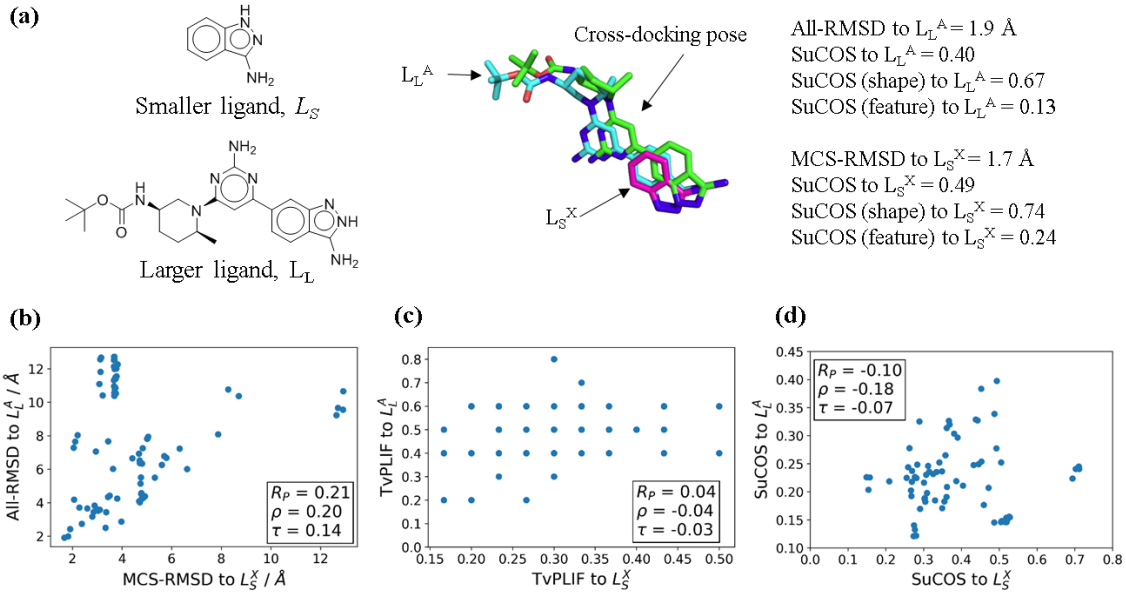


Figure 3.15. Cross-docking on 3nus/3qd3: an example of a case where plotting the SuCOS to the larger crystal ligand against the SuCOS to the smaller crystal ligand for all the cross docking poses gives a negative R_P. (a) The cross docking pose with the lowest All-RMSD to the larger ligand is shown together with L_S^X and L_L^A. The cross docking pose, L_S^X and L_L^A are shown in green, pink and cyan sticks respectively. (c) Scatter plots showing the Pearson R_P, Spearman's ρ and Kendall's τ of the cross-docking poses with the smaller and larger crystal structures for (b) RMSD, (c) TvPLIF, (d) and SuCOS.

It is interesting to note there were a small number of cases with a negative Pearson correlation coefficient. SuCOS had one case with a negative R_P, whereas TvPLIF and RMSD had zero and eight respectively (Figure 3.13b). In these cases, the cross-docking poses were visually inspected. For the one case of negative R_P for SuCOS (cross-docking of 3nus/3qd3), the corresponding R_p values for RMSD, TvPLIF and SuCOS are 0.21, 0.04 and -0.10 respectively (Figure 3.15). The docking achieved a pose within 2 Å RMSD of the smaller and larger ligand. In this docked pose, the core rings are slightly staggered with respect to the crystal ligand pose. This resembles the SuCOS false

negative examples shown in Figure 3.11f. The ligand has multiple heteroatom rings, which may explain why these translations do not score well with SuCOS but better with RMSD. This cross-docking pose has good shape overlap but poor feature overlap with both the smaller and larger ligand. Consequently, any docking pose that corresponds to translation of rings will not score well.

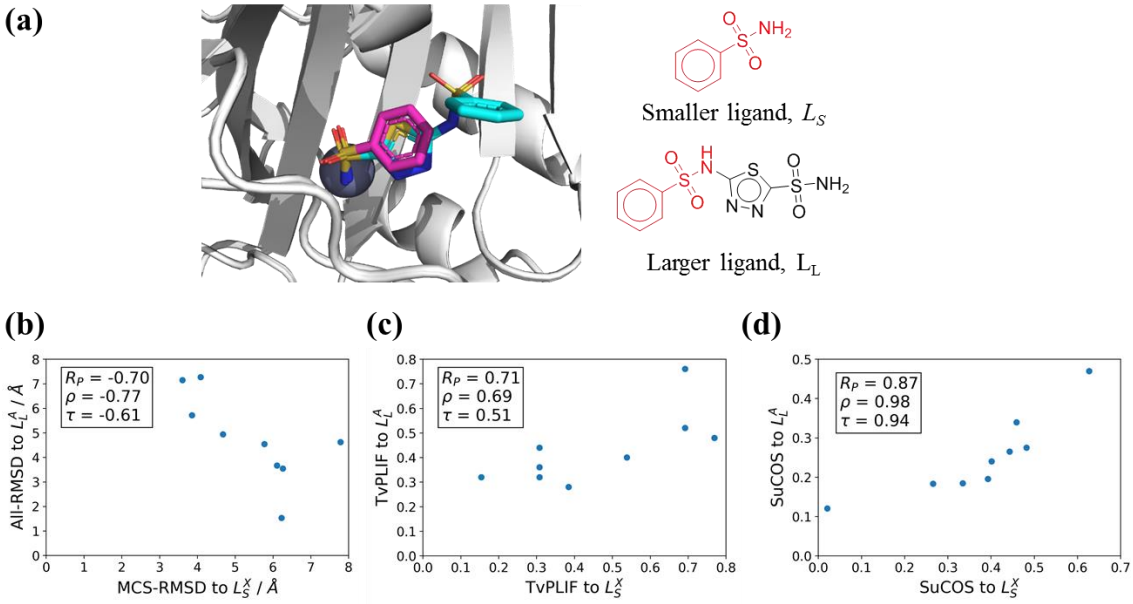


Figure 3.16. Cross-docking of 2wej/3d8w: an example of a case where plotting the All-RMSD to the L_L^A against the MCS-RMSD to L_S^X for all the cross-docking poses gives a negative R_P. (a) 3D structure of smaller (pink) and larger (cyan) ligands in the binding sites of the aligned crystal structures of 2wej/3d8w. The 2D structure of the smaller and larger ligands are shown on the right with the MCS that was used to compute the RMSD highlighted in red. (b) Scatter plots showing the Pearson R_P, Spearman ρ and Kendall's τ of the cross-docking poses with the smaller and larger crystal structures for (b) RMSD, (c) TvPLIF, (d) and SuCOS.

The cross-docking of 2wej/3d8w is an example of where RMSD has a negative R_P (Figure 3.16). This pair has good overlap of the crystal poses but the MCS-RMSD matches substructures that do not overlap (similar to 2xht/2vci in Table 3-2). This substructural mismatch resulted in the negative R_P found in the plot of RMSDs of the cross-docking poses to the large versus RMSD to the small (Figure 3.16b).

$$SuCOS = (1 - \lambda)(ScoreFeats) + \lambda(1 - ShapeProtrudeDist) \quad (3.15)$$

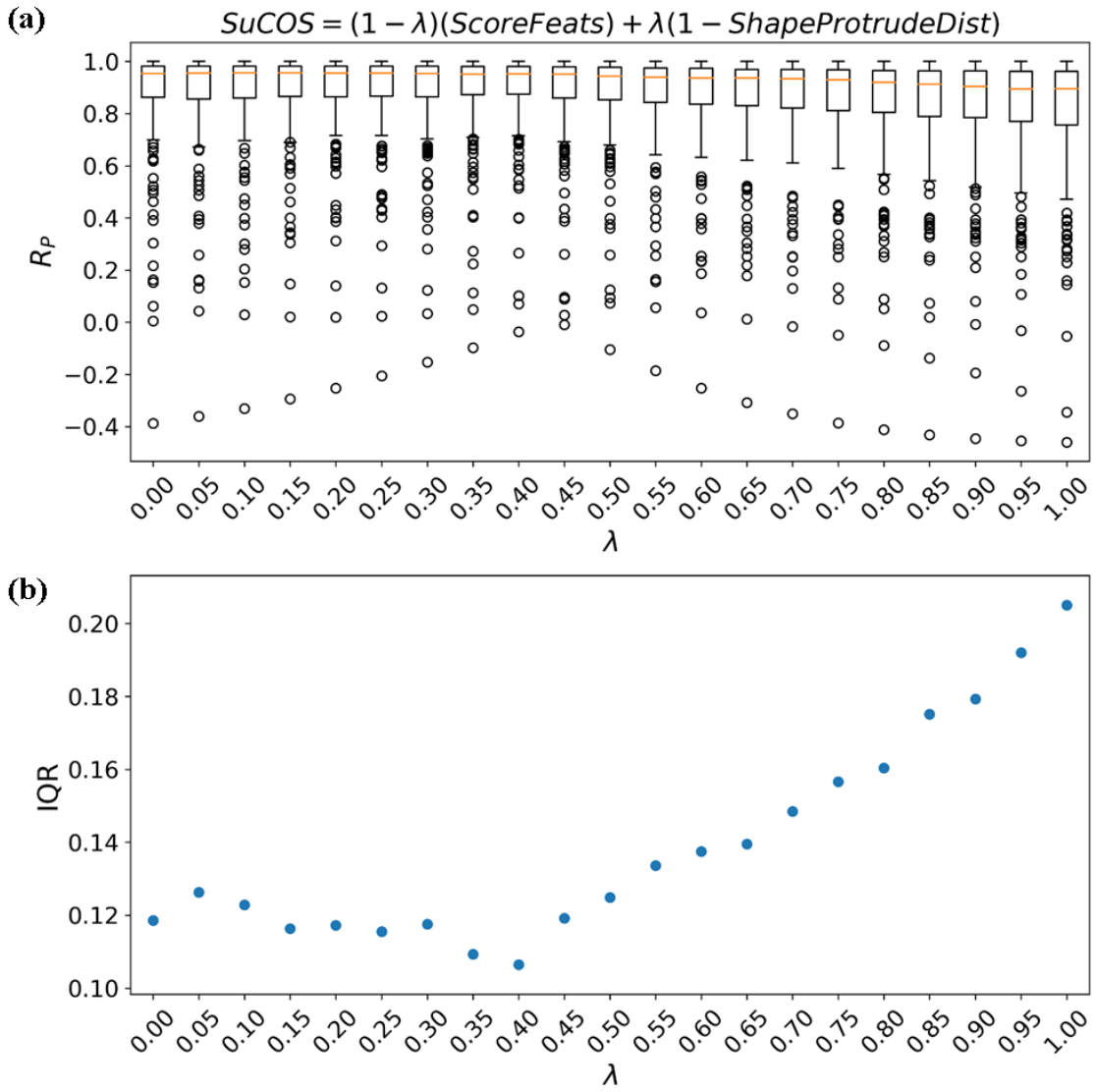


Figure 3.17. Investigating the effect of altering the weights of feature overlap and shape overlap in SuCOS. (a) For each weighting, the distribution of Pearson correlation coefficients of the cross-dockings were plotted as boxplots. For the two extreme weightings, $\lambda = 0$ represents SuCOS using only chemical feature overlap and no shape overlap, while $\lambda = 1$ represents SuCOS with all shape overlap and no chemical feature overlap. (b) Plotting the IQR of the boxplots in (a) against the weights.

The weights of chemical feature overlap and shape overlap in the SuCOS metric have so far been assumed to be equal *i.e.* $\lambda = 0.5$ in Equation (3.15). I investigated the effect of altering the weights of each component of SuCOS. The weights of chemical feature overlap and shape overlap were changed from $\lambda = 0$ (only chemical features) to $\lambda = 1$ (only shape) in increments of 0.05 and boxplots of the distributions of Pearson correlation coefficients of the cross-dockings were plotted for each (Figure 3.17). The

minimum IQR was found at a weight of $\lambda = 0.40$. This suggests that SuCOS may perform optimally with a slightly larger weighting of chemical feature overlap than shape overlap.

3.3.4 Computational Efficiency of MCS-RMSD, PLIF Similarity, and SuCOS

I compared the computational efficiency of all three measures. All benchmarks were performed on the same machine (running Fedora 28 and an Intel Core i7-6700 CPU running at 3.40 GHz with 32 GB of RAM). A single core was used for all benchmarks. The benchmarks were performed for Part I where values for MCS-RMSD, TvPLIF and SuCOS were calculated for each of the 284 pairs of larger and smaller ligands. For MCS-RMSD and SuCOS, one script was called to calculate the corresponding values for each pair. For TvPLIF, each pair required *arpeggio.py* to be executed twice: once to compute the interactions of the smaller ligand's complex and a second time for the larger ligand's complex. Then a further Python script was called to process the results and calculate the Tversky similarity. These two steps are referred to as "Arpeggio" and "post processing Arpeggio" in Figure 3.18, which shows the distributions of the execution times for MCS-RMSD, TvPLIF and SuCOS.

The median execution times for MCS-RMSD; Arpeggio & post-processing Arpeggio; and SuCOS were 0.26s; 1.14s & 1.60s; and 0.13s respectively. The median timing for TvPLIF is therefore $2 \times 1.14\text{s} + 1.60\text{s}$, or 3.88s. Calculation of TvPLIF is slowest; even a single *arpeggio.py* run takes on average longer than calculating MCS-RMSD or SuCOS. This can be attributed to Arpeggio having to process the protein structure and calculate the interatomic distances and angles for the protein-ligand complex. SuCOS was thus the fastest to calculate. In a practical scenario where docking screens are

performed on several million molecules, the time taken for these post-docking calculations will be magnified and hence, SuCOS may be even more appealing than MCS-RMSD and PLIF similarity, due to its lower computational cost and time.

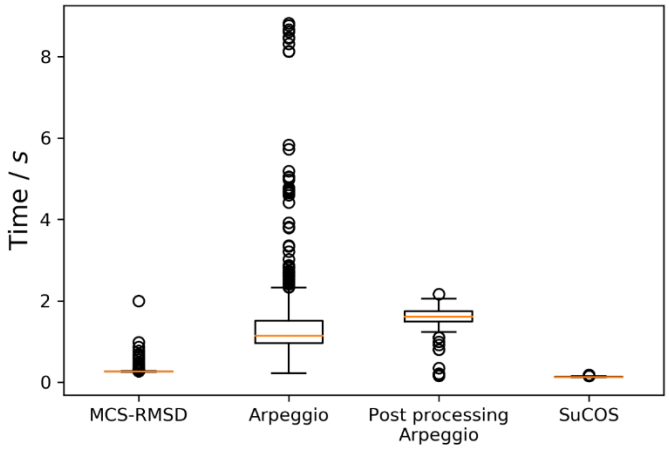


Figure 3.18. Execution times (wall clock) for calculation of MCS-RMSD, TvPLIF (Arpeggio and post-processing Arpeggio) and SuCOS for the 284 ligand pairs in Part I. For MCS-RMSD and SuCOS a single Python script was executed to calculate each of the 284 ligand pairs. For calculation of TvPLIF, this was split into running *arpeggio.py* twice for each pair: once for the smaller complex, and then a second time for the larger complex, and then running a post-processing script to calculate the Tversky similarity between the PLIFs of the smaller and larger complex.

3.4 Conclusion

I have compared three widely used metrics to investigate the conservation of binding mode of closely related smaller and larger ligands that bind to the same protein, namely: positional root mean square deviation (RMSD); protein-ligand interaction fingerprint (PLIF) similarity; and shape-chemical feature overlap. For the shape-chemical feature overlap metric, I have introduced SuCOS, an open-source RDKit-based metric. By investigating fragment-elaboration using the Malhotra-Karanicolas set, I have shown individual cases where each metric fails. RMSD is inappropriate to use when either molecule is pseudosymmetric, if multiple substructure matches are present, or there are bioisosteres. RMSD values depend heavily on the size of the molecules being compared, there is no upper limit, and it is difficult to define a

universal threshold for defining similarity. When comparing different molecules, multiple common substructure mismatches can sometimes occur, again invalidating the RMSD comparison.

PLIF similarity is heavily dependent on both the conformation of the protein and the ligand. Furthermore, it is possible for a ligand to have a good PLIF similarity score but have a very different pose than the comparator, as different groups within the molecule can retain the same interaction(s). This means the pose of the ligand needs to be visually inspected, making it more time consuming and less straightforward to interpret than ligand-centric metrics. In addition, there is no universally accepted definition of protein-ligand interactions, which can affect the results greatly and what PLIF similarity threshold to use. Also, equal importance is given to each interaction type. On the other hand, for different conformations of a given protein, PLIF similarity can capture which interactions are conserved, whereas the other two metrics cannot do so explicitly.

I have shown out of the three metrics, SuCOS obtains the best Pearson correlation coefficients when comparing poses of an elaborated molecule against its non-elaborated counterpart crystal structure and its true crystal pose. In a small number of cases with heterocyclic multi-ring systems, staggered conformations could result in poor SuCOS scores, but this could obviated by adjusting the weights. I have shown that SuCOS is useful as both a conservation of binding mode metric, and as a tool for structure-based virtual screening. It is implemented using the open-source cheminformatics API, RDKit, hence making it accessible and easy to build upon: for example, new pharmacophoric features could be included by adding the appropriate SMARTS expressions to the *BaseFeatures.fdef* file. It is available on https://github.com/susanhleung/SuCOS .

The objective of this chapter was to assess the best and most appropriate method of binding mode comparison, for the specific scenario after a fragment screening campaign, where poses of elaborated candidate molecules are compared to the poses of their experimental smaller counterparts. This could be a one-to-one pose comparison using SuCOS; however, fragment soaking campaigns can generate large amounts of structural data, so a related question would be how can we best use all of this structural data to inform us as to what to make next.

An example scenario where SuCOS could be applied is when the fragment hits are seen to bind to different sites within the binding pocket. Potential follow-up candidates could be proposed using a fragment linking strategy and the poses of the candidate linked fragment molecules could be ranked using SuCOS in a one-to-many calculation with respect to these reference fragment crystal structures. Another possible scenario could involve using SuCOS to cluster the binding modes of the fragment hits, to determine the optimal direction of fragment growth, or to prioritise candidate elaborated fragments that overlap with multiple clusters. I explore this idea of clustering binding modes using SuCOS in the next chapter, alongside the use of SuCOS to rescore docked binding poses to prioritise active molecules in virtual screening.

Chapter 4 Investigating the Ability of SuCOS to Classify Actives & Decoys in DUD-E

In the previous chapter, I introduced SuCOS, an open source RDKit based shape and chemical feature overlap score developed as an alternative to RMSD for measuring the conservation of binding mode for fragments and their elaborated counterparts. In this chapter, I investigated the use of SuCOS as a binary classifier to discriminate active molecules from decoy molecules in the DUD-E dataset. In the first study, I explored the use of a single crystallographic ligand of the corresponding DUD-E target as the reference molecule for the SuCOS calculations. I then explored the effect of varying the weights of SuCOS for each target. Several studies have already established that ligand shape-based measures can be effective in virtual screening when benchmarked using the DUD-E set; however, this first study builds the foundation and provides a baseline for the last study of this chapter, which explores data fusion methods.

In the last study of this chapter, I investigated the effect of using multiple molecules as reference molecules for the SuCOS calculation and the effect of two group fusion methods to combine the resulting multiple SuCOS values. To the best of my knowledge, no study has investigated whether there is a superior group fusion method when there are multiple reference protein-fragment structures. This last study aligns with my interests of how to use the maximum amount of information from a fragment screening campaign so that the most promising follow-up candidates are prioritised.

4.1 Introduction

4.1.1 DUD-E dataset

The Directory of Useful Decoys, Enhanced (DUD-E), is a set of 102 diverse protein targets, each of which has a set of experimentally determined active compounds along with property-matched decoys (inactive) compounds (Mysinger et al., 2012). It is a widely-used benchmarking set to compare structure-based virtual screening methods. The 102 targets can be classified into eight broad target classes: 26 kinases, 15 proteases, 11 nuclear hormone receptors, 5 G-protein-coupled receptors (GPCRs), 5 miscellaneous, 2 ion channels, 2 cytochromes, and 36 other enzymes. Mysinger *et al.* collected the actives from ChEMBL and clustered by ligand scaffold to ensure chemotype diversity. Mysinger *et al.* chose decoys by having similar physical properties *e.g.* molecular weight, to the actives, but different topology *e.g.* ECFP4 fingerprints.

However, Chen *et al.* recently reported hidden bias within the DUD-E dataset, in terms of analogue and decoy bias (Chen et al., 2019). For example, analogue bias within the actives can exist within each target and also across different targets. They speculated that the analogue bias for actives of a particular target may arise from a common scaffold, which machine learning models are able to recognise and thus the models learn from the ligand alone, and not from physically meaningful binding patterns such as interactions formed between the ligand and protein as originally hoped. Moreover, they argued that decoy bias may arise from how Mysinger *et al.* originally selected decoys for each DUD-E target; in order to decrease the false decoy rate, they selected molecules that were the most dissimilar by topological fingerprint for each active. Thus

meaning that the decoys may be easily distinguishable by ligand features alone. They therefore advocated increased awareness of possible dataset bias before using such datasets for reporting the performance of deep learning models.

4.1.2 Measuring the Performance of Classification Models

A common characteristic that we wish to predict in candidate compounds is whether they are active or inactive. This is an example of binary classification where there are four possible outcomes for each classification. A true positive, TP, is when the model predicts a compound to be active and the compound is truly active. A true negative, TN, is when the model predicts a compound to be inactive and the compound is truly inactive. A false positive, FP, is when a compound is predicted to be active but the compound is actually inactive. Finally, a false negative, FN, is when a compound is predicted to be inactive but the compound is actually active. These four prediction outcomes form a confusion matrix, Figure 4.1, which can also be used to derive useful metrics such as the true positive rate, TPR, and the false positive rate, FPR.

		True condition	
		Compound label active	Compound label inactive
Predicted condition	Predicted compound label active	True Positive (TP)	False Positive (FP)
	Predicted compound label inactive	False Negative (FN)	True Negative (TN)
		True Positive Rate (TPR) $TPR = \frac{\sum True\ positive}{\sum Condition\ positive}$	False Positive Rate (FPR) $FPR = \frac{\sum False\ positive}{\sum Condition\ negative}$
		False Negative Rate (FNR) $FNR = \frac{\sum False\ negative}{\sum Condition\ positive}$	True Negative Rate (TNR) $TNR = \frac{\sum True\ negative}{\sum Condition\ negative}$

Figure 4.1. Confusion matrix for treating SuCOS as a binary classifier.

The Receiver Operating Characteristics or ROC curve plots the TPR against the FPR, while varying a classification threshold. The area under the curve (AUC) represents

how well the model can distinguish actives from inactives. A perfect classifier would have an AUC of 1 and a random classifier would have an AUC of 0.5, and is represented by a straight line from the origin (0,0) to the top right corner (1,1). The further the AUC value is from 0.5, the better the model is at classification.

An alternative to ROC AUC for measuring the performance of a binary classifier is to measure the enrichment factor. One problem with ROC AUCs is that it measures the performance of a binary classifier across the whole dataset, whereas virtual screening programs are often interested in prioritising only the top ranked compounds.

The enrichment factor at *x%* is defined as the ratio between the predicted hit rate within the top *x%* of ranked compounds and the random hit rate:

$$EF_{x\%} = \frac{Number\ of\ true\ actives\ at\ x\%}{Number\ of\ compounds\ at\ x\%} \div \frac{Number\ of\ actives}{Number\ of\ compounds} \qquad (4.1)$$

As the EF depends on the number of actives and inactives, it is not just a measure of the performance of a method but its value also depends on the composition of the dataset, so its values cannot be directly compared across different targets.

The ROC enrichment (Jain and Nicholls, 2008; Nicholls, 2008), RE, is the ratio of the TPR to the FPR, at a given FPR threshold. It is advantageous over the EF as it takes into account the fraction of inactives and not the fraction of all compounds.

In this chapter I investigated the use of SuCOS as a binary classifier to discriminate active molecules from decoys in the DUD-E dataset. Its ability was quantified by measuring the ROC AUC and the ROC enrichment.

4.1.3 Data Fusion

Data fusion is the method of combining multiple data sources to produce a more informative single output than the individual data sources (Willett, 2013).

In the field of molecular modelling, data fusion has been used for ensemble docking, where the data to fuse are the multiple docking scores from docking one compound to different protein conformations, and the question is how to define the consensus docking score for that compound (Bajusz et al., 2019). In cheminformatics, data fusion has been used for similarity searching in ligand-based virtual screening (Willett, 2013).

Similarity searching is based on the molecular *similarity property principle* where molecules that are structurally similar are assumed to have similar properties (Johnson and Maggiora, 1992). When the principle is applied to structure-based drug discovery, the conformations of previously known binders can been used as references in a 3D similarity-based search.

Data fusion methods can be divided into two main approaches: *similarity fusion* and *group fusion* (Chen et al., 2010; Willett, 2013). Similarity fusion involves using different similarity methods against one reference molecule. Group fusion involves using a single similarity method but against multiple reference structures. The final step in both is to use a *fusion rule* to combine the multiple scores to give a final ranking.

In this chapter, I looked at using group fusion with SuCOS, where the reference database consists of X-ray crystal structures of different fragments bound to the same protein, and the query dataset is the DUD-E actives and decoys for a particular target (Section 4.3.3). A 3D conformation for each DUD-E active and decoy molecule is required for comparison using SuCOS. If n is the number of reference protein-fragment

X-ray structures, and *N* is the number of combined actives and decoys *i.e.* query molecules, the problem can be written as follows:

```
For i=1 to n,
     For j=1 to N,
          Compute the SuCOS similarity SuCOS(dij), between the
          ith reference and the jth query molecule
For j=1 to N,
     Using fusion rule to combine n SuCOS similarities for the
     jth query molecule to give a combined score.
```

A *fusion rule* combines multiple outputs *e.g.* similarity scores, into a single score. Many different fusion rules have been previously proposed, and the most common include the minimum, maximum, various types of averages, and the Euclidean fusion rule (Bajusz et al., 2019; Willett, 2013). The maximum, MAX, rule is calculated as,

$$MAX\_SIM = max\{sim_1(d_j), sim_2(d_j), \ldots sim_n(d_j)\} \tag{4.2}$$

Where d_j is the j^{th} query database molecule and sim_1, sim_2,….sim_n are the different methods of calculating similarities to the reference structure, for the case of similarity fusion, or similarities to multiple reference structures, for the case of group fusion. An alternative rule is the arithmetic, SUM, and is defined as,

$$SUM\_SIM = \sum_{x=1}^{n} sim_x(d_j) \tag{4.3}$$

There are many other fusion rules and I refer the reader to a comprehensive study conducted by Chen *et al.* that compares fifteen different fusion rules in similarity-based virtual screening (Chen et al., 2010). These fusion rules are not only applicable to raw similarity scores, but also to rankings.

4.1.1 Prior State of the Art and Chapter Aims

In this first study of this chapter, I investigated using SuCOS to rescore docked poses of the DUD-E dataset and compared its performance to the native docking score. In this study, I used the single ligand of the corresponding DUD-E target as the reference to rescore with SuCOS.

Previously, several studies have reported the use of 3D shape based methods to improve the enrichment of actives in virtual screening (Kumar and Zhang, 2018; Nicholls et al., 2010) (also see Section 1.2.4.4). Moreover, some of these have used DUD-E as the benchmark dataset, including Schreyer and Blundell's study with the Ultrafast Shape Recognition with CREDO Atom Types (USRCAT) (Schreyer and Blundell, 2012). USRCAT builds on the previous Ultrafast Shape Recognition (USR) approach, which is a moment-based method that relies on the relative positions of atoms in the molecule to describe the shape of the molecule. Whilst, the USR approach treats all atom types the same, USRCAT includes pharmacophoric information. Using the DUD-E dataset, they showed that USRCAT performed better than the original USR algorithm as it showed better enrichment factors.

Another example of a ligand shape-based method that used the DUD-E dataset as a benchmark was reported by Kearnes and Pande, who investigated decomposing the color force field of ROCS to improve the performance of virtual screening (Kearnes and Pande, 2016). They trained machine learning models that used the decomposed features of ROCS and showed that these trained models outperformed their baseline, which used standard ROCS with the equal weighting in TanimotoCombo, on the DUD-E and MUV dataset. To generate the ligand conformers for library molecules, they used OMEGA (Hawkins and Nicholls, 2012) and then used OpenEye's

OEBestOverlay (OpenEye Shape Toolkit, OpenEye Scientific Software) to align the reference and query conformer. They trained the models using a single reference molecule and five-fold cross validation and measured the performance of the models using ROC AUC and ROC enrichment.

With regards to these previous studies, the first part of this chapter that involves the investigation of SuCOS for rescoring of docked poses is only novel in the sense of the SuCOS method. However, by performing the benchmark with SuCOS, it enables the comparison between my tool and others. It also sets the foundation and baseline for the studies presented later in this chapter, in which I investigate the case when multiple different fragment-same protein crystal structures are available to use as references and the ideas of group fusion are applicable. For each query database molecule, the SuCOS value can be calculated to each of the *n* fragment-protein crystal structures; however, what is the best way to combine the scores? I am specifically interested in the situation when the candidate molecule is an elaborated fragment that may explore larger regions within the binding pocket than any one of the single reference fragments. Hence, I hypothesised that the SUM fusion rule may be more advantageous, as it can use all available information and the combination of scores of reference fragments that occupy slightly different areas within the binding pocket. I explored the use of the MAX method, and the SUM method, which I refer to as the 'cumulative method', or 'Cum', in this chapter.

This work involving group fusion methods and SuCOS builds on a previous related study reported by Drwal *et al.* who investigated whether fragments and crystallisation additives bind similarly to their drug-like counterparts. In their investigation, they used four protein targets that were well represented in the Protein Data Bank. They concluded that fragments have highly similar interactions with the protein compared to

their drug-like counterparts and that protein-additive complexes can also recapitulate interactions seen in the protein-drug-like complexes. In part of their study, they showed that use of the binding mode information of a molecule class *e.g.* additive, fragment or drug-like molecule, could be used to improve docking pose predictions for another molecule class. They investigated this rescoring using similarity by ROCS or interaction fingerprints (IFP) and for each they also investigated two scoring schemes: (i) maximal, where the maximum ROCS or IFP similarityto the all the molecules of a class was taken as the score for the docking pose, and (ii) consensus, where the docked pose was scored against a consensus IFP or consensus shape of a molecule class. However, Drwal *et al.* did not comment or conclude on whether the maximal or consensus score was more appropriate. In the studies presented in this chapter, one of the aims was to investigate if there is a superior group fusion method, MAX or SUM, given multiple reference protein-fragment structures and whether clustering the protein-fragment structures by binding mode to make a non-redundant set of reference protein-fragment structures would provide any benefit.

4.2 Methods

4.2.1 Using SuCOS for Virtual Screening

The DUD-E dataset (Mysinger et al., 2012) was used to assess the ability of SuCOS to discriminate actives from decoy molecules. I used the publicly-available predicted binding modes of the actives and decoys (Ragoza et al., 2017; Imrie et al., 2018; DUD-E docked poses, 2017), which were generated using the smina fork (Koes et al., 2013) of AutoDock Vina (Trott and Olson, 2010). Some molecules could not be parsed by RDKit (see Appendix Table B-1).

In Section 4.3.1, SuCOS values were calculated for all docked poses of the actives and decoys with respect to the crystallographic ligand of the corresponding DUD-E target. A single SuCOS value was kept, which was the maximum achieved out of all the docked poses for each ligand; similarly for the best AutoDock Vina score. Ranking by SuCOS was compared with ranking by the AutoDock Vina score using the area under the curve (AUC) of the receiver operating characteristic (ROC) curves, calculated for each of the 102 DUD-E targets. To quantify early enrichment, the ROC enrichment (Nicholls, 2008; Jain and Nicholls, 2008), RE, at 0.5%, 1%, 2%, 5%, was also calculated.

4.2.2 Preparation of Datasets

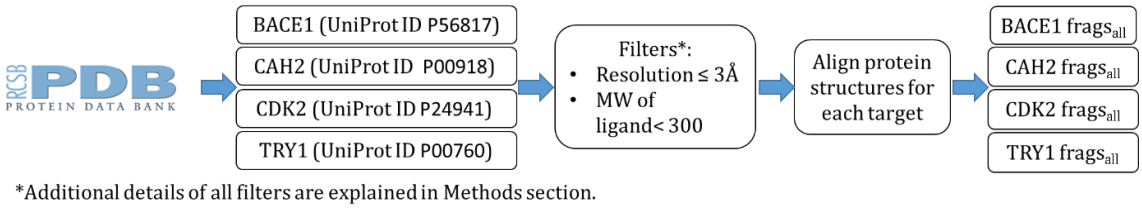


Figure 4.2. Workflow for the preparation of datasets for investigating data fusion with SuCOS and the DUD-E dataset. This procedure was repeated for BACE1, CAH2, CDK2 and TRY1.

Four protein targets were used in the investigation of group fusion methods for combining multiple fragment-protein structures to use as references for virtual screening (Section 4.3.3). The four targets, Beta-1-secretase (BACE1, UniProt ID P56817), Carbonic Anhydrase II (CAH2, UniProt ID P00918), Cyclin-dependent kinase 2 (CDK2, UniProt ID P24941), Bovine trypsin (TRY1, UniProt ID P00760), were chosen as they each have a large amount of structural data in the PDB, particularly with respect to fragments (Murray et al., 2007), and are targets of high pharmaceutical interest. They are also targets in the DUD-E dataset and so there are a set of actives and decoys assigned to each.

For each target, the PDB was filtered by the target's UniProt ID and only crystal structures containing a ligand with molecular weight ≤ 300, and resolution ≤ 3Å, were kept (Figure 4.2). PDB entries were not included if they only contained common crystallographic additives (listed in Appendix B.1). If multiple PDBs have the same three-letter ligand identifier then the one with the best resolution was kept. Also, PDB structures were discarded if the same ligand was present in the DUD-E active or decoy set for that target.

Each PDB structure was downloaded as the biological assembly and aligned to their respective DUD-E receptor file using the backbone atoms of the proteins and PyMOL's *align* function. The ligands in the aligned structures were saved as SDF files. To eliminate ligands that bind to an allosteric site, the centroid of each ligand was computed using RDKit's *ComputeCentroid* function and those with a centroid greater than 15 Å away from the centroid of the respective reference DUD-E ligand were not kept. The final list of protein-fragment PDB structures for each target, including the three-letter ligand identifiers, are listed in Appendix Table B-7 to Table B-10 and their corresponding fragments will be referred to as $frags_{all}$. The fragments were then clustered by firstly calculating the Tanimoto SuCOS between all fragment SDF pairs to generate a similarity matrix, *S*.

4.2.3 Tanimoto SuCOS

In the previous chapter, I discussed the use of SuCOS for comparing poses between a larger query molecule against a smaller reference molecule. The calculation was therefore asymmetric and the SuCOS value was dependent on which molecule was the reference and which was the query. In this chapter, I wanted to compute the binding pose similarity between poses of different fragments bound to the same protein, where

the shape and features of the reference and query molecule have equal importance and therefore I developed a symmetric SuCOS function, which I term the Tanimoto SuCOS.

The Tanimoto SuCOS is calculated in the same way as the original SuCOS but modifications were made to the chemical feature score and shape overlap score. For the Tanimoto chemical feature score, the Tanimoto coefficient was calculated by obtaining the number of chemical features in common between the two molecules, *c*, calculated by the *ScoreFeats* function between the two molecules, and the number of chemical features separately in each molecule, *a* and *b*, using the *GetNumFeatures* function. In Section 3.2.8, SuCOS was calculated by normalising *c* by the number of features in the smaller molecule; conversely, Tanimoto SuCOS normalises *c* by the union of the features of the two molecules, $a + b - c$. Thus using *a*, *b* and *c*, the Tanimoto similarity can be calculated using Equation (1.1). For the Tanimoto shape overlap calculation, RDKit's *ShapeTanimotoDis* was used to calculate the shape Tanimoto distance and then the shape Tanimoto similarity was calculated by $1 - ShapeTanimotoDis$.

4.2.4 Clustering of Fragments

A similarity matrix, S, was obtained by calculating the symmetric Tanimoto SuCOS between every combination of fragment pairs. The distance matrix, that is $1 - S$, was used to cluster the ligands by shape and chemical feature overlap using the *linkage* function, from *scipy.cluster.hierarchy,* with the *average* method. The clustering method is a bottom-up approach, where each observation starts off in its own cluster and after each iteration, the pairs of clusters that are nearest to each other are merged, and the distance matrix is updated. The *average* method calculates the distances by taking the mean distance between elements of each cluster *u* and *v*:

$$d(u,v) = \sum_{ij} \frac{d(u_i, v_j)}{(|u|.|v|)} \tag{4.4}$$

In Section 4.3.3, I investigated generating clusters using a threshold of t=0.8, where t is the Tanimoto SuCOS distance threshold, and then later looked at the effect of lowering the threshold to t=0.6 to generate more clusters. The representative fragment for each cluster was obtained by taking the fragment with the lowest value for the sum of distances between the other ligands in its cluster. The representative fragments after clustering will be referred to as $frags_{clustered}$ (Figure 4.3).

4.2.5 Comparison of Data Fusion Methods

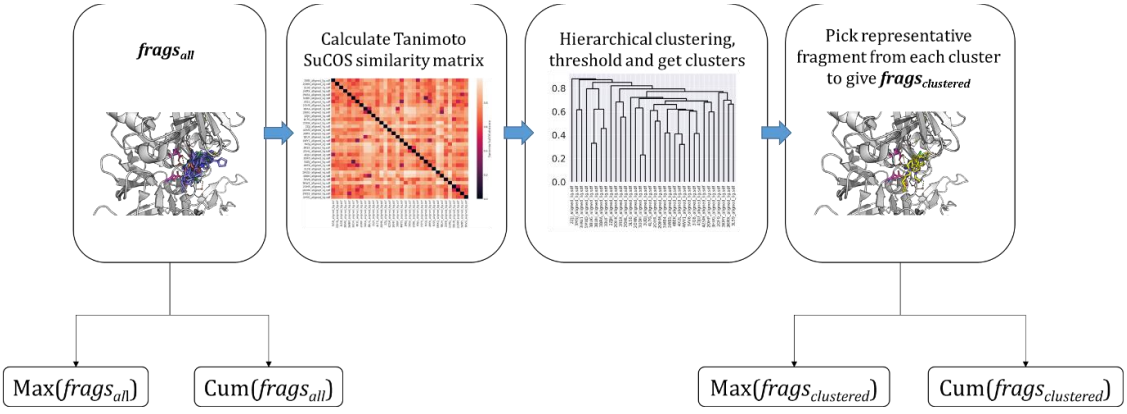


Figure 4.3. Summary of how all the fragments for a target, $frags_{all}$, were clustered to give $frags_{clustered}$. The two fragment datasets were then used to investigate two different group fusion methods, *Max* and *Cum*.

For each target, the compiled list of fragments can be used in several ways. I investigated two different data fusion methods: max and cumulative, based on equations (4.2) and (4.3) respectively. These two data fusion methods can be used with the whole list of compiled fragments for each target, $frags_{all}$, or, with the representative fragments of the clustered list, $frags_{clustered}$ (Figure 4.3 and Table 4-1).

For $frags_{all}$, SuCOS values were calculated for all docked poses of each active and decoy molecule with respect to each of the fragments in $frags_{all}$. For the max calculation, a single SuCOS value was kept for each active or decoy, which was the maximum achieved out of all the docked poses to each fragment *e.g.* in the case of BACE1 where there were 34 reference fragment molecules, if a docked active/decoy had 9 docked poses, then 306 (9x34) SuCOS values would be calculated and the maximum would be kept for that active/decoy. For the cumulative calculation, for each pose, the SuCOS score to all 34 fragments in $frags_{all}$ would be summed, and the pose with the largest sum would be kept for that active/decoy.

Similarly, the max and cumulative scores were calculated in the same way with $frags_{clustered}$, *i.e.* the representative fragments of the clustered list. For example, for the case of BACE1, instead of using all 34 reference fragments in $frags_{all}$, only the 5 clustered reference fragments in $frags_{clustered}$ were used.

Reference dataset	Data fusion method	Notation
$frags_{all}$	Max SuCOS	Max($frags_{all}$)
	Cumulative SuCOS	Cum($frags_{all}$)
$frags_{clustered}$	Max SuCOS	Max($frags_{clustered}$)
	Cumulative SuCOS	Cum($frags_{clustered}$)

Table 4-1. The different data fusion methods investigated in this chapter, alongside their notation.

4.3 Results and Discussion

4.3.1 Comparison of the Native AutoDock Vina Score with Rescoring Using SuCOS for Virtual Screening Using the DUD-E Dataset

As discussed in the introduction to this chapter (Section 4.1.1), several studies have already established that ligand shape-based measures can be effective in virtual

screening when benchmarked using the DUD-E set (Nicholls et al., 2010; Kearnes and Pande, 2016; Wang et al., 2019).

To investigate the effectiveness of SuCOS in virtual screening, I compared SuCOS-rescored AutoDock Vina docked poses with the original AutoDock Vina score for their ability to rank actives above decoys in the DUD-E dataset (Figure 4.4).

Metric	AutoDock Vina	SuCOS
AUC ROC	0.724	0.777
$RE_{0.5\%}$	16.313	48.908
$RE_{1\%}$	11.393	27.640
$RE_{2\%}$	8.011	16.046
$RE_{5\%}$	5.218	8.121

Table 4-2. Comparison of the mean AUC ROC and ROC enrichment values across all DUD-E targets when ranked by the native AutoDock Vina scoring function and SuCOS.

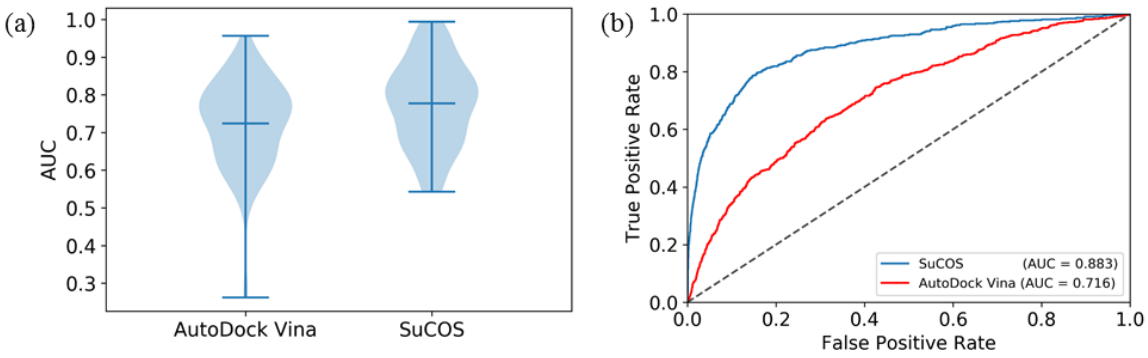


Figure 4.4. (a) The distribution of ROC AUCs for the 102 targets of the DUD-E dataset. The two violin plots show the different distributions obtained when scoring by the native AutoDock Vina scoring function versus rescoring the same pose with SuCOS against the representative ligand for that DUD-E target. For each plot, the maximum, minimum and mean are shown by the horizontal lines. (b) An example ROC plot: rescoring by SuCOS achieves a better AUC (0.883) than scoring with the AutoDock Vina score for HIV-1 protease (0.716). The dotted line represents the performance of a random classifier.

Rescoring by SuCOS achieved a mean AUC across all targets of 0.777, whereas scoring the same poses by the native AutoDock Vina scoring function achieved a corresponding value of 0.724 (Table 4-2 and Appendix Table B-2 for all targets). The median ROC AUC for SuCOS across all targets was 0.784 and is comparable to the median value of 0.749 found by a related previous study by Kearnes and Pande when they used ROCS TverskyCombo, without any machine learnt weights (Kearnes and Pande, 2016). However, one must take caution when comparing the value SuCOS achieved versus

theirs, as they did not include 12 out of the 102 targets, due to problems with some crystal ligands failing in the OMEGA conformer generation step. Even so, excluding these 12 targets from the SuCOS results, results in minimal change to the median ROC AUC (0.779). They also did not perform docking but generated conformers of the DUD-E actives and decoys using OMEGA and used OpenEye's *OEBestOverlay* to perform the alignment of shape and color volumes between the query and reference conformers.

In 64/102 cases, rescoring with SuCOS performed better than the AutoDock Vina score. SuCOS performed particularly well in some cases, with AUCs greater than 0.85, for 27 targets (FA7, TGFR1, PNPH, ADA, CAH2, MMP13, MAPK2, NRAM, CXCR4, PARP1, HMDH, ADA17, COMT, MET, FPPS, LKHA4, TYSY, ADRB1, HIVPR, GRIK1, DEF, SAHH, XIAP, WEE1, UROK, PUR2, and THB). For example, SuCOS achieved an AUC of 0.883 for HIV-1 protease (Figure 4.4b). SuCOS also performed better than the native AutoDock Vina scoring function in terms of early enrichment, having a mean ROC enrichment factor at 1% across all targets, $RE_{1\%}$, of 27.640, compared to 11.393 for the AutoDock Vina score (see Table 4-2 and Appendix Table B-3 – Table B-6 for $RE_{0.5\%}$, $RE_{1\%}$, $RE_{2\%}$, and $RE_{5\%}$, for all targets).

4.3.2 Investigating Optimal Weightings for SuCOS Across the DUD-E Dataset

In Section 3.3.3, I looked at how the Pearson correlation coefficient of the cross-dockings varied with altering the weights of each component of SuCOS. Here I investigated the effect of varying the weights of each component of SuCOS on the ROC AUCs for each target in the DUD-E dataset.

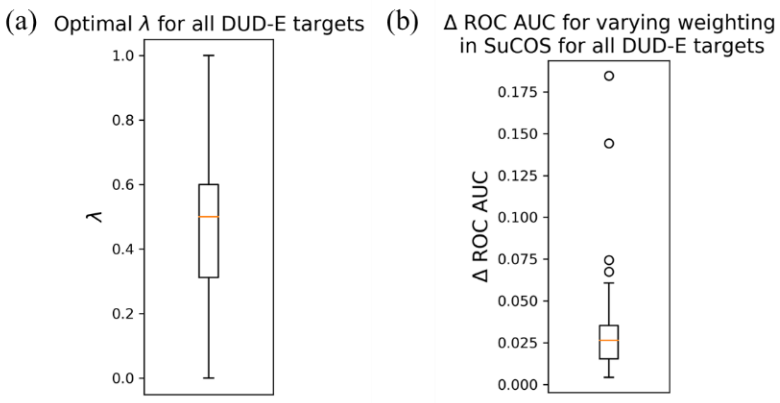


Figure 4.5. Investigating the optimal weights for SuCOS for all the DUD-E targets. (a) The value of λ that achieved the maximum ROC AUC was taken for each target and plotted as a box plot. (b) The Δ ROC AUC was calculated by taking the difference between the maximum ROC AUC and the minimum ROC AUC when varying λ for each DUD-E target.

For each target, the weights of the chemical feature overlap term and the shape overlap term were explored by varying λ (Equation 3.15) from zero to one in increments of 0.05, and the value of SuCOS was recalculated for each pose of the actives and decoys. As before (Section 4.3.1), the pose with the maximum SuCOS value was used from each docked ligand. Across all targets, the median optimal value of SuCOS weighting was $\lambda = 0.5$ (Figure 4.5a). The difference between the maximum ROC AUC and the minimum ROC AUC when varying λ for each DUD-E target is represented as *Δ ROC AUC* in Figure 4.5b. For most targets the *Δ ROC AUC* is small (80% of the targets had a *Δ ROC AUC* < 0.04); hence for these targets, the ROC AUC is robust to changes in the weighting of SuCOS. The effect of varying the SuCOS weighting on individual ROC AUCs for each target is shown in Appendix Figure B.1.

Examples of targets that favoured a higher shape weight include FA10, PGH1 and HIVINT. Upon inspection of these targets, it is not immediately obvious why this is; hence, more detailed investigation may be necessary. Conversely, an example of a target which disfavoured a higher weight of shape includes CAH2, where the ROC value has maximum of 0.865 at λ=0.5 and remains approximately the same but rapidly

drops with increasing proportion of shape, until it reaches 0.680 when λ=1, *i.e.* when SuCOS would be all shape and no chemical features. This could be explained by the dominance of the sulfonamide group in the DUD-E actives for CAH2, which have the same binding mode due to their chelation of the zinc ion and interaction with THR199 (discussed further in Section 4.3.3.3).

4.3.3 Investigating Group fusion methods for Combining Multiple Fragment-Protein Structures to Use as References in Virtual Screening

Target Name	Abbreviation	Target class	UniProt code	Number of fragment PDB structures	Number of fragment clusters (t=0.8)
Beta-secretase 1	BACE1	Protease	P56817	34	5
Carbonic anhydrase II	CAH2	Other Enzymes	P00918	186	11
Cyclin-dependent kinase 2	CDK2	Kinase	P24941	90	5
Trypsin I	TRY1	Protease	P00760	72	4

Table 4-3. The targets used to investigate different group fusion methods for combining multiple fragment-protein crystal structures to use as reference structures in virtual screening.

Following an initial fragment screen, the fragment hits may be seen to bind to numerous sub-pockets and/or have different binding modes. Each fragment-hit can be used as a reference molecule to search for larger follow-up elaborated fragment candidates by using binding pose similarity *e.g.* SuCOS similarity. However, SuCOS is designed to compute the similarity between one reference and one query molecule. For multiple reference fragment hits a group fusion method is required. Moreover, fragments are smaller than most drug-like molecules, hence they tend to occupy smaller spaces within the binding site; thus it is not clear which group fusion method is best for prioritizing active molecules (Figure 4.6).

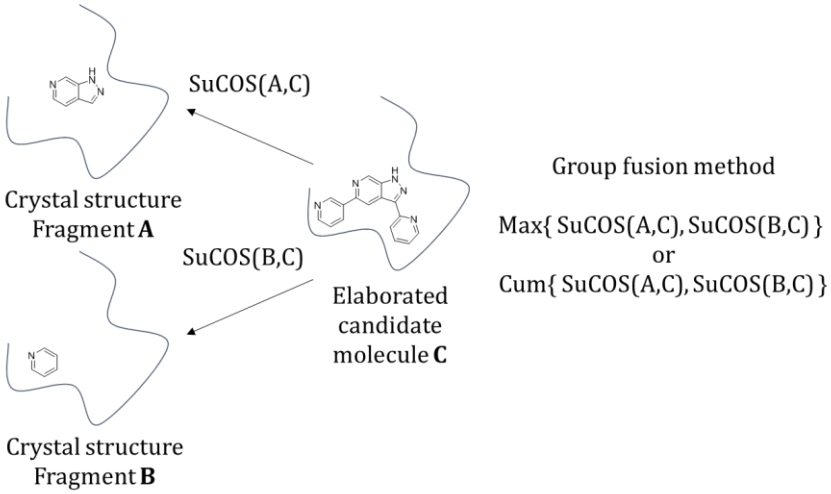


Figure 4.6. SuCOS similarity can be calculated between the poses of a single query and a single reference. Following a fragment screen, multiple fragment hits can be used to score each candidate molecule. A group fusion rule is required to combine the SuCOS similarities to each reference fragment-hit for each candidate. I investigated two group fusion methods: maximum and cumulative which I abbreviated to *max* and *cum* respectively.

I compiled datasets of protein-fragment crystal structures from the PDB for four DUD-E targets: BACE1, CAH2, CDK2, and TRY1, to explore two group fusion methods (see Figure 4.2). These targets were chosen as they each have a large amount of structural data in the PDB, particularly with respect to fragments (Drwal et al., 2017), and are targets of high pharmaceutical interest. The fragments within these datasets were used as references to score the corresponding DUD-E actives and decoys for each target. The distribution of the resolutions of the crystal structures for the four different targets are shown in Figure 4.7. The proportions of PDBs with a resolution greater than 2.5 Å are 18% (6/34), 3% (3/90), < 1% (1/186) and 0% (0/72) for BACE1, CDK2, CAH2 and TRY1 respectively. The relatively high proportion in the BACE1 dataset may be concerning; hence future work may involve repeating the analysis for BACE1 and leaving out the PDB structures with a resolution greater than 2.5 Å.

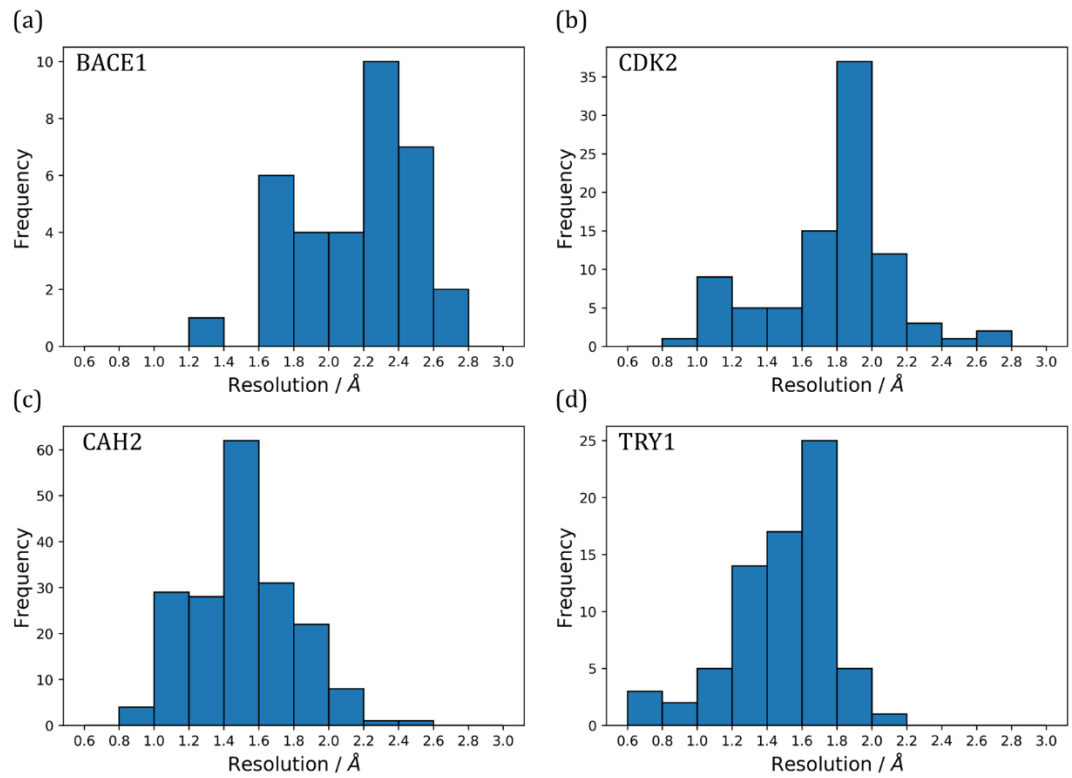


Figure 4.7. The distribution of the resolutions of the protein-fragment PDB structures for the four targets investigated in Chapter 4: (a) BACE1, (b) CDK2, (c) CAH2 and (d) TRY1.

The two different group fusion rules I investigated were maximum and cumulative. I also looked at their use with two different reference structure subsets for each target: using all the fragments in the dataset, $frags_{all}$, or using a clustered set, $frags_{clustered}$ (Figure 4.3). For the latter, I used Tanimoto SuCOS to cluster based on binding pose (Section 4.2.3-4.2.4). I compared the results against using the single DUD-E reference ligand and using each of the fragments individually as the reference. This also indicated the 'most useful' fragment and the molecular features that are important for obtaining high enrichment when ranking the DUD-E actives and decoys of that target. I evaluated the performance of the different methods by calculating the ROC AUC for each target set.

The hypothesis for clustering the fragment binding modes is to remove any bias towards similar fragment-hit binding modes. Depending on how the composition of the

screened fragment library is chosen, the library may be biased towards ligands with a certain molecular topology that all bind similarly. This would result in the possible over-representation of certain binding modes. Clustering by 3D conformation and picking a representative fragment from each cluster, means that only substantially different binding modes are retained. One of these binding modes clusters may give rise to compounds which are tighter binders but at the time of the fragment screen, if the protein target is not well studied, it is unknown which binding mode cluster this is. Thus, clustering of the fragment-hits should remove any over-represented binding modes and the results of the virtual screening should be less dependent on the composition of the originally screened fragment library.

In the comparison of the two methods of group fusion and whether or not to cluster the fragment data, I addressed the following questions:

- How does the distribution of ROC AUCs for individual fragments in $frags_{all}$ compare to the single DUD-E ligand?
- What are the features of the reference fragments that results in the highest ROC AUCs?
- What is best data fusion method to use – cumulative or max – for $frags_{all}$?
    - How does Max($frags_{all}$) and Cum($frags_{all}$) compare with the max and median fragment?
- What is the spread of ROC AUCs within each cluster?
- How does the representative fragments from $frags_{clustered}$ rank when used individually for ROC AUC?
- What is best data fusion method to use – cumulative or max – for $frags_{clustered}$?

    - How does Max($frags_{clustered}$) and Cum($frags_{clustered}$) compare with the max and median fragment within $frags_{clustered}$?
- Is there a best fusion method? And with which database $frags_{all}$ or $frags_{clustered}$?
    - Is there any method better than the best fragment?
- When lowering the binding pose clustering threshold to t=0.6 to create more clusters;
    - How does this affect the ROC spread within each cluster?
    - How does this affect the Max and Cum data fusion methods?

4.3.3.1 Beta-Secretase 1, BACE1

Beta-Secretase 1 (BACE1) is a transmembrane aspartyl protease that cleaves the amyloid precursor protein, APP, at the β-site and is involved in the formation of myelin sheaths in peripheral nerve cells. It is also implicated in Alzheimer's disease. Amyloid-β (Aβ) peptides are formed by two consecutive cleavages of APP, the first by BACE1 and the second by γ-secretase. Aβ is the primary constituent of the amyloid plaques found in the brains of Alzheimer's disease patients (Shimizu et al., 2008; Zhu et al., 2010).

BACE1 is part of the aspartyl protease family with two aspartic acids, ASP32 and ASP228, making up the catalytic dyad. These two acidic residues often form hydrogen-bonds with BACE1 inhibitors (Figure 4.8a).

A total of 34 structures of BACE1-fragment complexes were downloaded and prepared (see Methods Section 4.2.2) and the fragments clustered to yield five clusters using a cutoff of t=0.8 (for distance matrix and hierarchical cluster dendrogram see Appendix Figure B.2).

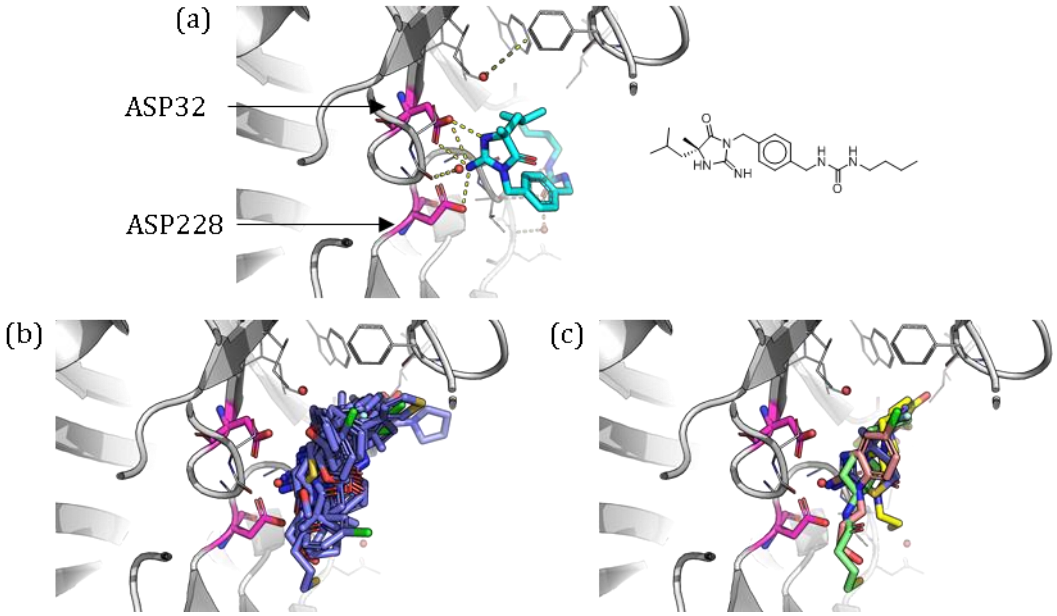


Figure 4.8. The different binding modes of the reference DUD-E ligand and the $frags_{all}$ and $frags_{clustered}$. The catalytic dyad, responsible for the cleavage of amyloid precursor protein, APP, is made up of two residues, ASP32 and ASP228, which are shown in pick sticks. (a) The DUD-E reference ligand (PDB ID 3L5D) is shown in cyan sticks. (b) For BACE1, $frags_{all}$ is comprised of 34 fragment crystal structures, which are shown in purple sticks. (c) The representative fragments from $frags_{clustered}$, in various coloured sticks, can be arguably seen to occupy different shape and pharmacophoric space within the binding site. This figure was produced using PyMOL (Schrödinger, LLC.).

Appendix Table B-7 gives the ROC AUC achieved when each of the 34 fragments was used as the reference and the distribution of ROC AUCs shown in Figure 4.9. The highest ROC AUC was 0.818 achieved by the fragment of PDB ID 3WB5. This is similar to the performance of using the DUD-E reference ligand as the reference. Inspection of their 3D conformations shows that both have a set of pharmacophoric features that are able to form interactions with the catalytic dyad in the hinge region (Figure 4.10). This pharmacophore may therefore be crucial in discriminating the actives from the decoys.

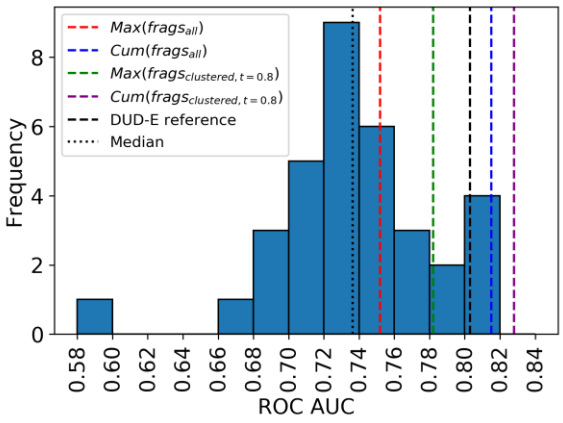


Figure 4.9. The distribution of ROC AUCs achieved when each of the 34 BACE1 fragments in $frags_{all}$ is used as the reference molecule when rescoring with SuCOS the docked poses of DUD-E BACE1 actives and decoys. The performance of the BACE1 DUD-E reference ligand along with the Max($frags_{all}$), Cum($frags_{all}$), Max($frags_{clustered,t=0.8}$), Cum($frags_{clustered,t=0.8}$) and the median of the distribution are shown by the vertical dotted lines.

BACE1	$Frags_{all}$		$Frags_{clustered,t=0.8}$		DUD-E reference ligand	Best fragment
	Max SuCOS	Cumulative SuCOS	Max SuCOS	Cumulative SuCOS		
	0.752	0.815	0.782	**0.828**	0.803	0.818

Table 4-4. Summary of the ROC AUCs achieved by the investigated group fusion methods for BACE1. For λ=0.5, Cum($frags_{clustered,t=0.8}$) performed the best, whilst Max($frags_{all}$) performed the worst, in terms of ROC AUC.

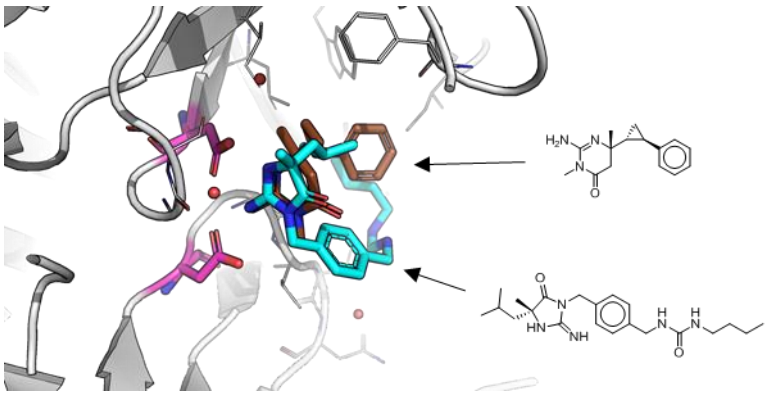


Figure 4.10. The DUD-E reference ligand for BACE1 (PDB ID 3L5D), in cyan sticks, has an oxoimidazolidine ring that forms similar interactions to the catalytic dyad residues, shown in magenta, compared to fragment of 3WB5, in brown sticks, which has a dihydropyrimidine ring that is able to mimic the oxoimidazolidine ring. This figure was produced using PyMOL (Schrödinger, LLC.).

The best of the individual fragments performed better than Max($frags_{all}$) and Cum($frags_{all}$) which had ROC AUCs of 0.752 and 0.815 respectively (Table 4-4). This suggests that for BACE1 it is better to have a good single reference molecule than to use a *Max* or *Cum* fusion of all the structural data from a fragment screen to score

candidate molecules. However, in reality for a target of interest that is not well studied, the fragment which is the 'best' reference will be unknown, unless there is a lot of activity data, as is the case for the DUD-E targets.

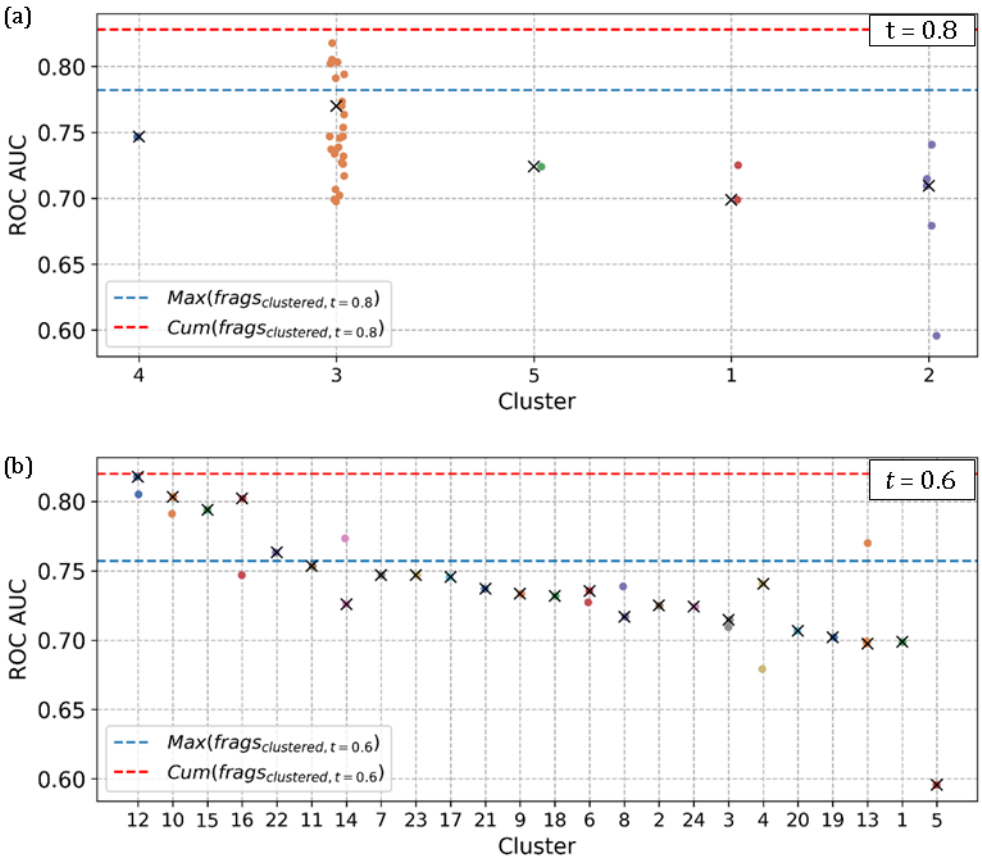


Figure 4.11. The distribution of ROC AUCs for each fragment, grouped by cluster. Hierarchical clustering was performed using (a) a cutoff of t=0.8, which formed 5 clusters, and (b) a cutoff of t=0.6, which formed 24 clusters. The black crosses signify the representative fragment in each cluster. The clusters are ordered from left to right by decreasing median ROC AUC achieved by the cluster.

Cum($frags_{clustered,\ t=0.8}$) achieved a ROC AUC of 0.828, which is better than using any of the individual fragments as a reference and also better than the other three data fusion methods Max($frags_{all}$), Cum($frags_{all}$) and Max($frags_{clustered,\ t=0.8}$) (Table 4-4). Also, interestingly the five fragments in this clustered set did not rank well in terms of individual ROC AUC; they have ranks of 30, 26, 7, 11 and 23 out of 34 and a best ROC AUC of 0.770 (Appendix Table B-7). The better performance of data fusion using Cum($frags_{clustered\ ,\ t=0.8}$) shows that the combination of these fragment structures acts as a better reference than any of the individual fragments. This may be due to different

binding mode subgroups in the DUD-E actives that score well in SuCOS to different representative fragments *i.e.* not one of the single representative fragments act as a good reference molecule for all DUD-E actives as the actives may contain a range of different binding modes.

Clustering the reference fragments with threshold t=0.8 gave rise to five clusters. The majority of fragments (26/34) belonged to cluster 3. In contrast to this highly populated cluster, cluster 4 and 5 only contained a single fragment, and hence can be interpreted as outliers in terms of having a different binding mode (Figure 4.12). Cluster 3 achieved the best ROC AUC in terms of each cluster's representative fragment. The representative fragment from each cluster is shown in Figure 4.8c; even though they occupy the same region in the binding pocket, they can be seen to have different shapes and pharmacophores.

The ROC AUCs for each fragment, grouped by cluster is shown in Figure 4.11a. Clusters 2 and 3 show a wide range of ROC AUCs. For example, cluster 3 has a range of ROC AUCs from 0.698 to 0.818. This range can be explained by the range of binding modes shown by the fragments within each cluster (Figure 4.12). A lower clustering threshold could be used to generate clusters with more distinctive ROC AUCs.

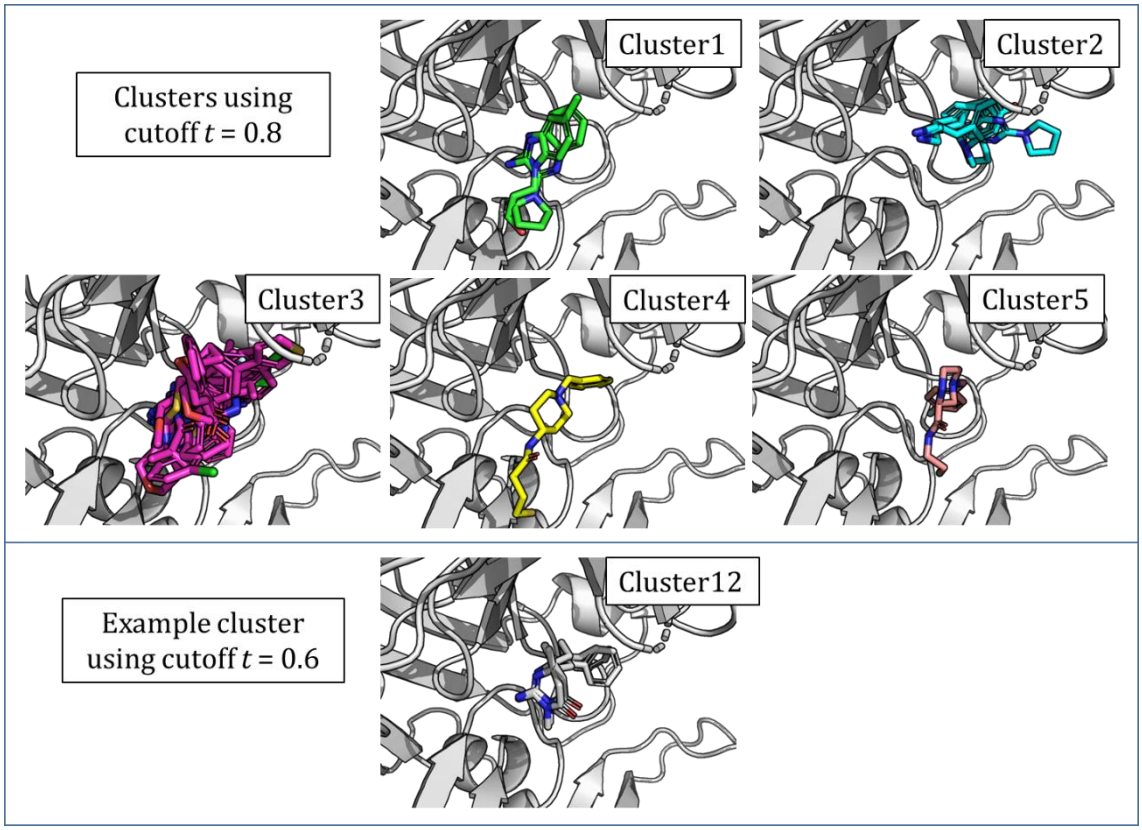


Figure 4.12. Illustration of the BACE1 fragment clusters resulting from the two different cutoffs, t=0.8 and t=0.6. Cluster 3 resulting from the less strict cutoff, t=0.8, shows a wide range of binding poses. Cluster 12 from the more strict cutoff, t=0.6, shows fragments with very similar binding modes. These two fragments (PDB IDs 3WB4 and 3WB5) were part of cluster 3 when clustered with threshold t=0.8.

To investigate this, the clustering threshold was lowered to t=0.6, which gave 24 clusters and resulted in the distribution of ROC AUCs shown in Figure 4.11b. Most of the clusters contained only one fragment structure. Visual inspection of the clusters show more similar binding modes within a cluster, which also leads to smaller ranges of ROC AUCs for a given cluster (for example, see cluster 12 shown in Figure 4.12). The ROC AUCs for Max($frags_{clustered}$) and Cum($frags_{clustered}$) were calculated again for these 24 clusters and values of 0.757 and 0.820 were obtained respectively. Compared to the corresponding results for the five clusters from threshold t=0.8, both decreased slightly.

4.3.3.2 Cyclin-Dependent Kinase, CDK2

Cyclin-dependent kinase 2 (CDK2) is a protein kinase involved in the regulation of the cell cycle and is most active in the G1 and S phase (Satyanarayana and Kaldis, 2009). Its active site is formed of a chain of 100 consecutive amino acids. CDK2 inhibitors typically target the ATP binding site and are largely classified into two broad types: type I and type II. Type I targets the ATP binding site in its active state, whereas type II inhibitors target CDK2 in its inactive, unbound state (Alexander et al., 2015).

Type I inhibitors typically form hydrogen bonds to the backbone of the residues in the hinge region, residues 81-83. Type II inhibitors are typically associated with an outward flip of the conserved DFG (ASP145, PHE146, GLY147) motif, and occupies the space generated by the outward flip.

Figure 4.13a shows an example of a type I inhibitor, which exemplifies a commonly seen kinase purine-mimetic pharmacophore where the ligand has a central hydrogen bond acceptor, situated in between two hydrogen bond donors, that make interactions to LEU83 and a bridging water molecule (Beattie et al., 2002). Purine-mimetic kinase inhibitors have been previously reported to have one to three of these hydrogen bond interactions (Legraverend et al., 2000).

A total of 90 CDK2 fragment-protein crystal structures were compiled and used as $frags_{all}$ (Figure 4.13b, Appendix Table B-8). Clustering with threshold t=0.8 resulted in five clusters (Figure 4.13c).

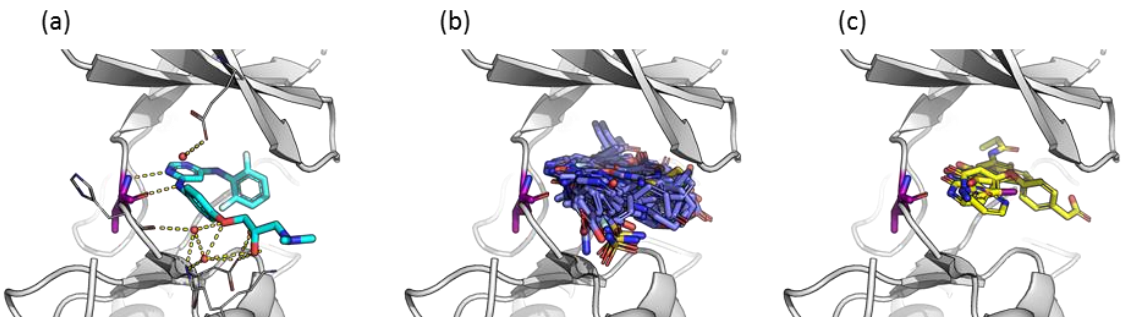


Figure 4.13. (a) The DUD-E reference ligand for CDK2 (PDB ID 1H00). The ligand is shown in cyan sticks and forms a hydrogen bond with the backbone carbonyl oxygen and backbone amide NH of hinge residue LEU83 shown in pink sticks. Interactions are also shown to bridging water molecules. (b) For CDK2, $frags_{all}$ is composed of 90 fragment crystal structures, which are shown in purple sticks. (c) $Frags_{clustered}$ is comprised of five representative fragments, which are shown in yellow sticks. This figure was produced using PyMOL (Schrödinger, LLC.).

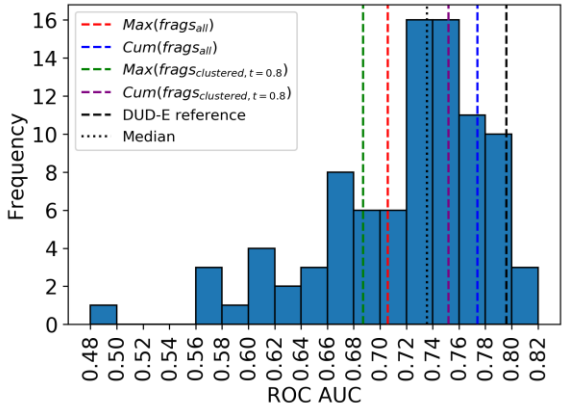


Figure 4.14. Distribution of ROC AUCs achieved when each of the 90 CDK2 fragments in $frags_{all}$ was used as the reference molecule to rescore with SuCOS the docked poses of DUD-E CDK2 actives and decoys. The performance of the CDK2 DUD-E reference ligand along with the Max($frags_{all}$), Cum($frags_{all}$), Max($frags_{clustered,t=0.8}$), Cum($frags_{clustered,t=0.8}$) and median of the distribution are also shown by the vertical dotted lines.

CDK2	$Frags_{all}$		$Frags_{clustered,t=0.8}$		DUD-E reference ligand	Best fragment
	Max SuCOS	Cumulative SuCOS	Max SuCOS	Cumulative SuCOS		
	0.706	0.774	0.687	0.752	0.796	**0.811**

Table 4-5. Summary of the ROC AUCs achieved by the investigated group fusion methods for CDK2. For λ=0.5, the single best fragment performed the best, whilst Max($frags_{clustered,t=0.8}$) performed the worst, in terms of ROC AUC.

The distribution of ROC AUCs for using each fragment as a reference is shown in Figure 4.14. The best fragment (PDB ID 1OIQ) achieved a ROC AUC of 0.811, which performed slightly better than using the single DUD-E ligand as the reference, with a ROC AUC of 0.796 (Table 4-5). Visual inspection of the best fragment and the DUD-E reference ligand, shows a very similar pharmacophore (Figure 4.15). This could

indicate that hydrogen bonding to the hinge region is crucial for discriminating the active ligands in the DUD-E dataset.

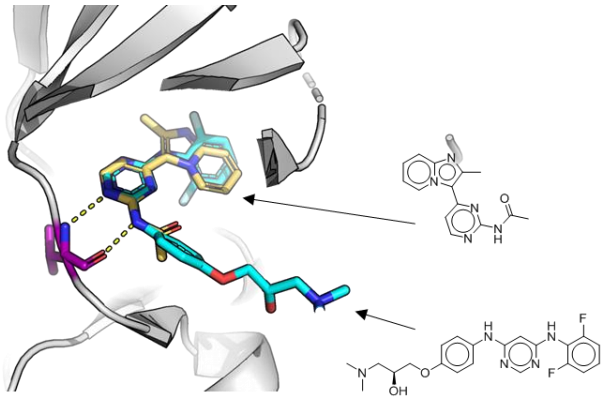


Figure 4.15. The best fragment (PDB ID 1OIQ), in yellow sticks, and the DUD-E reference ligand (PDB ID 1H00) for CDK2, in cyan sticks, show similar pharmacophoric features that can bind to the hinge residue (magenta sticks). This figure was produced using PyMOL (Schrödinger, LLC.).

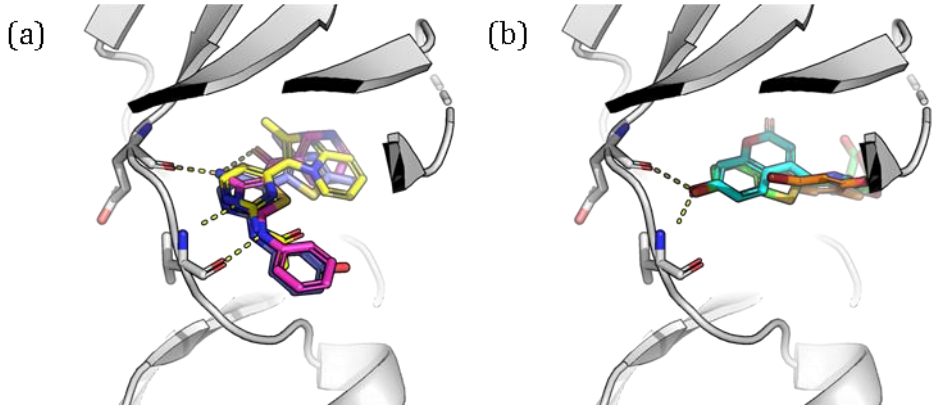


Figure 4.16. (a) The three best CDK2 reference fragments ranked by ROC AUC. They were used individually as references to rescore with SuCOS the DUD-E docked poses for CDK2. The best fragment is shown in yellow sticks. (b) The three worst fragments ranked by ROC AUC. Hinge binding residues 81 and 83 and are shown in white sticks. H-bonds are shown as yellow dashed lines. This figure was produced using PyMOL (Schrödinger, LLC.).

The second and third best scoring fragments, PDB IDs 1PXM and 3QTW, which have ROC AUCs of 0.805 and 0.801 respectively, show similarly a hydrogen bond donor and acceptor in the hinge region and are in the same cluster as the best fragment (Figure 4.16a). Conversely, the fragments with the worst ROC AUC: 6Q4D, 4D1X and 5ANG have ROC AUCs of 0.498, 0.577 and 0.578 respectively (Figure 4.16b). None have the required motifs for hydrogen bonding interactions shown by purine-mimetic inhibitors.

Scoring by Max($frags_{all}$) performs worse than the median, the best individual fragment and Cum($frags_{all}$), with ROC AUCs of 0.706, 0.736, 0.811 and 0.774 respectively (Figure 4.14).

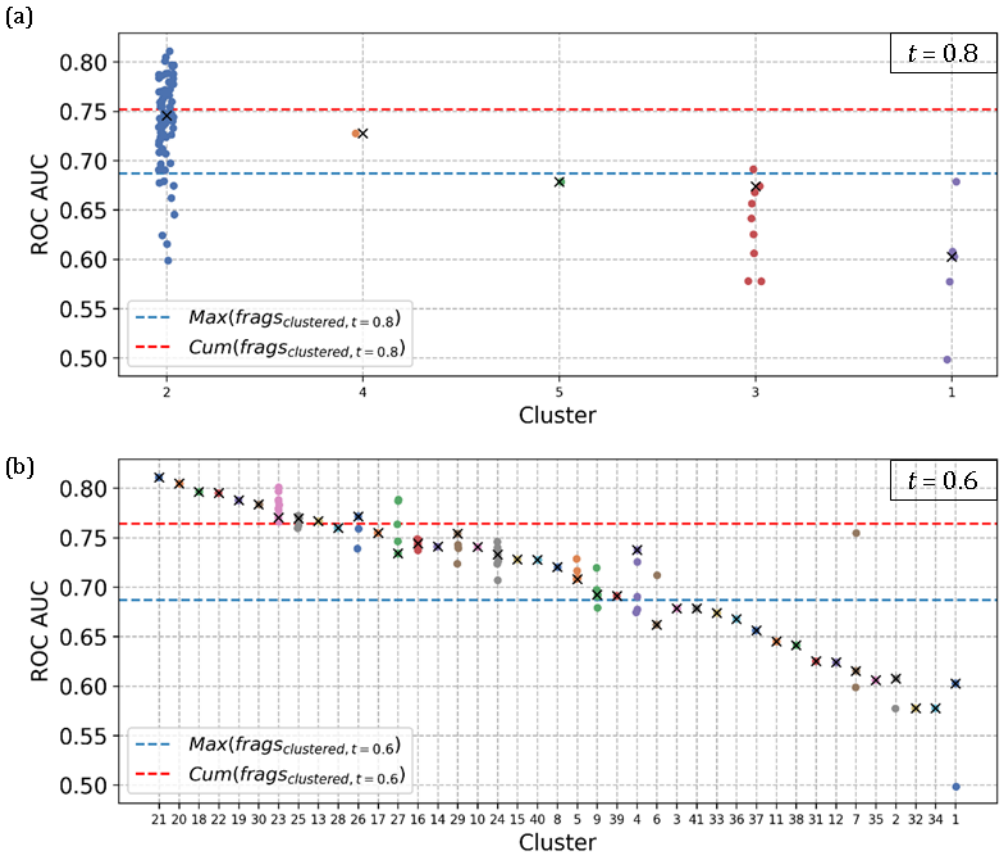


Figure 4.17. The distribution of ROC AUCs for each fragment, grouped by cluster. Hierarchical clustering was performed using (a) a cutoff of t=0.8, which formed 5 clusters, and (b) a cutoff of t=0.6, which formed 41 clusters. The black crosses signify the representative fragment in each cluster. The clusters are ordered from left to right by decreasing median ROC AUC achieved by the cluster.

Clustering the reference fragments with threshold t=0.8 produced five clusters and similarly to the previous investigation with BACE1, the majority of the fragment are contained within a single cluster, which is cluster 2 (Figure 4.17a). Like the previous BACE1 study, this overpopulated cluster contrasts with cluster 4 and cluster 5, which only contain one fragment and represent outliers with respect to fragment binding mode (Figure 4.18). Therefore, inclusion of these fragments as reference molecules when investigating group fusion methods may be disadvantageous as they may also be

outliers with respect to the binding modes of the DUD-E actives *i.e.* have different binding modes to the majority of the DUD-E actives.

The clustered fragments show a wide range of ROC AUCs for clusters that contain more than one fragment (Figure 4.17a). For example, ROC AUCs in cluster 2 range from 0.599 to 0.811 and upon visual inspection, there is a lot of variation in binding mode (Figure 4.18).

The five representative fragments in $frags_{clustered,t=0.8}$ are ranked 85, 34, 74, 51 and 71 out of 90 and have a max and median ROC AUC of 0.746 and 0.678 respectively (Appendix Table B-8). Cum($frags_{clustered,t=0.8}$) achieved a ROC AUC of 0.752, thus using the fragments in a cumulative fusion method had similar performance compared to using the best representative fragment within $frag_{clustered,t=0.8}$. Cum($frags_{clustered,t=0.8}$) also performed better than Max($frags_{clustered,t=0.8}$) which had a ROC AUC of 0.687.

However, using Cum($frags_{all}$) performed better still, compared to Cum($frags_{clustered}$). This is in contrast to the previous study with BACE1, where Cum($frags_{clustered}$) performed slighter better than Cum($frags_{all}$). This may be because $frags_{all}$ contains a set of fragments with a biased in binding mode *e.g.* most fragments are contained in cluster 2, which shows a common binding motif to the hinge region. Having a set of reference fragments that have a bias in binding mode may be advantageous as the CDK2 DUD-E actives may also have the same bias in binding mode. Further evidence of this bias in binding mode in the DUD-E actives can be seen in Figure 4.17a, where the fragments with the best ROC AUCs are all in cluster 2. Therefore using $frags_{clustered}$, which represents a set of reference fragments with non-redundant binding modes will not be advantageous. Whereas for BACE1, there may be less of a binding mode bias in the DUD-E actives.

Also the fragments in $frag_{clustered}$ have poor ranks and ROC AUCs, thus my clustering method and method for picking the representative may not be optimal. Potentially, creating more clusters may increase the ROC AUCs achieved by Max($frags_{clustered,t=0.8}$) and Cum($frags_{clustered,t=0.8}$).

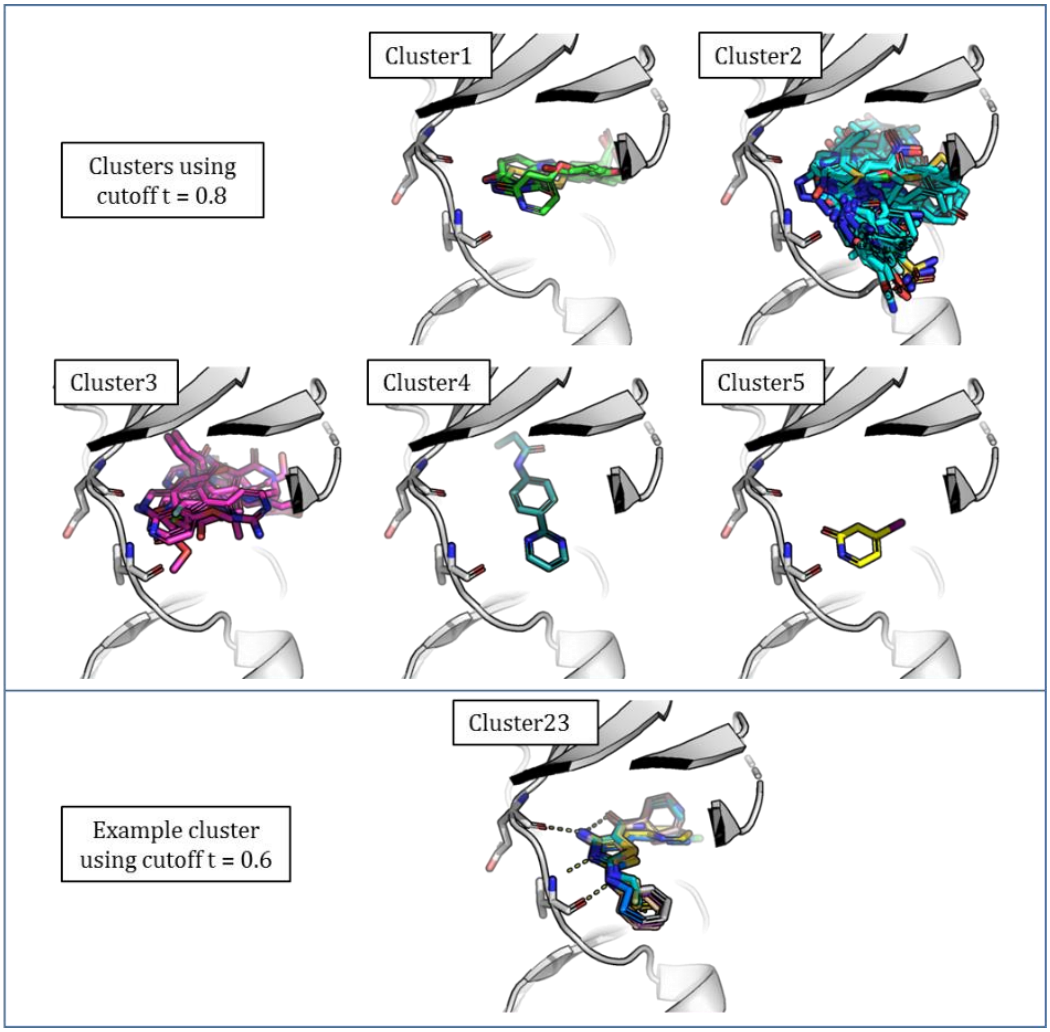


Figure 4.18. The CDK2 clusters resulting from two different cutoffs, t=0.8 and t=0.6. This figure was produced using PyMOL (Schrödinger, LLC.).

Max($frags_{all}$) and Max($frags_{clustered,t=0.8}$) showed worse ROC AUCs than the median of the ROC AUCs when the 90 fragments were used individually as reference molecules. A reason for this may be because the data fusion method is not only acting as a binary classifier but also as a method for choosing which docked pose to represent the active/decoy molecule. Computing SuCOS to all reference fragments and keeping the

maximum score could mean the wrong pose is chosen *i.e.* there may be a docked pose that is closer to the true crystal pose but some reference fragments may have better binding mode or SuCOS similarity to the wrong docked pose, which leads to the wrong docked pose being scored more favorably than the docked pose that is closer to the true crystal pose.

I investigated the effect of increasing the number of clusters, by decreasing the clustering threshold to t=0.6. This created 41 clusters and the ROC AUCs for Max($frags_{clustered,t=0.6}$) and Cum($frags_{clustered,t=0.6}$) were 0.687 and 0.764 respectively (Figure 4.17b). Increasing the number of clusters in this study with CDK2 improved the Cum($frags_{clustered}$) but Max($frags_{clustered}$) did not change. Visual inspection of the clusters shows they are also more well defined in 3D space and have narrower ranges of ROC AUCs (Figure 4.17). For example, Figure 4.18 shows cluster 23, which has nine fragments that show the common purine-mimetic pharmacophore and have a ROC AUC range of 0.766 to 0.801. However, many clusters have just one structure. It can also be seen that Cum($frags_{clustered}$) performed better than Max($frags_{clustered}$) for both thresholds.

4.3.3.3 Carbonic Anhydrase 2, CAH2

Carbonic anhydrase 2 is a metallo-enzyme that catalyses the reverse hydration of carbon dioxide. Abnormal levels or activities of this enzyme are associated with osteopetrosis and renal tubular acidosis (Shah et al., 2004). Its structure consists of a central beta-sheet surrounded by non-helical structures (Håkansson and Liljas, 1994). In the active site, the zinc ion is tetrahedrally coordinated by three histidine residues and a water or hydroxide ion. The binding site is between this zinc ion and hydrophobic residues of the beta-sheet.

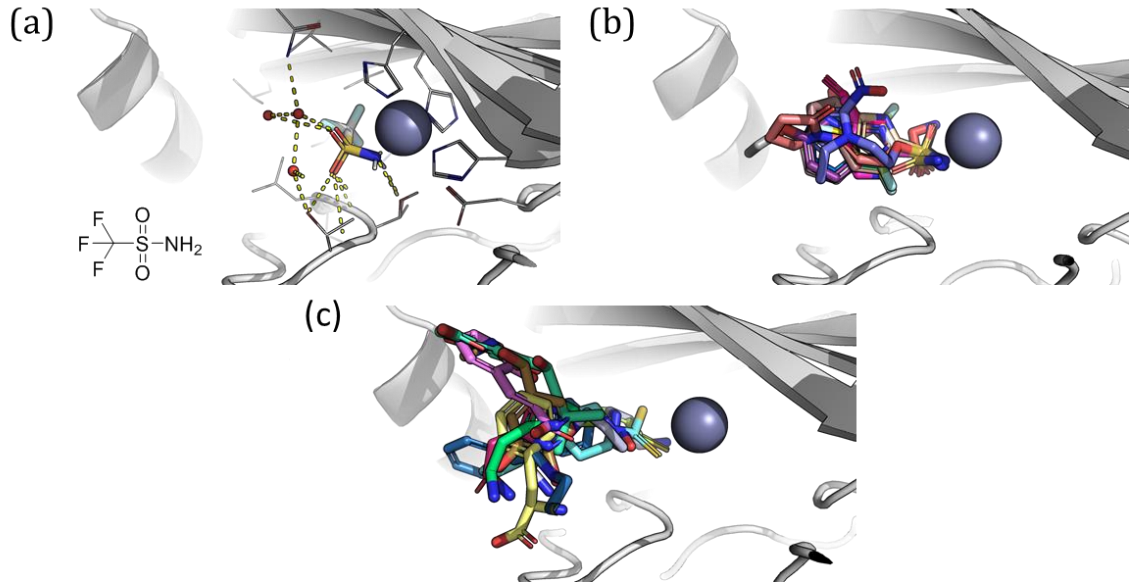


Figure 4.19. (a) Trifluoromethane sulfonamide is the DUD-E reference ligand for CAH2 (PDB ID 1BCD). It interacts with the Zn ion in the binding site, shown by the grey sphere, which is tetrahedrally coordinated by the ligand and the three histidine resides, shown in white sticks. (b) The top 20 reference fragments with the best ROC AUCs. All show the common sulfonamide binding to the Zn ion. (c) The 10 fragments with the worst ROC AUCs. This figure was produced using PyMOL (Schrödinger, LLC.).

The ligand for the CAH2 DUD-E reference structure (PDB ID 1BCD) is trifluoromethane sulfonamide (Figure 4.19a). The nitrogen of the ligand is bound to the zinc atom in the active site and the trifluoromethane end is bound to the hydrophobic part of the active site (Figure 4.19a).

Sulfonamides are one of the most important and common substructure in carbonic anhydrase inhibitors as they bind to the zinc ion in the binding site, providing an anchoring group. The majority of clinically approved carbonic anhydrase inhibitors contain this moiety (Alterio et al., 2012). The sulfonamide nitrogen is usually deprotonated when it binds to the Zn ion, and the sulfonamide group usually forms two hydrogen-bonds with THR199.

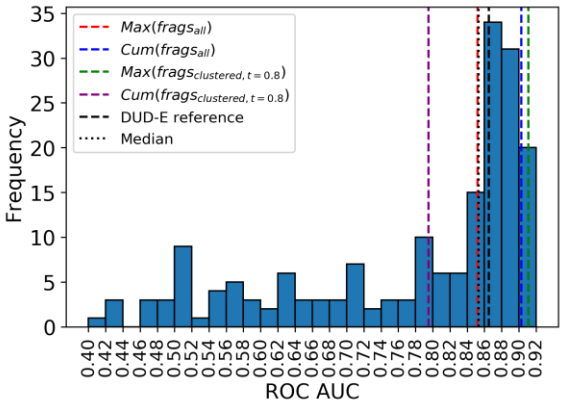


Figure 4.20. Distribution of ROC AUCs achieved when each of the 186 CAH2 fragments in *frags*$_{all}$ is used as the reference molecule to rescore with SuCOS the docked poses of DUD-E CAH2 actives and decoys. The performance of the CAH2 DUD-E reference ligand along with the Max(*frags*$_{all}$), Cum(*frags*$_{all}$), Max(*frags*$_{clustered,t=0.8}$), Cum(*frags*$_{clustered,t=0.8}$) and median of the distribution are also shown by the vertical dotted lines.

CAH2	*Frags*$_{all}$		*Frags*$_{clustered,t=0.8}$		DUD-E reference ligand	Best fragment
	Max SuCOS	Cumulative SuCOS	Max SuCOS	Cumulative SuCOS		
	0.852	0.903	0.911	0.795	0.865	**0.920**

Table 4-6. Summary of the ROC AUCs achieved by the investigated group fusion methods for CAH2. For λ=0.5, the single best fragment performed the best, whilst Cum(*frags*$_{clustered,t=0.8}$) performed the worst, in terms of ROC AUC.

A total of 186 fragment crystal CAH2 structures were downloaded and used in *frags*$_{all}$ (Table 4-3, Appendix Table B-9). The distribution of ROC AUCs for using each fragment as a reference is shown in Figure 4.20. Twenty of the fragments give rise to ROC AUCs greater than 0.9 and performed better than the DUD-E CAH2 reference ligand. All have the sulfonamide group that can bind to the Zn ion (Figure 4.19b). This contrasts with the 10 fragments with the worst ROC AUCs, which do not contain the sulfonamide group (Figure 4.19c).

Scoring by Max(*frags*$_{all}$) achieves a ROC AUC of 0.852 which is worse than best individual fragment and worse than Cum(*frags*$_{all}$), and similar to the median of all the ROC AUCs for all fragments (Table 4-6 and Figure 4.20). A possible reason for the poorer performance of Max(*frags*$_{all}$) compared to Cum(*frags*$_{all}$) could be because some

of the reference fragments leads to the incorrect rescoring of the docked poses as an irrelevant docked pose is chosen over a near native pose.

Clustering using a cutoff of t=0.8 resulted in eleven clusters (Figure 4.22), and again many of the clusters show a wide range of ROC AUCs (Figure 4.21). Similar to the previous two studies with BACE1 and CDK2, some clusters can be seen to be highly populated *e.g.* cluster 5, whereas other clusters are singletons. Further analysis is needed to conclude whether or not these singleton clusters contribute to the greater enrichment of DUD-E actives or whether they just add noise. Moreover, 19 of the 20 top ranked fragments were in cluster 5, which all show interaction with the Zn ion in the active site Figure 4.19b. Cluster 6 also contains fragments that achieved relatively high ROC AUCS and upon visual inspection, also show groups that interact with the Zn ion (Figure 4.22). Thus this indicates that there may be biased in the binding mode of the DUD-E actives that is somewhat mirrored by the biased in the binding modes of this reference fragment dataset.

The eleven fragments in $frags_{clustered,t=0.8}$ are ranked 10, 16, 122, 140, 165, 169, 170, 171, 180, 181 and 183 (Appendix Table B-9). Cum($frags_{clustered,t=0.8}$) performed worst out of the group fusion methods, with a ROC AUC of 0.795. This is inconsistent with the previous studies involving BACE1 and CDK2. This could be explained by the fragments in $frags_{clustered,t=0.8}$ having poor ranks and ROC AUCs; however, this was also the case for CDK2. Another explanation could be related to biased in binding modes in the DUD-E dataset for CAH2 actives. The idea of clustering was to reduce the set of reference fragments in $frags_{all}$, to a non-redundant set with regards to binding mode. However, if the CAH2 DUD-E actives have a bias in binding mode *e.g.* the majority have a common pharmacophore or binding motif, then the use of the cumulative

method with a reference fragment set with non-redundant binding modes will not perform well.

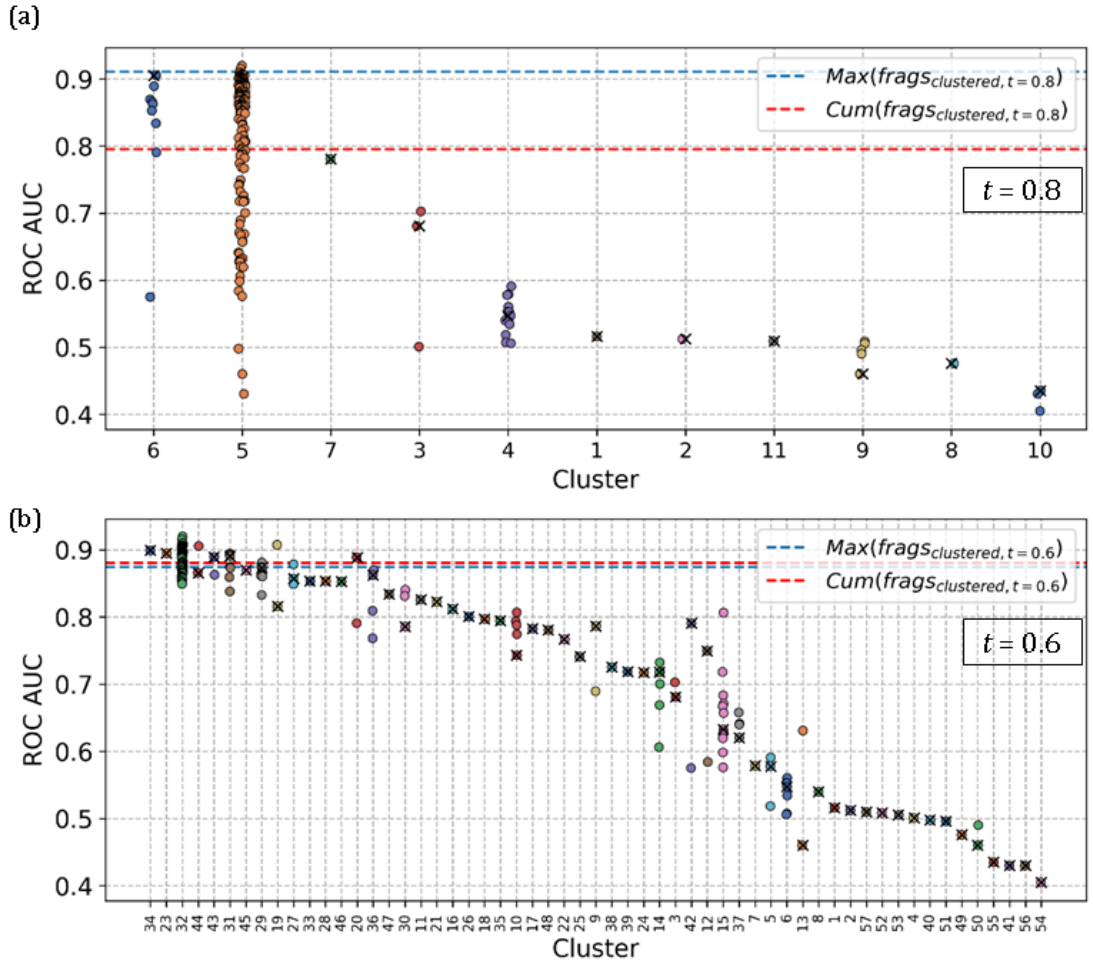


Figure 4.21. The distribution of ROC AUCs for each fragment, grouped by cluster. Hierarchical clustering was performed using (a) a cutoff of t=0.8, which formed 11 clusters, and (b) a cutoff of t=0.6, which formed 57 clusters. The black crosses signify the representative fragment in each cluster. The clusters are ordered from left to right by decreasing median ROC AUC achieved by the cluster.

In contrast Max($frags_{clustered,t=0.8}$) performed relatively well and obtained a ROC AUC better than any of the individual fragments in $frags_{clustered,t=0.8}$, as the best had a ROC AUC of 0.905.

Max($frags_{clustered,t=0.8}$) performed better than Max($frags_{all}$) (ROC AUC 0.911 versus ROC AUC 0.852). This indicates that some of the fragments have negative contribution to the enrichment. Again, an explanation for this may be because some reference

fragments lead to an irrelevant docked pose being chosen over the near native pose for the active/decoy molecule.

The effect of using more clusters was investigated by lowering the clustering threshold to t=0.6 which gave 57 clusters (Figure 4.21). Smaller ranges of ROC AUCs are seen within each cluster, for example, visual inspection of cluster 32 reveals a common sulfonamide between all 69 fragment structures in the cluster (Figure 4.22). For the 57 clusters, the ROC AUCs for Max($frags_{clustered,t=0.6}$) and Cum($frags_{clustered,t=0.6}$) were 0.874 and 0.880, which decreased and increased with respect to the corresponding group fusion result for threshold t=0.8.

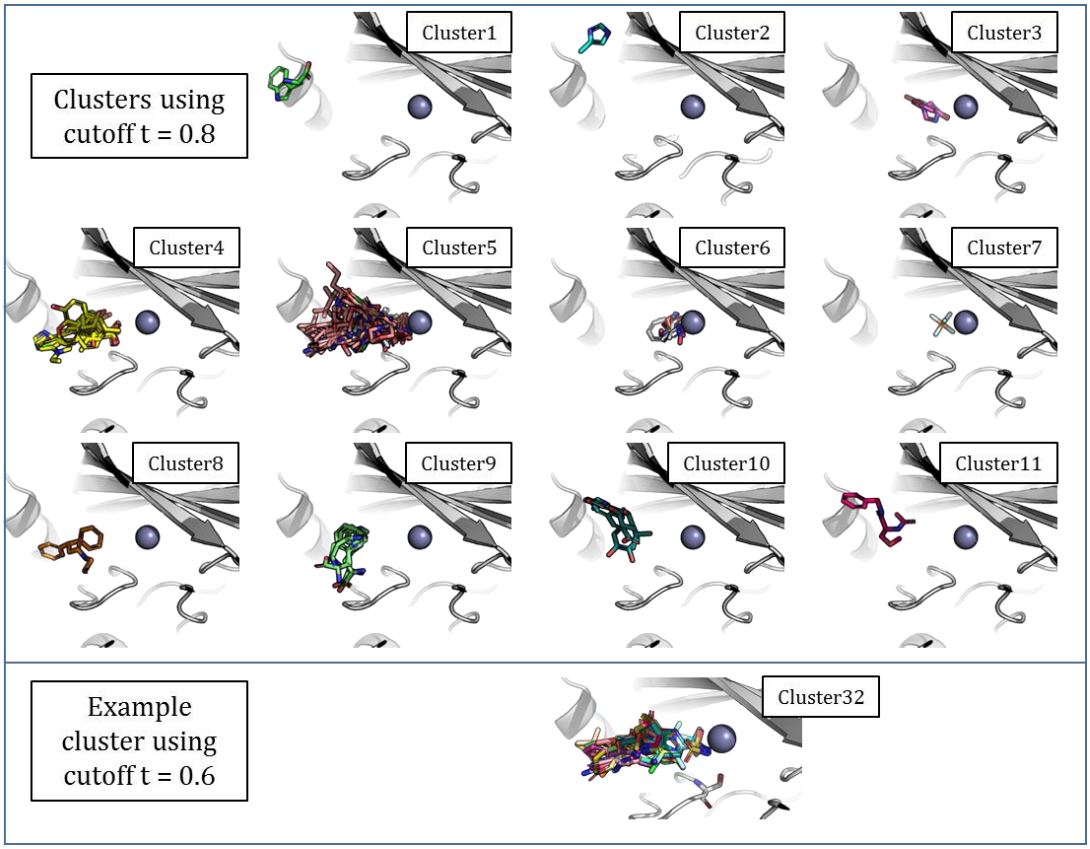


Figure 4.22. The CAH2 clusters resulting from two different cutoffs, t=0.8 and t=0.6. This figure was produced using PyMOL (Schrödinger, LLC.).

As the sulfonamide group is a common substructure in CAH2 inhibitors, and one that appears frequently in the top scoring fragments, an interesting question arises – what

proportion of the CAH2 DUD-E active molecules contains the sulfonamide group? A substructure search using the sulfonamide group SMARTS pattern **-[S;D4](=[O;D1])(=[O;D1])-[N;D1]* was performed on the DUD-E CAH2 actives, $frags_{all}$ and $frags_{clustered,t=0.8}$. The proportion of molecules containing the sulfonamide group were 99% (7361 out of 7469) for the DUD-E actives, 57% (109 out of 190) for $frags_{all}$ and 25% (3 out of 12) for $frags_{clustered,t=0.8}$.

As nearly all DUD-E actives contain the sulfonamide group, they are likely to bind in a similar fashion; with a sulfonamide group interacting with the Zn ion and forming an anchoring interaction. Therefore, using a variety of 3D poses *i.e.* from the clustered fragments, will not give any useful information and explains why Cum($frags_{clustered,t=0.8}$) performed so poorly.

4.3.3.4 Trypsin 1, TRY1

Trypsin I, TRY1, is a serine protease found in the digestive system of many vertebrates, and is involved in the hydrolysis of peptides at the C-terminal side of lysine or arginine (Schmidt et al., 2003).

All known trypsins have a conserved fold and a conserved catalytic triad in the active site, which consists of residues SER195, HIS57 and ASP102, with the flanking residues also being conserved. Benzamidine is a competitive inhibitor of trypsin and is known to exhibit a conserved binding mode and is a substructure of many different TRY1 ligands (Stubbs, 1998; Guvench et al., 2005). It occupies the specificity pocket, where the amidine group makes hydrogen-bonds with, GLY219, ASP189 and SER 190, which form a common anchor point seen in its related derivatives (Sherawat et al., 2007).

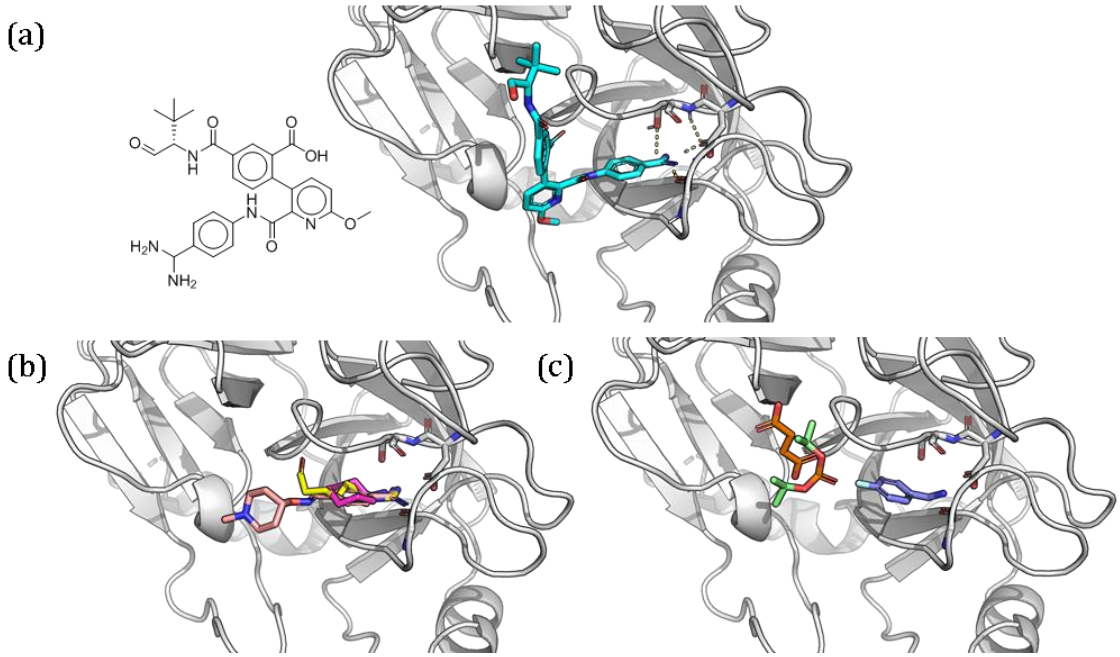


Figure 4.23. (a) The DUD-E reference ligand for TRY1 (PDB ID 2AYW). Interactions with the catalytic triad (white sticks) are shown by the dotted lines. (b) The three reference fragments with the best ROC AUCs. Two contain the amidine group that is common to trypsin inhibitors. (c) The three reference fragments with the worst ROC AUCs. The two worst reference fragments are shown in orange and red sticks and bind away from the catalytic triad. The third worst reference fragment is shown in purple sticks and is still capable of forming interactions with the catalytic triad. This figure was produced using PyMOL (Schrödinger, LLC.).

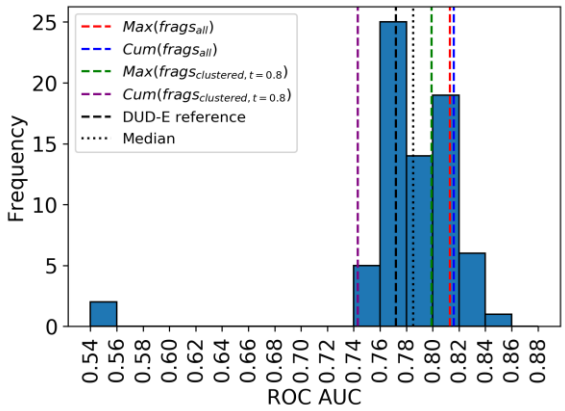


Figure 4.24. Distribution of ROC AUC's achieved when each of the 72 TRY1 fragments in *frags*$_{all}$ is used as the reference molecule to rescore with SuCOS the docked poses of DUD-E TRY1 actives and decoys. The performance of the TRY1 DUD-E reference ligand along with the Max(*frags*$_{all}$), Cum(*frags*$_{all}$), Max(*frags*$_{clustered,t=0.8}$), Cum(*frags*$_{clustered,t=0.8}$) and median of the distribution are also shown by the vertical dotted lines.

	Frags$_{all}$		*Frags*$_{clustered,\ t=0.8}$		DUD-E reference ligand	Best fragment
TRY1	Max SuCOS	Cumulative SuCOS	Max SuCOS	Cumulative SuCOS		
	0.813	0.816	0.799	0.743	0.772	**0.843**

Table 4-7. Summary of the ROC AUCs achieved by the investigated group fusion methods for TRY1. For λ=0.5, the single best fragment performed the best, whilst Cum(*frags*$_{clustered,t=0.8}$) performed the worst, in terms of ROC AUC.

A total of 72 fragment structures were downloaded and the distribution of ROC AUCs for each fragment, when used as a single reference molecule, is shown in Figure 4.24 (for individual ROC AUCs see Appendix Table B-10). The three top performing fragments have ROC AUCs of 0.843, 0.840 and 0.835 (PDB IDs 3RXA, 3VPK and 3A7Z). The second and third best fragment contain the amidine group, but the top ranked does not and instead has an amine group which is still capable of forming similar interactions with GLY219, ASP189 and SER 190 in the specificity pocket (Figure 4.23b).

The three fragments which showed the worst ROC AUCs have ROC AUCs of 0.550, 0.552 and 0.750. All three do not contain the amidine group. The worst two fragments bind away from the anchoring site containing the catalytic triad (Figure 4.23c). The third worst fragment has a relatively good ROC AUC and contains an amine group that is be able to interact with the catalytic triad (Figure 4.23c). Thus, these anchoring interactions with the catalytic triad may be necessary for classifying actives from inactives.

Cum($frags_{all}$) and Max($frags_{all}$) performed similarly, achieving a ROC AUC of 0.816 and 0.813 respectively (Table 4-7) and both performed worse than the best individual fragment but better than the median (Figure 4.24).

Clustering $frags_{all}$ with threshold t=0.8 produced four clusters, where again one cluster was highly populated; in this case it was cluster 1 which had 69 fragments, and ROC AUCs ranging from 0.750 to 0.843 (Figure 4.25 and Figure 4.26), whereas the remaining clusters only contained one fragment. Again, this suggests that there may be a bias in binding mode in the DUD-E actives for TRY1, that is to some extent mirrored in this reference fragment dataset.

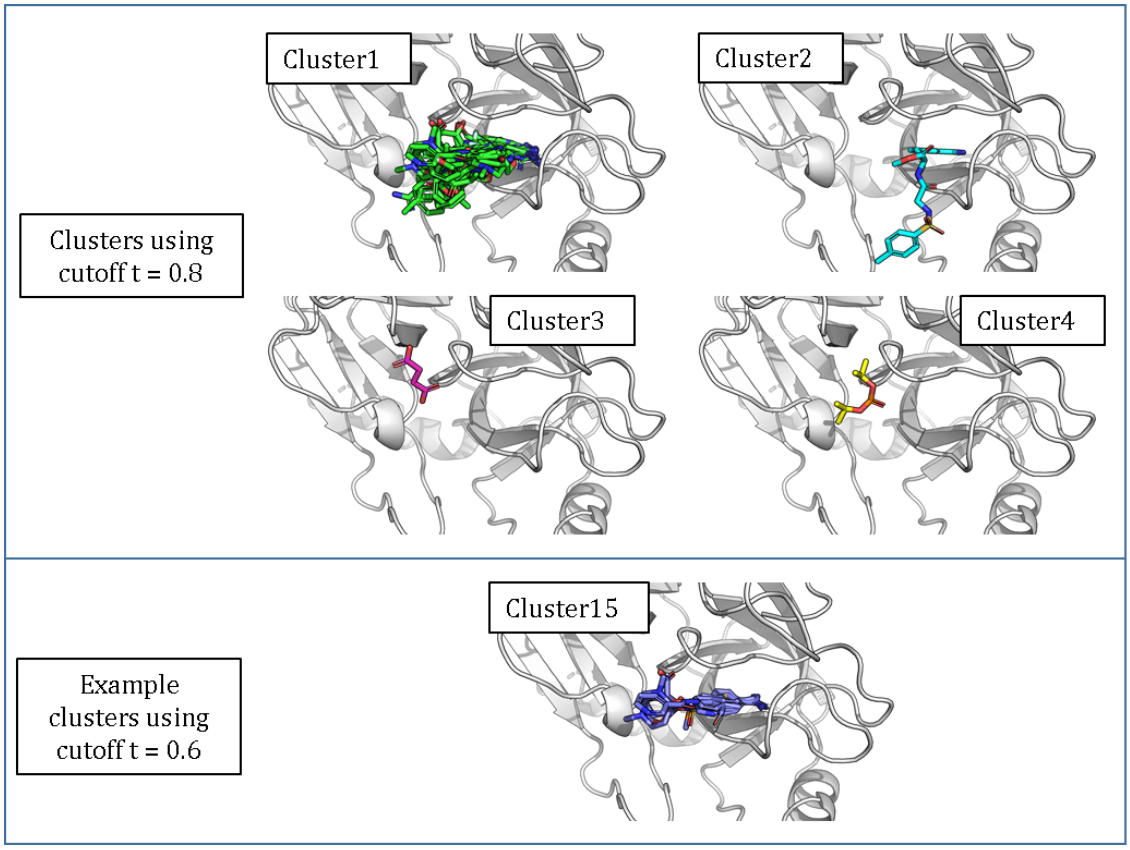


Figure 4.25. The TRY1 clusters resulting from two different cutoffs, t=0.8 and t=0.6. This figure was produced using PyMOL (Schrödinger, LLC.).

The representative fragments in $frags_{clustered,t=0.8}$ are ranked 13, 29, 71 and 72 out of 72 and have a maximum and median ROC AUC of 0.816 and 0.675 respectively (Appendix Table B-10). Cum($frags_{clustered,t=0.8}$) and Max($frags_{clustered,t=0.8}$) performed worse than the best fragment within $frags_{clustered,t=0.8}$ but better than their median; thus in this case for TRY1, using either a cumulative or a maximum fusion method with fragments that show different binding modes/regions of binding was not advantageous.

The clustering threshold was lowered to t=0.6 which give 19 clusters, with distributions of ROC AUCs shown in Figure 4.26b. The ROC AUCs obtained with the representative fragments from the 19 clusters was 0.819 and 0.763 for Cum($frags_{clustered,t=0.6}$) and Max($frags_{clustered,t=0.6}$) respectively. Thus lowering the clustering threshold to t=0.6 improved the ROC AUC for Cum($frags_{clustered}$) but for Max($frags_{clustered}$) it decreased.

Again visual inspection of the clusters formed from threshold t=0.6 showed more similar binding modes within each cluster and a smaller range of ROC AUCs within each cluster when compared to the clusters formed from threshold t=0.8 (Figure 4.25 shows cluster 15 as an example).

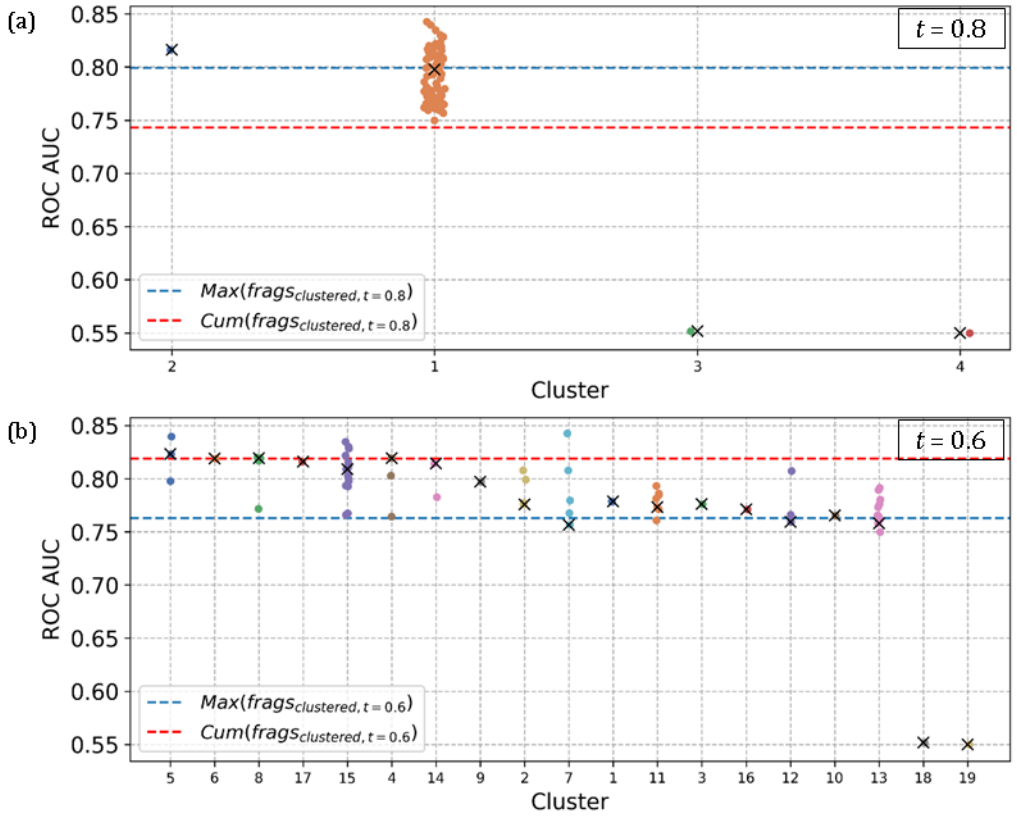


Figure 4.26. The distribution of ROC AUCs for each fragment, grouped by cluster. Hierarchical clustering was performed using (a) a cutoff of t=0.8, which formed 4 clusters, and (b) a cutoff of t=0.6, which formed 19 clusters. The black crosses signify the representative fragment in each cluster. The clusters are ordered from left to right by decreasing median ROC AUC achieved by the cluster.

Similar to the case of CAH2 and the sulfonamide group, TRY1 also has an anchoring group that features frequently in the reference fragments that give rise to high ROC AUCs, namely the amidine group. Hence, to investigate potential bias of this functional group within the actives of TRY1 in DUD-E, the DUD-E actives were searched for the amidine group, using SMARTS pattern *N=C(N)**, and was found in 84% of the actives (5706 out of 6922). A similar substructure search was performed on $frags_{all}$ and $frags_{clustered,t=0.8}$, which gave proportions of 54% (39 out of 72) and 50% (2 out of 4),

respectively. Thus, the large proportion of amidines in the TRY1 DUD-E actives has implications in which reference fragment(s) gives the higher ROC AUCs, *i.e.* reference fragments with the amidine substructure are likely to have similar binding mode to 84% of the actives. Hence, the method of clustering the reference fragments to generate a variety of binding modes was not advantageous.

4.1.1.1 Summary

In conclusion, apart from BACE1, all fragment datasets had at least one fragment which when used on its own as a reference, resulted in a ROC AUC better than any of the data fusion methods. Hence, for well-studied targets *e.g.* those with actives molecules that show a bias towards established ligand binding motifs, a single fragment that possesses this established ligand binding motif is the best to use as the reference to obtain the highest AUCs. For example for CAH2 or TRY1, the reference fragment should contain the key sulfonamide or amidine anchoring group respectively. However, the 'best' fragment is generally unknown and even for these well-studied targets, it may limit the advance of discovering new pharmacophores, anchoring groups or scaffolds, which may be tighter binders.

As there is no consistent best method across the four targets from this study (see Table 4-8 for summary), no 'one-size-fits-all' conclusion can be drawn as to if there is a 'best' data fusion method for when there are multiple fragment-protein structures, where each could be used as references. Also, from this study, it is not clear whether or not it is best to cluster the fragment binding poses before using the set as references. However, if there is clearly a common anchoring group or if there is a known bias in binding modes in the active molecules of the validation dataset, then clustering may not be of any use. Further investigations could involve removing outliers, or singleton clusters, to see if

the ROC AUC improves. Currently, it is unclear if using a reference set that contains diversity in binding mode is advantageous, and if it is, under what circumstances, as for the cases of CAH2 and TRY1, the evidence that the majority of actives contain the common anchoring groups suggest it was not advantageous.

By picking targets that have established ligand binding motifs, my investigation may have been biased, as a particular molecule class may be over-represented in the actives of a particular target. For example, sulfonamides for CAH2, and amidines for TRY1. This theory could be related to the hidden bias within the DUD-E dataset that Chen *et al.* reported (Chen et al., 2019) that I discussed earlier in the introduction to this chapter (Section 4.1.1). The over-represented substructures *e.g.* analogue bias for actives could lead to a bias in the binding mode of the actives *e.g.* the majority of actives share a common binding mode in the binding site. Moreover, decoy bias, where the decoys were chosen based on high topological fingerprint dissimilarity to the actives could mean that the decoys have a dissimilar binding mode with respect to the actives. Hence, if the majority of active molecules bind in a similar mode then the use of a diverse set of reference fragments, in terms of binding mode, to score against would not be effective. Moreover, scoring against the diverse set using a cumulative group fusion method would also not be effective and instead a better strategy would be to use a reference fragment that shares the common binding mode present in the actives to score against. This again ties in with knowing which is the 'best' reference fragment.

The best individual fragments for each target consistently showed important interactions in the binding site reported by past literature; whereas the least informative fragments bind away from the site of important interactions and are also usually in another binding mode cluster. Thus, when designing candidate ligands, a potential strategy could involve using only candidates that contain the 'essential' feature(s), if

known, with the possibility of using additional optional feature(s) to target sub-pockets in the binding site.

However, one must bear in mind that these validation studies involving four DUD-E targets are dissimilar to the situation of a prospective study after an initial fragment screening campaign, where there is little or no ligand binding mode data other than that produced from the screen. Furthermore, there may also be few known actives for the target. It is possible that a more carefully chosen validation dataset would have addressed the group fusion question better for this situation.

Condition	BACE1	CDK2	CAH2	TRY1
Cum($frags_{all}$) > Max($frags_{all}$)	Yes	Yes	Yes	Very similar
Cum($frags_{clustered,t=0.8}$) > Max($frags_{clustered,t=0.8}$)	Yes	Yes	No	No
Cum($frags_{all}$) > Cum($frags_{clustered,t=0.8}$)	No	Yes	Yes	Yes
Max($frags_{all}$) > Max($frags_{clustered,t=0.8}$)	No	Yes	No	Yes
Cum($frags_{clustered,t=0.8}$) > $frag_{clustered,best}$	Yes	Very similar	No	No
Max($frags_{clustered,t=0.8}$) > $frag_{clustered,best}$	Very similar	No	Very similar	No

Table 4-8. Summary of the group fusion results for the four targets.

4.4 Conclusions

In the previous chapter, I explored the use of SuCOS for the specific case of comparing binding poses of fragments and their elaborated counterparts. In this chapter, I extended the investigation by looking at using SuCOS in virtual screening, for compounds that are not necessarily related by a common substructure.

I used the DUD-E dataset to compare ranking using the native AutoDock Vina scoring function versus rescoring with SuCOS, and found that SuCOS was better on average. For each DUD-E target, the optimal weights of the shape and chemical feature components in SuCOS were explored and although there was variation in the optimal

weights, the median weight was found to be half shape and half chemical features, which supports the original choice of weights used in the previous chapter.

Finally, I investigated how to use SuCOS where multiple protein-fragment-hit structures are available *i.e.* after a fragment screen. As SuCOS is computed between a single reference and a single query molecule, I investigated two group fusion methods – cumulative and maximum – to obtain a single SuCOS value between multiple references and one query molecule. I also examined the effect of clustering the reference fragments by shape and chemical feature overlap, using Tanimoto SuCOS, to generate a set of fragments which have non-redundent binding modes. To explore the two group fusion methods and the effects of clustering, I compiled a set of reference protein-fragment crystal structures for four DUD-E targets, namely BACE1, CAH2, CDK2 and TRY1.

Neither group fusion method, maximum or cumulative, was consistently better across the four targets. Moreover, it was unclear whether clustering the reference fragments by Tanimoto SuCOS improved the ROC AUC. However, the best performing reference fragments often had a substructure in common, which was also frequently found in the DUD-E actives. For example, the best reference fragments for CAH2 contained the sulfonamide group, which was also present in most of the active DUD-E molecules. Thus for these targets, clustering the fragments on Tanimoto SuCOS may not be useful, as the actives will most likely belong to the same binding mode cluster.

For all four targets, clustering the fragments on Tanimoto SuCOS also tended to result in one or two clusters that were highly populated, whilst the remainder were sparsely populated. Inclusion of fragments that represent outliers with respect to binding modes could have had an adverse effect on the group fusion methods results as these outlier

binding modes may not bear any resemblance to any of the binding modes of the DUD-E actives. Hence, the inclusion of these outliers may have just added noise to the result. Further investigations could involve leaving out these singleton clusters and seeing how this effects the enrichment of DUD-E actives.

Moreover, the representative fragment of the highly populated cluster tended to have better ROC AUC than the representative fragment of the less populated clusters; thus indicating that there is a biased in binding mode in the DUD-E actives that is to some extent mirrored by the biased in binding modes of the curated reference fragment dataset.

However, I intended on applying the group fusion method to retain as much information as possible and to prioritise follow-up compounds for targets that are not as well studied, where there are not many known actives nor established binding motifs, and the cluster that represents essential features for activity is not known. In this situation, the method of clustering may be advantageous but further investigations are necessary.

Chapter 5 Using Bayesian Optimisation for Ligand-Based & Structure-Based VS

5.1 Introduction

Following a fragment screening campaign, numerous fragment-protein crystal structures may result. The vital question is, how can we use this data to best inform our decision as to what to make next? The hit-to-lead optimisation process can be thought of as a Bayesian optimisation problem – our objective function can be a single objective, such as measured potency or predicted binding affinity (from docking, or single point energy calculation), or a multi-objective goal. We can define our search space as all the potential candidate molecules that can be synthesised following the fragment screen.

In this chapter, I explore the use of Bayesian optimisation on two ligand-based validation datasets and four structure-based validation datasets. The two ligand-based validation datasets have been previously explored in the context of Bayesian Optimisation; however, I build upon this previous study by exploring alternative molecular descriptors and alternative kernels, with the aim to identify the best performing. To my knowledge, currently no one has yet explored Bayesian optimisation with a structure-based validation dataset; hence, this study aimed to compare the performance of two 3D-descriptors against ligand-based descriptors. The results have

potential application for prospective studies in hit-to-lead optimisation, for example following a fragment screening campaign.

5.1.1 Introduction to Bayesian Optimisation

Bayesian optimisation is a strategy to minimise or maximise an objective function that is expensive to calculate.

The maximisation problem can be written as:

$$x^* = \underset{x \in A}{\operatorname{argmax}} f(x) \qquad (5.1)$$

where $f(x)$ is the objective function, x can take any value in the exploration space of interest A, and x^* is the point in the domain of interest which yields the optimal value for the objective function. In our case, x can be thought of as a candidate small molecule, while A is the set of molecules in the chemical space defined by the reagents of reactions we can use. The explicit form of f is unknown but we are able to evaluate it at any point x within the domain of interest. By iteratively querying the search space, we can update the Bayesian response surface.

Bayesian optimisation has applications in the optimisation of hyperparameters for machine learning (Snoek et al., 2012), reinforcement learning (Brochu et al., 2010) and in finding the lowest energy conformer for a molecule with multiple rotatable bonds (Chan et al., 2019a, 2019b). They are deemed successful if they can find the optimal solution in fewer iterations than another method. Bayesian optimisation is particularly useful for problems where evaluations of the objective function are expensive to calculate *e.g.* a quantum mechanics calculation or a biophysical assay, and when the number of iterations is not too large. When a Gaussian process, GP, is used as the

surrogate model (discussed in Section 5.1.2), the computational complexity is $O(N^3)$, where N is the number of observations; hence, as the computational cost is exponential with respect to the number of observations, caution should be taken for running a large number of iterations.

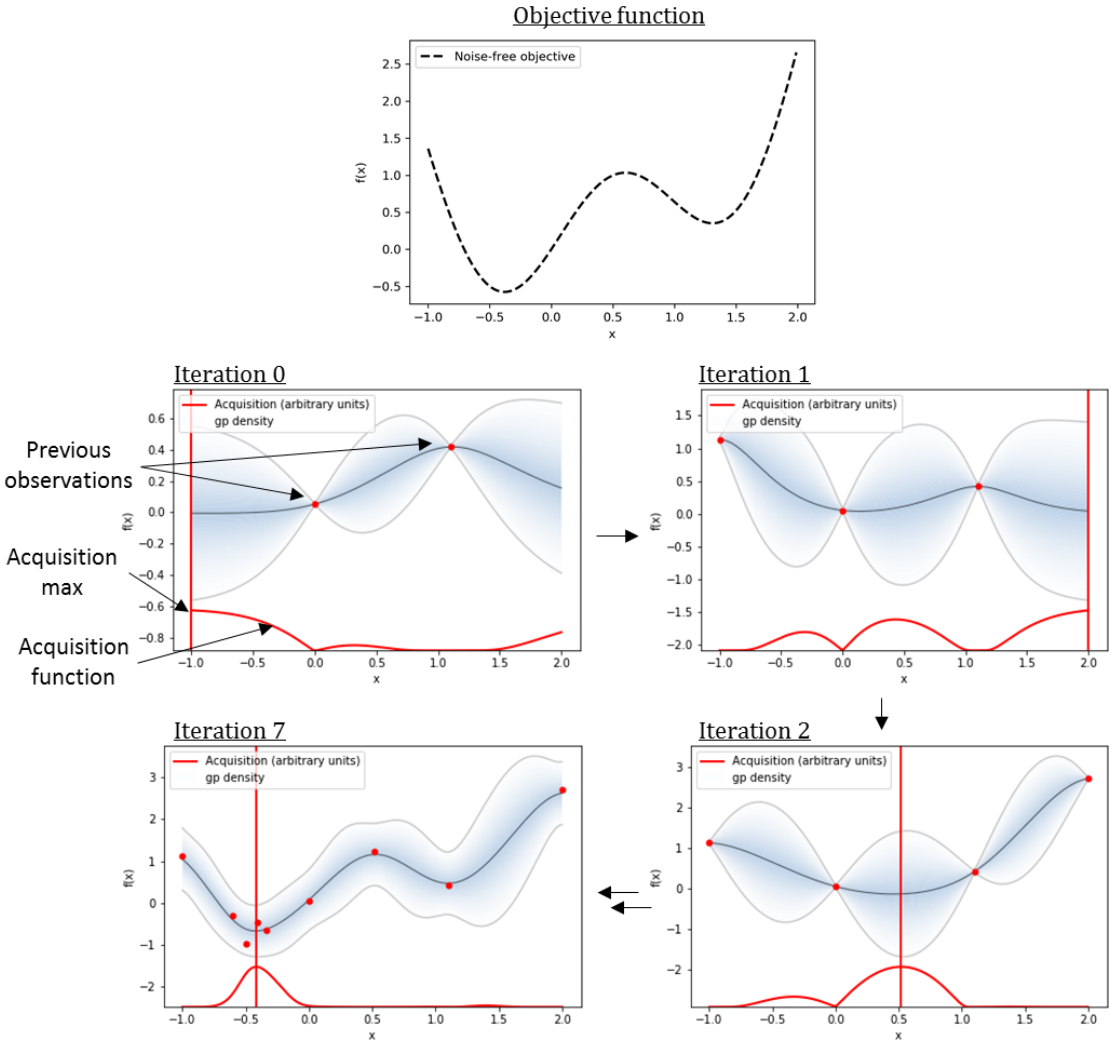


Figure 5.1. An example of a Bayesian optimisation run over seven iterations. The top plot shows the true objective function whose exact form is usually unknown. The four plots below show iterations 0, 1, 2 and 7, where the mean of the probabilistic model is shown by the black curve, previous observations by the red dots, the confidence intervals by the grey boundaries and the acquisition function by the red curve at the bottom of each plot. The vertical line in each plot denotes where the acquisition function is maximum and is where the next observation will occur. This optimisation was initialized with two points. This figure is adapted from (Shahriari et al., 2016).

Bayesian optimisation acts as an alternative to grid searching and random searching; both methods do not use previous evaluations to determine where to sample next, and

as a result they tend to be inefficient and require a greater number of iterations to find the optimal value. In contrast, Bayesian optimisation updates the model after each iteration and so makes *informed* decisions, thus the goal is reached within fewer iterations.

The general Bayesian optimisation process for finding the global maximum is outlined in the following steps:

1. Place a Gaussian process prior (or an alternative probabilistic surrogate model) on the objective function f.
2. Choose the next point to sample, x_n, where the acquisition function is maximum.
3. Make an observation at x_n of the objective function, $y_n = f(x_n)$.
4. Update the posterior distribution on f using all data.
5. Update the acquisition function.
6. Repeat steps 2-5 until number of iterations is reached.
7. Return the point with the optimal value of $f(x)$.

In the following sections, I introduce the concepts of the probabilistic surrogate model, the acquisition function and the importance of molecular representation and the choice of kernel; however, for a more comprehensive review, I refer the reader to (Shahriari et al., 2016; Brochu et al., 2010).

To relate the steps of Bayesian Optimisation to work performed this chapter, an example of an objective function could be pIC_{50}, where the goal is to find the molecule with the highest activity or pIC_{50} (Section 5.3.1) for a particular biological target. At the start of the optimisation process (iteration 0, Figure 5.1), we may only know the activities of two molecules out of our total candidate dataset. From this limited data, we

can already use a probabilistic surrogate model *e.g.* a Gaussian Process, to estimate the pIC_{50}s of all other candidate molecules and the associated uncertainty with each estimate. We can begin the next iteration by choosing the next molecule to observe. An observation in this context could mean synthesising and measuring its biological activity against the target. We choose the molecule based on the tradeoff between predicted activity and uncertainty in the prediction. This tradeoff is represented in the acquisition function. After we observe the new molecule's activity, we can update the probabilistic surrogate model based on this value and all previously observed activities, which results in the update of all other molecules' predicted activity and their associated uncertainty (iteration 1, Figure 5.1). The acquisition function is also updated. This is an iterative process where in this case, a molecule's biological activity is measured at each iteration. The Bayesian Optimisation process ends after a predefined number of iterations is reached (iteration 7, Figure 5.1), where we can conclude with the most potent molecule found. The motive for using Bayesian Optimisation is to find the most potent molecules in fewer iterations than other methods, such as the hit-to-lead process traditionally led by medicinal chemists or random sampling.

There are two main components in the Bayesian optimisation framework – the probabilistic surrogate model and the acquisition function.

5.1.2 Probabilistic Surrogate Models

In Bayesian optimisation, one of the most popular probabilistic surrogate models are Gaussian Processes, or GPs. A GP is a distribution over functions, $f(x)$, of which the distribution is fully specified by its mean function, $\mu(x)$, and positive definite kernel or covariance function, $k(x, x')$, where x are the input values and (x, x') all possible pairs in the input domain:

$$f(x) \sim GP(\mu(x), k(x, x')) \quad (5.2)$$

Alternative probabilistic surrogate models include random forests (Hutter et al., 2011).

5.1.3 Acquisition Function

Bayesian optimisation uses an acquisition function which determines where to sample next. There are many different acquisition functions and they differ in the trade-off between exploration and exploitation. A good acquisition function should have a balance between both. Exploitation refers to sampling the objective function where there is a high predicted mean, whereas exploration samples where the model has a large amount of uncertainty. If the acquisition function is too exploitive, the optimisation can get stuck in a local maximum, whereas if it is too explorative, the sampler may rarely pick points with good values and hence will take more iterations to reach the optimal value.

Expected improvement, EI, is one of the most commonly used acquisition functions and can be defined as:

$$EI(x) = \sigma(x)(Z\Phi(Z) + \phi(Z)) \quad (5.3)$$

where Z is,

$$Z = \frac{f(x^+) - \mu(x)}{\sigma(x)} \quad (5.4)$$

and $f(x^+)$, x^+, $\mu(x)$ and $\sigma(x)$ are the value of the currently best point observed, the point where the currently best value was observed, the predicted mean and predicted standard deviation, respectively; and $\Phi(.)$ and $\phi(.)$ are the cumulative distribution

function and the probability density function of the standard normal distribution respectively. Equation (5.3) has two terms, the former, $Z\Phi(Z)$, relates to exploitation and is controlled by the mean, whereas the latter, $\phi(Z)$, term relates to exploration and is controlled by the standard deviation.

5.1.4 Greedy Search and Random Search

In contrast to Bayesian optimisation techniques where there is a trade-off between exploration and exploitation, greedy search algorithms always pick the optimal predicted value to sample next *e.g.* always chooses the molecule with the highest predicted pIC_{50} to make next. Greedy search algorithms do not perform well when there are many local optima as it will get stuck and cannot further explore the optimisation landscape.

Random search is one of the simplest search algorithms. It does not require computation of any derivatives to search the domain but instead searches over a uniform probability distribution. It is often used as a baseline method to compare the performance of different search algorithms.

5.1.5 Choice of Kernel

In Gaussian Process regression, the choice of kernel, also known as the covariance function, determines the correlations between different points in space and the smoothness of the predictive model. If two points, x, x^* are very close to each other in the input space, then we would expect the observed values at those points to be similar. If an inappropriate kernel is chosen, then the surface of the predictive model will be bumpy *i.e.* molecules that appear very similar in space will have very different

biological activities; hence the model will be poor and perform similar to random sampling.

The square-exponential kernel, k_{SE}, is a commonly used kernel. It is also known as the Gaussian kernel or the radial basis function (RBF) kernel. This is one of the kernels that I explored in this chapter.

$$k_{SE}(x, x^*) = a_0 e^{\frac{-|x-x^*|^2}{2l^2}} \quad (5.5)$$

where a_0 and l are the variance and length scale respectively. However, in the field of cheminformatics, especially when using molecular fingerprints such as Morgan fingerprints, the method of measuring the similarity between two fingerprints is usually not by a squared-exponential kernel but rather by Tanimoto similarity, also known as the Jaccard index (Bajusz et al., 2015; Tanimoto, 1957).

$$k_{Tanimoto}(x, x^*) = \frac{K(x, x^*)}{K(x, x) + K(x^*, x^*) - K(x, x^*)} \quad (5.6)$$

where $K(.,.)$ is the scalar product between the two components. The Tanimoto kernel has not been implemented in GPy (GPy, Version 1.9.6, 2012) but it is a kernel that I investigated in this chapter.

The Manhattan distance, also known as the city block distance, is the sum of the absolute differences of their Cartesian coordinates. It is so called because in a 2D case, it is akin to how a taxi driver would drive in the grid-like road layout in Manhattan. The general definition is,

$$d_{Man}(x, x^*) = \sum_{i=1}^{n} |x_i - x_i^*| \tag{5.7}$$

In this chapter, I use the Manhattan distance to calculate the similarities between ElectroShape vectors. The ElectroShape authors (Armstrong et al., 2010), transformed the Manhattan distance equation (5.7) into an ElectroShape similarity score in order to calculate the similarity between the Electroshape vectors. By dividing by 15 the distance is normalized, as there are 15 ElectroShape descriptor values (Section 5.2.8), and inversion changes the distance into a similarity with a range between zero and one:

$$k_{InvMan}(x, x^*) = \frac{1}{1 + \frac{1}{15} d_{Man}(x, x^*)} \tag{5.8}$$

and is what I use in this chapter for the Manhattan kernel (Section 5.3.1).

5.1.6 Molecular Representations

Molecular representation has long played an essential role in cheminformatics; any accurate predictive machine learning model is dependent on a good molecular representation (Winter et al., 2019; Huang and Von Lilienfeld, 2016).

There are numerous ways of representing a molecule computationally and also many ways of calculating similarity between these representations. One of the most widely used representations are 2D methods that encode the atom connectivity in the molecule into molecular fingerprints that are 1D bit-strings.

The extended-connectivity fingerprint, ECFP, is an example of such a fingerprint. It is also known as the circular fingerprint and is based upon its older variant, the Morgan

fingerprint (Rogers and Hahn, 2010; Morgan, 1965), which I discussed in Section 1.2.3.1.

Another widely-used but non-circular 2D molecular fingerprint is the MACCS fingerprint (Durant et al., 2002). Each MACCS fingerprint is a 166 bit vector, where each of the 166 bits is associated with a SMARTS pattern, and the 1 or 0 represents the presence or absence of that substructure or atom. For example, one of the 166 bits represents the presence/absence of a ring of size 4 in the molecule.

In this chapter, I explore the effect of different molecular representations on the Bayesian Optimisation process.

5.1.7 Multi-Armed Bandit Problem

In the multi-armed bandit problem (Mahajan and Teneketzis, 2008), the optimisation is over a discrete space $\{x_a\}_{a=1}^{K} \subset \chi$, where χ is the feature space, K is the number of discrete points and $x_1, x_2, \ldots x_K$ are the K possible inputs. For example, in a drug discovery setting, χ could be all possible Morgan fingerprints, K is the total number of all possible Morgan fingerprints, and x_1 is the Morgan fingerprint of the first molecule. The name multi-armed bandit originates from the situation where a gambler is playing a row of slot machines, or *one-armed bandits*, and each machine provides its own reward with unknown probability. The gambler must decide which machines to play next and in what order, to gain the most reward. There must again exist a trade-off between exploration and exploitation, in order to achieve the highest reward.

In this chapter, Bayesian optimisation was performed over a discrete space, using a multi-armed bandit type domain. The optimisation chooses from an input set of molecules which one to query at each iteration and is essentially a prioritisation

exercise of what to make next. A situation where this would be useful would be in the context of a combinatorial explosion produced by reaction enumeration, where more molecules are suggested than is feasible to synthesis and biologically test; hence, Bayesian optimisation would prioritise which follow-up compounds to make first. The advantage of this method over molecular generators *i.e.* Variational Autoencoders described below, is that there is greater control over the molecular properties of all input molecules. For example, they could be generated by poised enumeration (Cox et al., 2016) (also discussed in Sections 1.3.2 and 2.2.2) so all input and output molecules can have high synthetic tractability and have a valid synthetic route annotation. Additionally, the synthesis route to the input molecules could be restricted to just a few synthetic reactions *i.e.* only molecules made by amide reaction, so synthesis could be easily automated and parallelised.

This use of multi-armed bandits contrasts to using molecular generators *e.g.* Variational Autoencoders (VAEs) (Kingma and Welling, 2013), which can suggest novel molecules, not present in the input set, in which case the Bayesian optimisation would be over a continuous space. One advantage of VAEs is molecular novelty, which may not be suggested by poised enumeration or by experienced medicinal chemists.

5.1.8 Prior State of the Art and Chapter Aims

The work presented in this chapter firstly builds on the work reported by Pyzer-Knapp who investigated using Bayesian Optimisation to accelerate lead discovery (Pyzer-Knapp, 2018). He used two ligand-based validation datasets to investigate two different Bayesian optimisation techniques: expected improvement (EI) and adaptive expected improvement (AEI), and benchmarked them against a greedy search and a random search. AEI is similar to EI but the amount of exploration versus exploitation can be

dynamically adjusted based on sampled levels of uncertainty (Jasrasaria and Pyzer-Knapp, 2019).

The first ligand-based validation dataset involved inhibitors of the matrix metalloproteinase 12 (MMP-12) (Pickett et al., 2011) and the second was the malaria dataset (Spangenberg et al., 2013). He suggested that the former MMP-12 dataset was an easier optimisation challenge that has a optimisation landscape with a large wide basin whereas the latter malaria dataset is more challenging as it is composed of three different datasets, and is a whole cell study, with competing SARs.

He measured the performance of the various search methods using four criteria, which included measuring the number of 'desirable' molecules found versus number of iterations and found that for the simple MMP-12 dataset, there was similar performance of the Bayesian Optimisation methods compared to the greedy search but all were better than random sampling. However, for the more complex malaria dataset, he found better performance of the Bayesian Optimisation techniques compared to the greedy search.

In this chapter, I build on the work of Pyzer-Knapp and in the first two studies of this chapter, I investigate the effects of using different molecular representations and alternative kernels on the same two ligand validation sets. These effects were not explored in Pyzer-Knapp's study; he only investigated using RDKit's Morgan Fingerprints, with radius of 512 , to represent the ligands and only investigated using the squared-exponential kernel, also known as the RBF kernel, for the Bayesian Optimisation.

Following these ligand-based validations, I investigate the effects of using structure-based molecular descriptors on the Bayesian Optimisation process and see how this compares to the ligand-based descriptors. To the best of my knowledge, there have been

no reports of using Bayesian Optimisation that uses 3D structural information. The motivation of this structure-based investigation is linked to the use-case following a fragment-screening campaign where there may be iterations of fragment elaboration and crystal structure determination, with the objective of maximising binding affinity or potency.

5.2 Methods

5.2.1 MMP-12 Dataset Preparation

The first dataset involves inhibitors for matrix metalloproteinase 12, MMP-12. MMP-12 is part of a family of the matrix metalloproteinase family which play a role in the disassembly of the extracellular matrix, is expressed mainly by macrophages, and is also associated with many pathological conditions such as aneurysm formation and rheumatoid arthritis (Dean et al., 2008).

The MMP-12 inhibitors were originally reported by Pickett *et al.* who investigated using a genetic algorithm for automated lead optimisation (Pickett et al., 2011). All the inhibitors have a common biaryl sulfonamide scaffold (Figure 5.2a), with varying R_1 and R_2 groups, and were made as part of an effort to synthesise a complete 50 x 50 array with biological testing (Figure 5.2b). Each row and column in the grid map represents a matched molecular series, MMS, which are a set of compounds that differ in substituent at the same position (Ehmki and Kramer, 2017). This concept is an extension of matched molecular pairs, MMPs, which uses the same definition but only between a *pair* of compounds (Kenny and Sadowski, 2005).

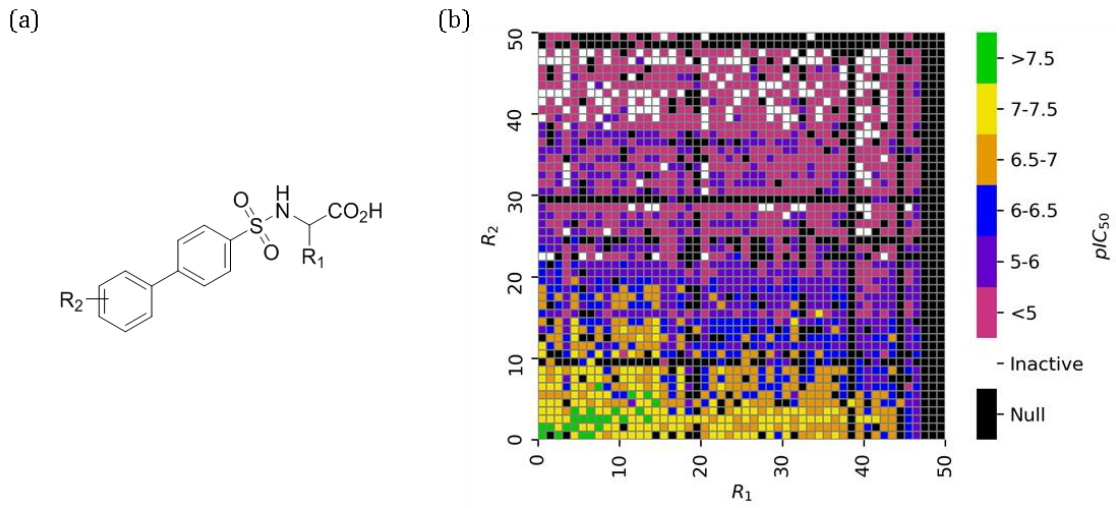


Figure 5.2. (a) The common biaryl sulfonamide scaffold in the MMP-12 dataset. (b) Heat map showing activities and the results of the 50x50 biaryl sulfonamide array proposed by Pickett *et al.* Some of candidates were not synthesised, not assayed or the assay failed, and these are classified as "Null" in the heat map (black cells). This figure is reproduced from (Pickett et al., 2011) where they sort the R_1 and R_2 groups based on the activity of the most potent compound containing that R group, and colour is used to indicate potency, with pIC_{50} values from <5 to >7.5 coloured from crimson, purple, blue, orange, yellow to green.

Out of the possible 2,500 biaryl sulfonamide candidates, 1,880 were synthesised with a measured IC_{50} value. The remaining 620 candidate molecules were either not made, had a failed biological assay, or not assayed at all.

The MMP-12 dataset was taken from the Supporting Information of Pickett *et al.* (Pickett et al., 2011). I used the same dataset cleaning process that Pyzer-Knapp used on the MMP-12 dataset (Pyzer-Knapp, 2018): for all compounds labelled '*inactive'* the activity was set to 0, while compounds labelled with *'assay failed'*, *'not assayed'* or *'not made'* were removed, which left 1,880 candidate molecules, each of which have a measured IC_{50} value. The pIC_{50}, that is the negative log IC_{50}, was used for the optimisations.

5.2.2 Malaria Dataset Preparation

The 20k hits are from a combination of whole cell assay datasets for *Plasmodium falciparum* (*P. falciparum*) from three separate sources: GlaxoSmithKline Tres Cantos

Antimalarial Set (TCAMS), Novartis-GNF Malaria Box Data set and St. Jude Children's Research Hospital's Dataset.

The public malaria dataset was downloaded from the Malaria Box supporting information from the Medicines for Malaria Venture website (Malaria Box supporting information, 2013) and contains 18,924 molecules with activity in micromolar EC_{50}. The pEC_{50}s, that is the negative log10 of the molar EC_{50}s, were used for the optimisations.

5.2.3 Running the Bayesian Optimisation

For all Bayesian optimisation calculations performed in this chapter, I used GPyOpt (version 1.2.5) with EI as the acquisition function, *optimise_restarts* set to 10, which specifies the number of restarts in the optimization, and *normalize_Y* set to *False*, which means the outputs are not normalized before performing the optimisation. The domain type was set to *'bandit'* which refers to the multi-armed bandits optimisation (see Section 5.1.7). For the RBF kernel, GPy (version 1.9.6) was used, with a *variance* of 1 and *lengthscale* of 5.

For the random searches, the Python package *random.sample* was used to sample randomly the candidate molecules. All Bayesian optimisation experiments were sequential *e.g.* each iteration picked one molecule to sample and observe.

For the MMP-12 dataset (Section 5.3.1), similar to Pyzer-Knapp, each optimisation run was seeded with three randomly selected molecules. For each optimisation, the maximum number of iterations was set to 400 and ten repeat runs were performed for each experiment. The number of iterations was chosen after observing that the majority

of the explored methods plateaued to the maximum in the evolution of best activity found (Figure 5.5).

For the malaria dataset (Section 5.3.2), optimisations were seeded with twenty randomly chosen molecules and ten repeat runs were performed for each experiment. Due to time constraints and the computational complexity being $O(n^3)$, I only ran the optimisations for 1,000 iterations. This corresponds to sampling only ~5% of the malaria dataset's total population. However, the calculations were performed on a machine running Fedora 30 and an Intel(R) Xeon(R) Gold 5118 CPU running at 2.30 GHz with 32 GB of RAM, 24 cores and this took ~6 hours per run; hence for ten repeats this would be equivalent to ~60 hours per experiment (Figure 5.12).

For investigation of vectorised RDKit pharmacophore fingerprints and PLIFs (Section 5.3.3 and 5.3.4 respectively), optimisations were seeded with five randomly chosen molecules and the maximum number of iterations set to 100, with the exception of optimisations involving TRY1 where the maximum number of iterations was set to 90, because GPyOpt's default number of anchor points is five and the TRY1 dataset only has 108 molecules. Ten repeat runs were performed for each experiment.

Currently there is no Tanimoto or inverse Manhattan kernel available in GPy so I implemented these. The Tanimoto kernel uses the SciPy (Jones et al., 2001; Virtanen et al., 2019) function *cdist* with *'jaccard'* as the metric to calculate the Tanimoto distance, $tani\_d$, and this was transformed into a similarity by using $1 - tani\_d$. For the inverse Manhattan kernel, *scipy.spatial.distance.cityblock* was used to calculate the Manhattan distance $d_{Man}(x, x^*)$, equation (5.7) and the covariance matrix was calculated using equation (5.8).

For the 2D molecular fingerprint descriptors, RDKit (version 2019.03.1) was used to calculate the MACCS fingerprints and Morgan fingerprints using with a radius of two and the number of bits were 512, 1024 or 2048 as stated in the results.

5.2.4 Evaluation of Method Performance

For each validation dataset, I evaluated the performance of the various Bayesian optimisation methods using three different plots:

1. Maximum activity found versus number of iterations;
2. Number of desirable molecules found versus number of iterations, for Section 5.3.1, or number of molecules within the top 10^{th} percentile versus number of iterations, for Sections 5.3.2, 5.3.3 and 5.3.4.
3. Distribution of activities for all points sampled during the optimisation.

A good method should find the maximum activity within the fewest number of iterations, find more desirable molecules within the same number of iterations, or find more molecules within the top 10^{th} percentile within the same number of iterations and also have a good sampling distribution where it has enhanced sampling of the active molecules.

5.2.5 Building Machine Learning Models

Another method of assessing the effect of the different molecular descriptors and different kernels is to treat the problem as a case of supervised machine learning. I investigated the same molecular descriptors and kernels as when running the Bayesian optimisations with different descriptors and kernels.

Machine learning models were built using Gaussian process regression (GPR) on the MMP-12 dataset to predict pIC_{50}s (Section 5.3.1.1). The GPR models were built using GPy (GPy, Version 1.9.6, 2012) and tested using five-fold cross validation. This meant that the 1,880 molecules were split into five equally sized subsets *i.e.* five groups of 376 molecules, and four of the five groups were used for training and the remaining group was used for testing. This process was performed five times *i.e.* five different models were built, where each time a different group was left out. Finally, the scores of each of the five models were combined into one score by taking the average. The five-fold cross validation was performed using scikit-learn's *Kfold* with *shuffle* set to *True*.

For the evaluation of the machine learning models, I computed the coefficient of determination, R^2, and the mean square error, MSE. The functions *r2_score* and *mean_squared_error* from *sklearn.metrics* were used to calculate the R^2 and MSE respectively.

For a given i^{th} molecule, let $y_{calc,i}$ be the machine learning model's predicted affinity and y_i the corresponding experimentally observed affinity for that molecule, then the MSE can be calculated as,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_{calc,i})^2 \tag{5.9}$$

where N is the total number of molecules or observation points.

The R^2 is calculated from three sum of squares formulas,

$$\text{Total Sum of Squares, } TSS = \sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{5.10}$$

$$\text{Explained Sum of Squares, } ESS = \sum_{i=1}^{N}(y_{calc,i} - \bar{y})^2 \tag{5.11}$$

$$\text{Residual Sum of Squares, } RSS = \sum_{i=1}^{N}(y_i - y_{calc,i})^2 \tag{5.12}$$

$$R^2 = \frac{ESS}{TSS} \equiv \frac{TSS - RSS}{TSS} \equiv 1 - \frac{RSS}{TSS} \tag{5.13}$$

Where $\bar{y}$ is the mean of the observed experimental data,

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{5.14}$$

5.2.6 Preparation of Datasets from PDBbind

PDBbind (Liu et al., 2015) is a database of protein-ligand complexes with experimentally measured binding affinity data. PDBbind includes the General Set and the Refined Set, where the former makes up the main body of the database and the latter only includes protein-ligand complexes with higher quality structural and binding data.

The PDBbind General Set (v2018) was downloaded and filtered for each target according to their UniProt ID: Beta-1-secretase (BACE1, UniProt ID P56817), Carbonic Anhydrase II (CAH2, UniProt ID P00918), Cyclin-dependent kinase 2 (CDK2, UniProt ID P24941), Bovine trypsin (TRY1, UniProt ID P00760). PDB structures were filtered out if they had a resolution ≥ 3Å. The data was split into if the binding data was IC_{50} or $K_{i/d}$ and the larger set taken forward. As it was possible that multiple PDB structures contained the same ligand, the dataset was uniquified according to the three-letter ligand code, keeping the structure with the best binding data (lowest IC_{50} or lowest $K_{i/d}$). If multiple PDBs had the same ligand, with the same

binding constant then the one with the highest resolution and more recent structure was kept.

The PDB structures were aligned using PyMOL's *align* function using the backbone atoms to the reference PDB for that DUD-E target, and the ligand was identified according to the three-letter ligand code and saved as a SDF file. RDKit's *GetMolFrags* function was used to identify if more than one ligand was present in the structure, and if so, the ligand that was closest to the reference ligand for that DUD-E target was saved as a SDF file. The closest was determined by calculating the distances between centroids using RDKit's *computecentroid* function. These ligand SDFs were then used to calculate the vectorised pharmacophoric features (Section 5.2.7).

For the studies involving PLIFs (Section 5.3.4), further processing of the PDB structures was performed. The complexes were renumbered according to the UniProt residue numbering using the UniProtKB/SwissProt database (Martin, 2005). For each structure, the binding site was defined as all residues within 6.5 Å of the ligand for more than 70% of the PDB structures. The binding site for all the PDBs were checked for mutated, missing or modified residues and if present, these structures were not used. For the remaining structures, *clean_pdb.py* (described in Section 2.2.7) was run to clean the PDBs before PLIFs were calculated by Arpeggio (Jubb et al., 2017). The same post-processing was performed as described in Chapter 2, Section 2.2.7.

5.2.7 Vectorised Pharmacophoric Features

For structural information to be used as the search domain in Bayesian optimisation, the information must be transformed into feature vectors. I explored the use of vectorised pharmacophoric features (Section 5.3.2) and PLIFs (Section 5.3.3) as feature vectors for

the domain in Bayesian optimisation. The latter already exists in the vector form; however, the former must be vectorised. In the following, I explain how I convert each ligand's RDKit 3D pharmacophoric features into a feature vector.

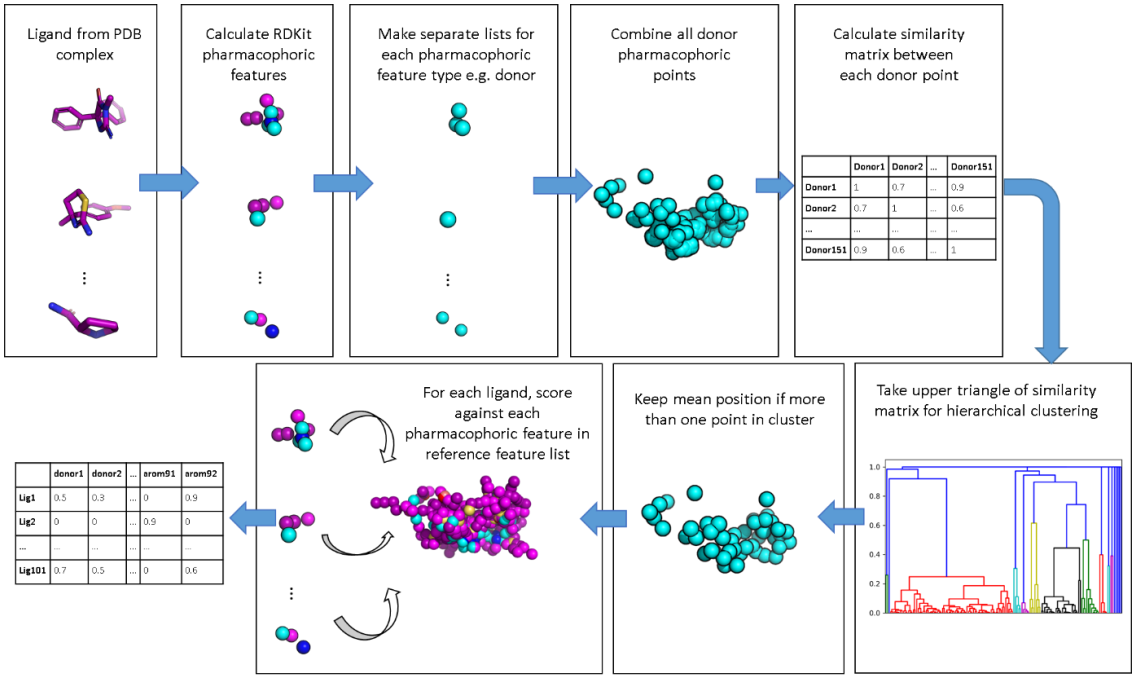


Figure 5.3. Schematic showing how the pharmacophoric features are vectorised for each ligand. Details of each step are described in the text.

For each ligand SDF, the pharmacophoric features were calculated using all eight RDKit pharmacophoric features, which are the same as the pharmacophoric features used in SuCOS (Chapter 3, Section 3.2.8) *i.e. Donor*, *Acceptor*, *NegIonizable*, *PosIonizable*, *ZnBinder*, *Aromatic*, *Hydrophobe*, and *LumpedHydrophobe*. For each pharmacophore type, a similarity matrix was calculated between all points, using the *ScoreFeats* function in RDKit. For example, for the donor pharmacophore type, all the donor pharmacophore points of all ligands were identified and a symmetric similarity matrix was calculated using *ScoreFeats* between all pairs of donor points. If *n* donor pharmacophore points were identified, this similarity matrix would have a shape of *n* by

n and all elements on the diagonal would be one *e.g.* donor point 1 and donor point 1 are identical and would overlap fully.

The upper triangle (*np.triu_indices* , *k*=1) of this similarity matrix was then used in hierarchical clustering using *linkage* function, from *scipy.cluster.hierarchy,* with the *average* method and a defined cutoff. In Section 5.3.3 I investigated using two different cutoffs, *t*=0.9 and 0.99 and in Section 5.3.4 only cutoff *t*=0.99 was used. One pharmacophoric point was kept from each cluster, and if there was more than one point in the cluster, then the mean x, y, z coordinates of that cluster was used. All these clustered pharmacophoric points for all pharmacophoric types were appended to a list and represent the columns of the vectorised pharmacophoric features. I call this list of pharmacophoric features as the reference pharmacophoric feature list.

Finally, each molecule's pharmacophoric features was then scored to each feature in this reference feature list using the *ScoreFeats* function in RDKit, which created a *numpy* array of shape (*n*, *m*) where *n* is the number of molecules and *m* is the number of features in the reference pharmacophoric feature list. This array was used in the Bayesian optimisation in Section 5.3.3. The overall workflow of creating the vectorised pharmacophoric features is shown in Figure 5.3.

5.2.8 ElectroShape

ElectroShape (Armstrong et al., 2010) is a non-superpositional ligand-based 3D molecular descriptor based upon the Ultra-Fast Shape Recognition algorithm, USR (Ballester et al., 2010). USR is a shape-comparison method that computes a set of shape-based descriptors for each molecule and calculates the distances between the descriptors for molecular similarity. The descriptor only includes shape information and

no information about the electrostatics of the molecule. ElectroShape builds upon USR by adding partial charge as the fourth dimension, and also encodes the chirality of the shape. For each molecule, four centroids are identified and the distance to each atom is calculated which creates four distributions. The mean, standard deviation and third root of the third central moment of each distribution is taken which results in twelve numbers. The three remaining numbers are derived from the partial charge dimension.

In Section 5.3.1, I describe the results from investigating using the ElectroShape descriptor as the search space. For each compound I generated 100 conformers using RDKit's *AllChem.EmbedMultipleConfs*, which uses the ETKDG method by default (Riniker and Landrum, 2015). For each conformer, an energy minimisation was performed using RDKit's *Minimize* function on the MMFF force field created with *MMFFGetMoleculeForceField*, with the default arguments (of maximum iterations set to 200), and then the conformer's energy was calculated using the *CalcEnergy* function. Finally, the ODDT package (Wójcikowski et al., 2015) was used to calculate the ElectroShape descriptor for the single lowest energy conformer of each molecule.

5.3 Results and Discussion

5.3.1 Bayesian Optimisation for Ligand-Based Screening: Using the MMP-12 dataset

Pyzer-Knapp did not investigate how the molecular representation or the choice of kernel affected the performance of the Bayesian optimisation, so in order to investigate whether these affect the efficiency of searching using Bayesian optimisation for ligand-based screening, I used two validation sets, the MMP-12 set and the malaria Box set,

which are the same datasets that Pyzer-Knapp investigated (Pyzer-Knapp, 2018). The former consists of 1,880 molecules with associated IC_{50} values, while the second consists of 18,924 molecules with associated EC_{50} values. The distribution of the 1,880 pIC_{50}s for the MMP-12 dataset ranges from 0 to 8, and is shown in Figure 5.4.

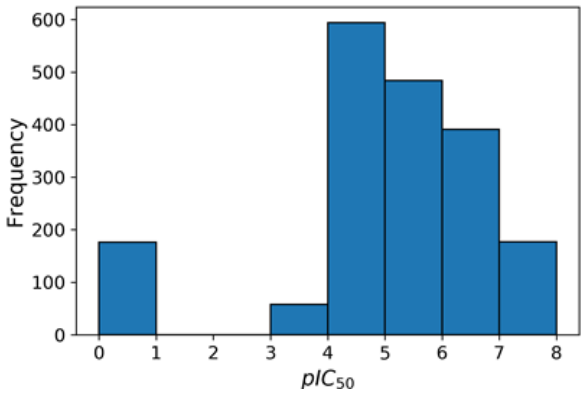


Figure 5.4. Histogram showing the distribution of the pIC_{50}s of the 1,880 compounds of the MMP-12 dataset.

For my investigations, the following Bayesian optimisation experiments were investigated:

(i) the effect of using an RBF kernel versus a Tanimoto kernel,

(ii) the effect of using a differing number of bits in the Morgan Fingerprint,

(iii) the effect of using a MACCS fingerprint versus a Morgan Fingerprint, and

(iv) the effect using a ligand-based 3D descriptor, specifically, ElectroShape.

Bayesian optimisation was run using the MMP-12 dataset with experiments (i)-(iv) using the methods outlined in Section 5.2.3. The results are shown in Figure 5.5, where the maximum pIC_{50} found for each optimisation, averaged over 10 runs, is shown against the iteration number.

Similar performance was found when using the RBF kernel compared to the Tanimoto kernel, with Morgan fingerprints, *nbits* set to 512 and both performed better than

random searching (Figure 5.5a). Interestingly, our RBF kernel result is better than the RBF result found by Pyzer-Knapp for the same kernel, same acquisition function, and same fingerprint, as they only found the maximum pIC_{50} after >800 iterations, compared to mine which took < 200 iterations. I used GPyOpt's implementation, but it was not clear how Pyzer-Knapp implemented their method. Random searching does not on average find the maximum pIC_{50} after 400 iterations but it is very close as it reaches a mean pIC_{50} value of 7.89.

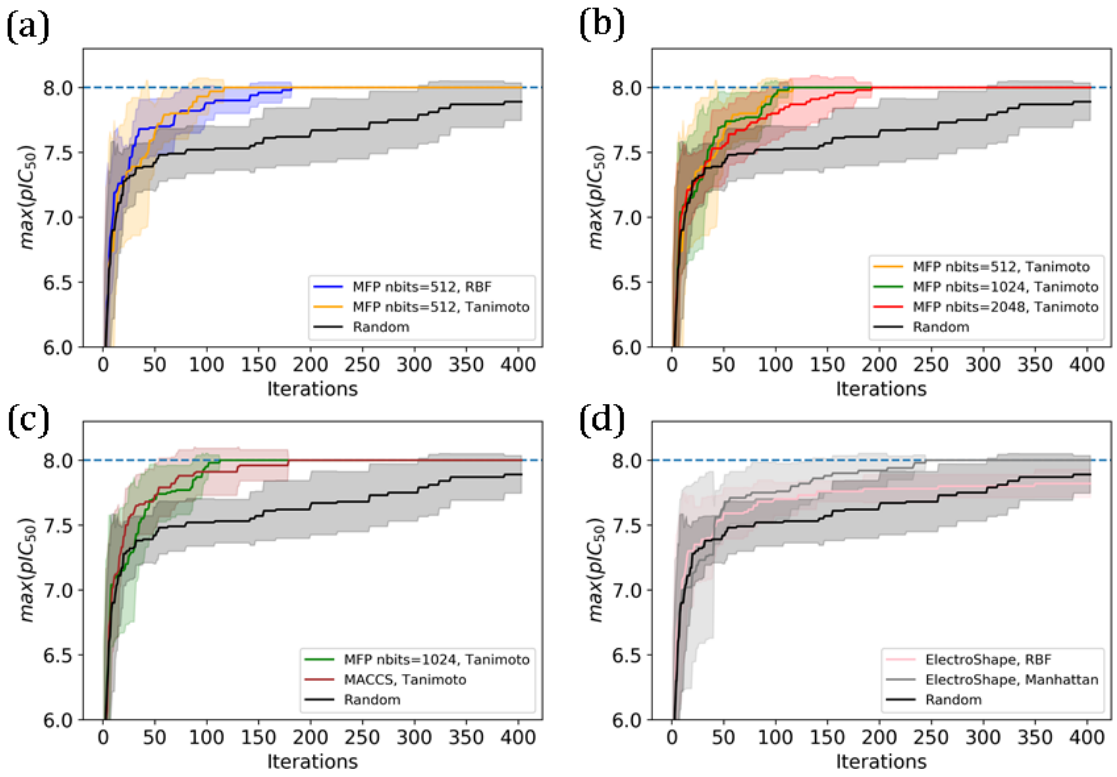


Figure 5.5. Bayesian optimisation was run on the MMP-12 dataset with different molecular representations and different kernels. The evolution of the best pIC_{50} found so far, averaged over 10 repeat runs, was plotted against the iteration number. The shaded areas represent ± 1 standard deviation from the average. The dotted line represents the maximum pIC_{50} in the dataset. (a) Comparison of the RBF kernel to the Tanimoto kernel, using Morgan Fingerprints, r=2, nbits=512. (b) Comparison of increasing the number of bits, *nbits*, used in the Morgan Fingerprint, r=2. (c) Comparison of using MACCS fingerprint versus Morgan fingerprints, r=2, nbits=1024. (d) Comparison of ElectroShape using the RBF kernel versus a Manhattan based kernel.

Next, to explore the effect of changing the number of bits on the Morgan Fingerprint, the optimisation was run using *nbits* = 512, 1024 and 2048 together with the Tanimoto kernel; the results are shown in Figure 5.5b. All had similar performance and all were

better than random. Thus for this dataset, the number of bits on the Morgan Fingerprint does not affect the results, which may be explained by the type of compounds in the dataset, as all the molecules have the same biaryl sulfonamide scaffold (Figure 5.2). The histogram of the Tanimoto similarity across the whole dataset shows similar distributions for the three different *nbit* values; hence this may explain the similar optimisation curves for all (Figure 5.6).

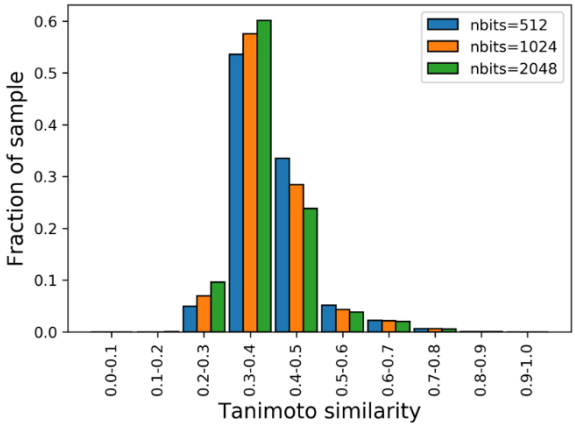


Figure 5.6. Histogram showing similar distributions of molecular similarity while varying the number of bits in the Morgan fingerprints. Tanimoto similarities were calculated across the 1,880 molecules in MMP-12 dataset, using RDKit Morgan fingerprint, *r*=2, and *nbit*=512 (blue), 1024 (orange), 2048 (green).

I also explored using MACCS fingerprints, which showed similar performance to Morgan Fingerprints *nbits*=1024 (Figure 5.5c). Again, both performed better than random. Interestingly, the distribution of Tanimoto similarities when using MACCS keys as the molecular representation shows a higher similarity distribution amongst the dataset than when using Morgan Fingerprint, MFP, as the molecular representation (Figure 5.6 versus Figure 5.7). One could argue that the MFP fingerprint should perform better than the MACCS keys, as the former encodes the environment of every atom and hence it should be more discriminative than the latter, which only captures the presence or absence of particular substructural groups. However, the similar performance of the two types of fingerprints could be explained by the composition of

the dataset; all the molecules share the same scaffold with varying R_1 and R_2 groups, hence the molecules are relatively similar. A validation on a different dataset, involving molecules with higher dissimilarity and not part of a 2D $n \times m$ synthesis array, could test this hypothesis. If true then the MFP should have better performance than the MACCS key. This is explored in the following ligand-based validation dataset (Section 5.3.2).

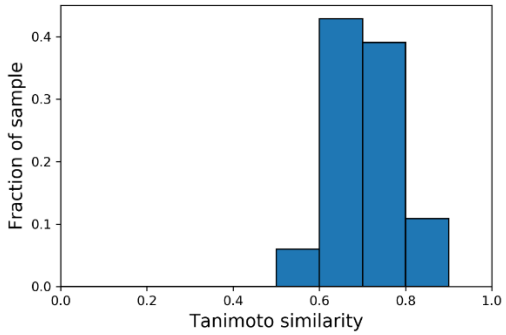


Figure 5.7. Histogram showing a higher similarity distribution for the MMP-12 dataset when the Tanimoto similarities were calculated using MACCS fingerprints instead of Morgan fingerprints.

Finally I investigated using ElectroShape (Armstrong et al., 2010). Using the methods described in Section 5.2.7, ElectroShape descriptors were generated for each compound and Bayesian optimisation was performed with the RBF kernel and the Manhattan kernel, Equation (5.8) and the results are shown in Figure 5.5d. I found ElectroShape with the Manhattan kernel performed better than with the RBF kernel. In fact, ElectroShape with the RBF kernel had similar performance to random sampling until ~300 iterations when it started to perform worse. The performance of ElectroShape with either kernel was also worse than the 2D fingerprint methods for finding the maximum pIC_{50} in the fewest number of iterations (Figure 5.5 a-c). One explanation for this may be due to the conformational space being too large to search, and/or the lowest energy conformer generated may not be the active conformer that is bound in the binding site.

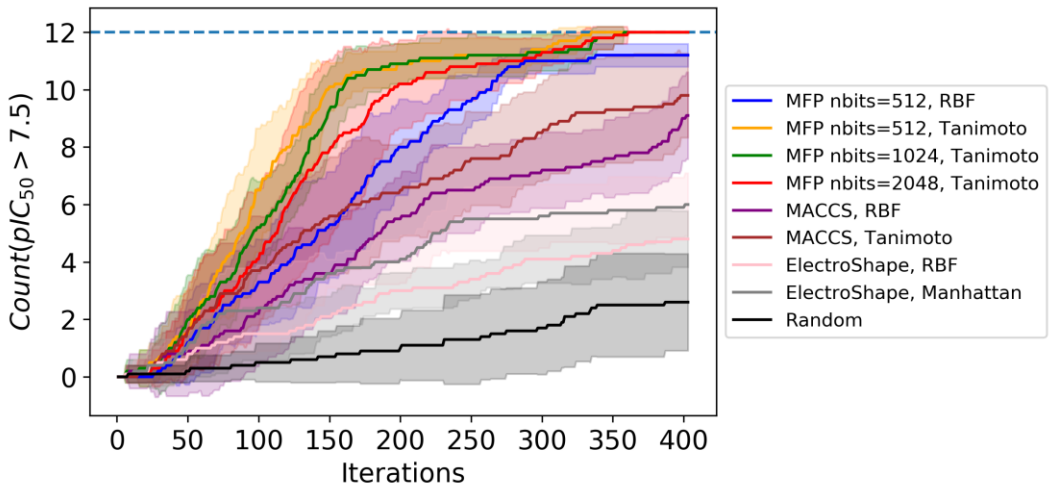


Figure 5.8. Comparison between the different methods for the recovery rate of desirable molecules. A desirable molecule is defined as one with a $pIC_{50} > 7.5$ and there are a total of 12 desirable molecules in this dataset, represented by the blue dotted line. The shaded regions show ± 1 standard deviation away from the mean of the ten repeat runs.

An alternative way of evaluating the performance of each method is to count the number of "desirable" molecules found as the number of iterations increases. Pickett *et al.* and also Pyzer-Knapp defined a *desirable* molecule for this dataset as one with a $pIC_{50} > 7.5$. There were 12 desirable molecules in this dataset. Figure 5.8 shows the number of desirable molecules found against the number of iterations for each method. Random sampling had the worst performance, with an average of 2.6 desirable molecules after 400 iterations. ElectroShape with the RBF and the Manhattan kernel performed slightly better than random, with an average of 4.8 and 6 desirable molecules after 400 iterations, respectively. The 2D fingerprint methods performed better. All methods using the Morgan fingerprint performed better than the MACCS methods. The best performer was the Morgan Fingerprint with the Tanimoto kernel, as they found all 12 desirable molecules by 400 iterations. Again, it is interesting to note that our method using MFP with an RBF kernel performed better than Pyzer-Knapp, as within 400 iterations, I found an average of 11.2 desirable molecules whereas they found ~5.

I also looked at the distribution of all the points sampled during each optimisation, ran for 400 iterations, for each method (Figure 5.9). The best method should have enhanced sampling of the more potent molecules and hence a good distribution would be skewed towards higher pIC_{50} values. As expected, random sampling and the overall distribution have similar proportions for each pIC_{50} bin, for example, they both have slightly more than 40% of molecules with $pIC_{50} < 5$. The summary of the performance of the different methods is analogous to that for the recovery rate of desirable molecules (Figure 5.8). Bayesian optimisation with ElectroShape, with either the RBF kernel or the Manhattan kernel, has poorer sampling distributions than the 2D fingerprint methods, but are still better than the random sampling distribution. MACCS fingerprints performed worse than MFP-based methods and the methods that show the best performance are those using the Morgan Fingerprint with the Tanimoto kernel and *nbits*=512,1024,2048. Once more, this highlights that the number of bits in the Morgan fingerprint does not change the results.

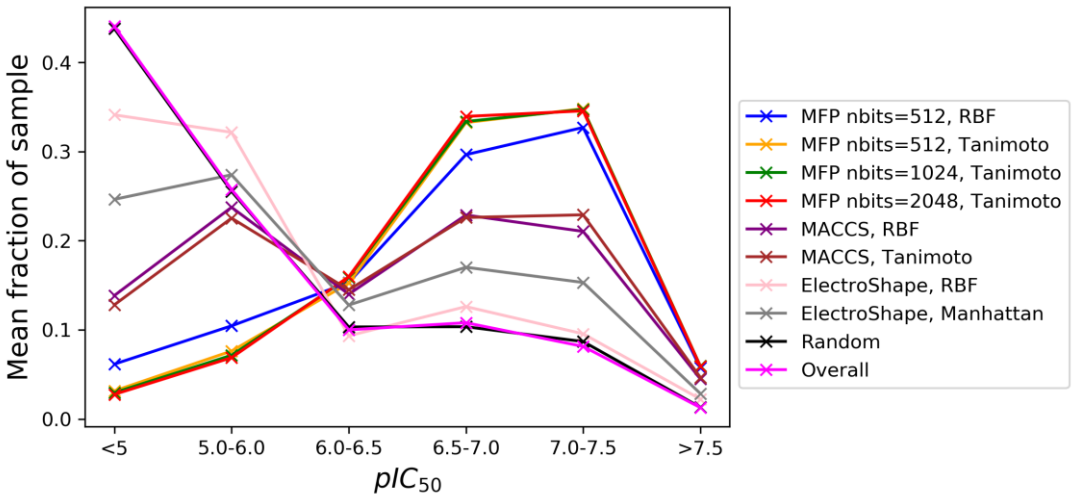


Figure 5.9. Distribution of all the points sampled during each optimisation for each method.

5.3.1.1 Gaussian Process Regression

As an alternative to the iterative sampling from chemical space, this problem can also be tackled from a machine learning perspective. This is another way of assessing how appropriate the representation space and kernel is. A Gaussian Process Regression model was trained with the MMP-12 dataset and the combination of features and kernels outlined in Table 5-1 (see Section 5.2.5 for methods). It was tested using five-fold cross-validation and the performance of the models was evaluated using the coefficient of determination, R^2, and Mean Squared Error, MSE, using the mean over the five-folds. Morgan Fingerprint methods performed the best, having the best mean R^2 and mean MSE (Table 5-1, first four rows).

Molecular representation	Kernel	Mean R^2	MSE
MFP, nbits=512	RBF	0.64	1.24
MFP, nbits=512	Tanimoto	0.63	1.28
MFP, nbits=1024	Tanimoto	0.63	1.27
MFP, nbits=2048	Tanimoto	0.64	1.26
MACCS	Tanimoto	0.34	2.29
MACCS	RBF	0.51	1.71
ElectroShape	RBF	0.27	2.53
ElectroShape	Manhattan	0.27	2.53

Table 5-1. A Gaussian Process Regression model was trained on the MMP-12 dataset, in a 5-fold cross validation model and the average R^2 and MSE values shown for each of the different methods I investigated.

5.3.1.2 What to make next in the MMP-12 series?

As mentioned in Section 5.2.1, the 1,880 molecules were synthesised as part of an effort to make a 50x50 array with biological testing. 620 candidates were either not made, not assayed or the assay failed (Figure 5.10).

Therefore, out of the 620 molecules that have no data, it would be interesting to know out of these molecules which one should be made next? The next suggestion can be

either: suggest the candidate with the highest predicted pIC_{50} (greedy, exploitive, short term gain) or suggest the candidate that will benefit the model in the long term (the next molecule in the iteration in the Bayesian optimisation).

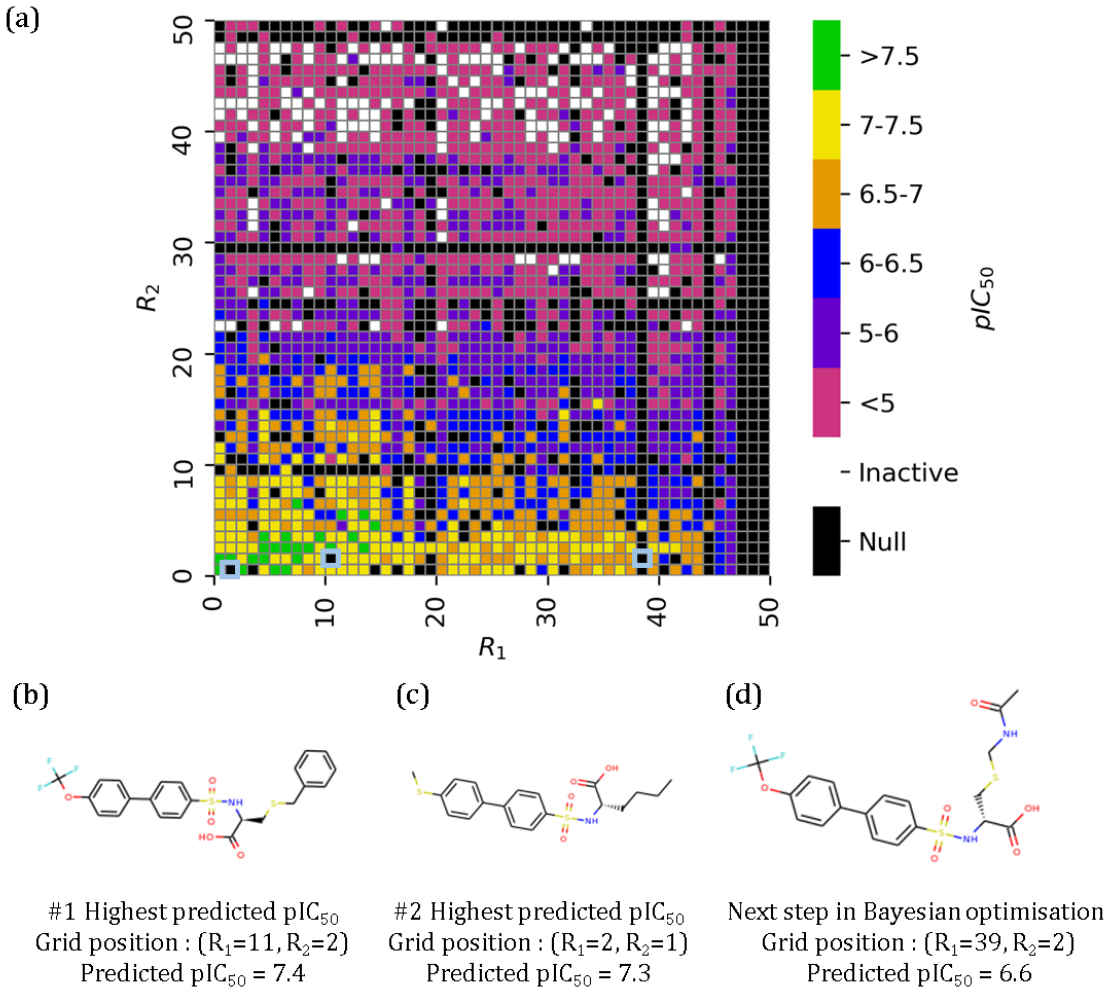


Figure 5.10. (a) Heat map showing activities and the results of the 50x50 biaryl sulfonamide array proposed by Pickett *et al.*The three molecules that are suggested by my methods are highlighted by the light blue boxes. (b)-(d) Molecules proposed to be made by the Bayesian optimisation and a GPR model trained on the 1,880 known candidates.

To generate the molecule that is suggested next by Bayesian optimisation *i.e.* the 1,881th iteration, Bayesian optimisation with EI acquisition function and Tanimoto kernel was initiated with all 1,880 known data points; Morgan fingerprints (r=2, *nbits*=512) and corresponding pIC_{50}s. It was run for one iteration, which picked the compound at (R_1=39, R_2=2) to make next, which has a predicted pIC_{50} value of 6.6

(Figure 5.10d). This candidate has a R_1 group which is not very well explored and only one candidate with the same R_1 has a measured pIC_{50} value. Therefore, this suggested candidate can be interpreted as one which would explore the optimisation landscape.

To generate the candidates with the highest predicted pIC_{50}, a GPR model was trained on all 1,880 data points which was then used to predict pIC_{50}s for the remaining 620 candidates with missing data. The candidates with the highest predicted pIC_{50} values are located at (R_1= 11, R_2= 2) and (R_1= 2, R_2= 1) and have predicted pIC_{50}s of 7.4 and 7.3 respectively (Figure 5.10 b and c). In contrast to the Bayesian optimisation prediction of what to make next, these two points have a high degree of data coverage in their respective rows and columns, *i.e.* for their matched molecular series, and hence there is more certainty in their prediction. These predicted pIC_{50}s are also higher than the one picked by the Bayesian optimisation process.

5.3.2 Bayesian Optimisation for Ligand-Based screening: Using the Malaria Dataset

As the Morgan fingerprint with the Tanimoto kernel showed the best performance with the MMP-12 dataset with regards to performance of the Bayesian Optimisations (Figure 5.8 and Figure 5.9), regardless of the number of bits in the fingerprint, I conducted similar investigations on a more 'complex' dataset for further validation.

This dataset involves compounds that target *Plasmodium falciparum*, a unicellular protozoan parasite that causes malaria in humans. The compounds have measured EC_{50} values and the range of pEC_{50}s for the dataset is shown in Figure 5.11. Bayesian optimisations were run using the methods described in Section 5.2.3. It is worth noting

that due to time constraints only 1,000 iterations were run, which is only a small proportion (~5%) of the entire dataset.

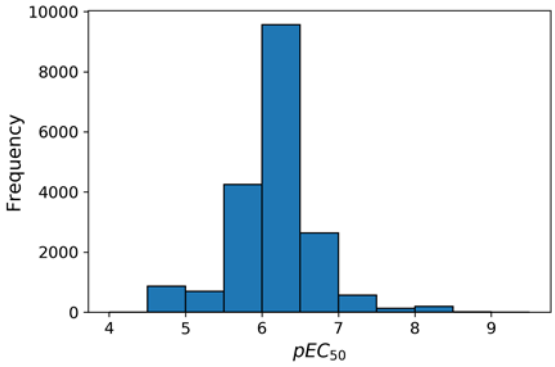


Figure 5.11. Histogram showing the distribution of the pEC_{50}s of the 18,924 compounds of the malaria dataset.

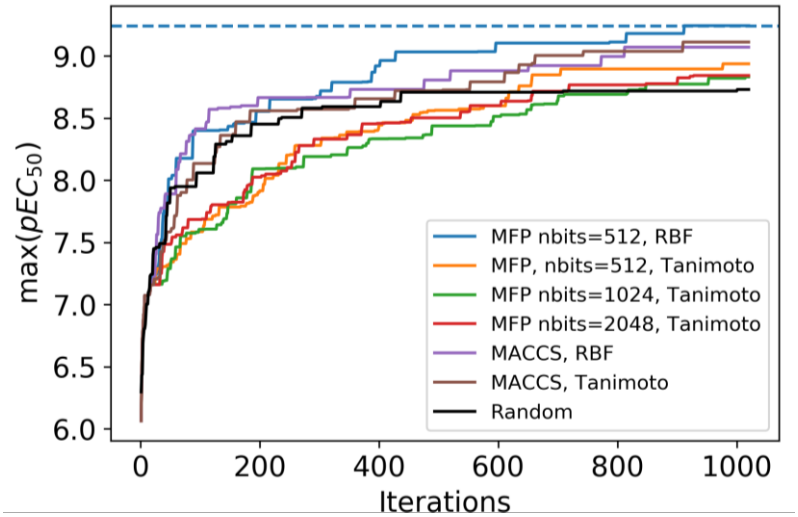


Figure 5.12. Evolution of the maximum pEC_{50} for the different Bayesian optimisation methods for the malaria dataset. The blue dotted line shows the maximum pEC_{50} of the dataset. The average over the ten repeat runs is shown for each method. For clarity, the ± 1 standard deviation errors from each average are not shown (see Appendix Figure C.1 for plot with ± 1 standard deviation errors).

The evolution of the maximum pEC_{50} found for each method is shown in Figure 5.12. Initially, random sampling performed better than or as well as the Bayesian optimisation methods. As the number of iterations increases, random sampling plateaus out, as it is not able to find a molecule with a greater pEC_{50}, whereas the Bayesian optimisation methods show continued improvement in pEC_{50}s. It is interesting that MFP with the RBF kernel performed marginally the best for this method of evaluation;

however, if ±1 standard deviation errors from the mean are considered (Appendix Figure C.1), the method performed similarly compared to the others.

Next, I looked at the retrieval of the top 10th percentile of molecules in terms of pEC_{50}; every Bayesian optimisation method clearly performs better than random sampling (Figure 5.13). Morgan fingerprint with the Tanimoto kernel and any number of bits investigated, outperformed the Morgan fingerprint with the RBF and MACCS keys with either the RBF or Tanimoto kernel. The latter three have similar performance for the retrieval of molecules in the top 10th percentile. This result contrasts with the plot of max(pEC_{50}) versus number of iterations shown in Figure 5.12, where Morgan fingerprint with the Tanimoto kernel does not have the best performance. The difference in the two figures highlights the different methods of evaluation. Morgan fingerprint with the Tanimoto kernel sampled more of the top 10th percentile *i.e.* showed good performance in Figure 5.12, but was not able to sample the molecules with highest activity *i.e.* poorer performance in Figure 5.13.

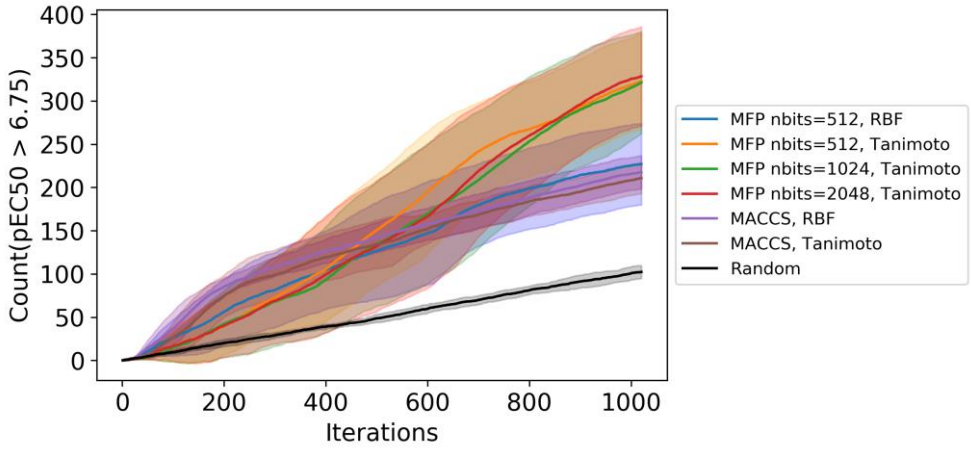


Figure 5.13. Recovery rate of the top 10th percentile ($pEC_{50} \geq 6.75$) of the malaria dataset for the various Bayesian optimisation methods, alongside random sampling.

These results involving the top 10th percentile (Figure 5.13) also align with the result obtained from the analogous plot involving desirable molecules in the MMP-12 dataset

(Figure 5.8). As discussed earlier, MACCS keys only encode the presence or absence of particular functional groups so it is expected to have worse results than a richer representation like the Morgan fingerprint, which accounts for every atom's environment.

Finally, the distribution of all molecules sampled after 1,000 iterations of either Bayesian optimisation or random sampling is shown in Figure 5.14. Again Morgan fingerprints with the Tanimoto kernel, shows the best distribution, as it has enhanced sampling of the more potent molecules. Morgan fingerprints with the RBF kernel and both the MACCS methods have worse distributions but still have better sampling distributions than random.

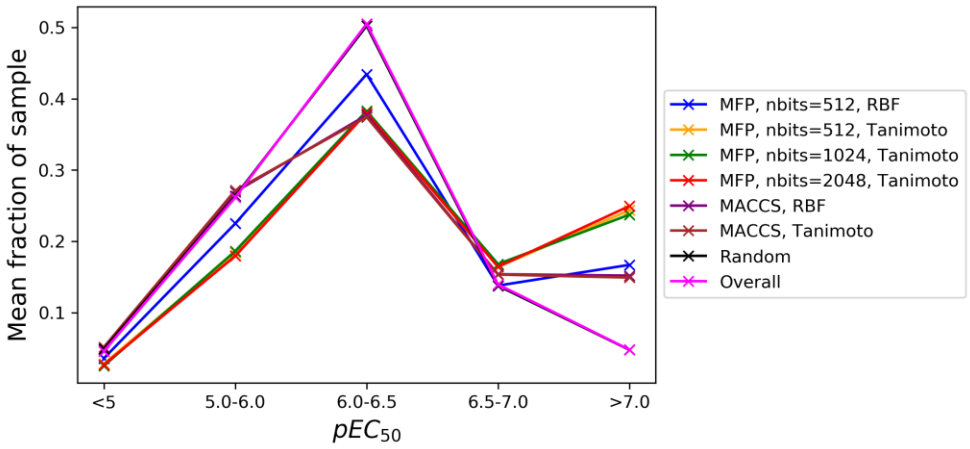


Figure 5.14. Distribution of all pEC_{50}s sampled during the Bayesian optimisations with the malaria dataset. Random sampling and the overall distribution is shown as a baseline comparison.

In conclusion, similar results were obtained from this malaria dataset as for the MMP-12 dataset, where the greatest differences between the methods are seen when plotting the recovery rate of the desirable molecules or the recovery rate of the top decile and also when plotting the distribution of all points sampled. From both plots for both datasets, Morgan fingerprint with the Tanimoto kernel performs best but no advantage can be seen for increasing the number of bits in the fingerprint beyond 512 bits.

Interestingly, the plot of the evolution of the maximum pEC_{50} found show that MFP with the RBF kernel marginally has the best performance; however, with overlapping one standard deviations.

As these optimisations with the larger malaria dataset are computationally more expensive than the smaller MMP-12 dataset, a more conclusive result could be made if a larger proportion of points were evaluated. This could be done by performing more iterations, or doing observations in batches, as Pyzer-Knapp did for this dataset.

5.3.3 Structure-Based Bayesian Optimisation using RDKit Pharmacophoric Feature Maps

So far, I have investigated using 2D molecular descriptors to describe the search space in the Bayesian optimisation. To the best of our knowledge, no one has investigated the use of a 3D descriptor as the Bayesian optimisation search space. In this section, I looked at using vectorised RDKit pharmacophoric feature maps as a 3D descriptor and used four diverse protein targets for validation, namely beta-1-secretase, BACE1; carbonic anhydrase II, CAH2; cyclin-dependent kinase II, CDK2; and bovine trypsin, TRY1. I chose these because they have abundant structural data.

Name	Target ID	UniProt ID	Number of structures	Activity data
Beta-1-secretase	BACE1	P56817	228	IC_{50}
Carbonic anhydrase II	CAH2	P00918	290	$K_{i/d}$
Cyclin-dependent kinase II	CDK2	P24941	169	IC_{50}
Bovine trypsin	TRY1	P00760	108	$K_{i/d}$

Table 5-2. Targets used to investigate vectorised pharmacophoric features as the Bayesian optimisation search space.

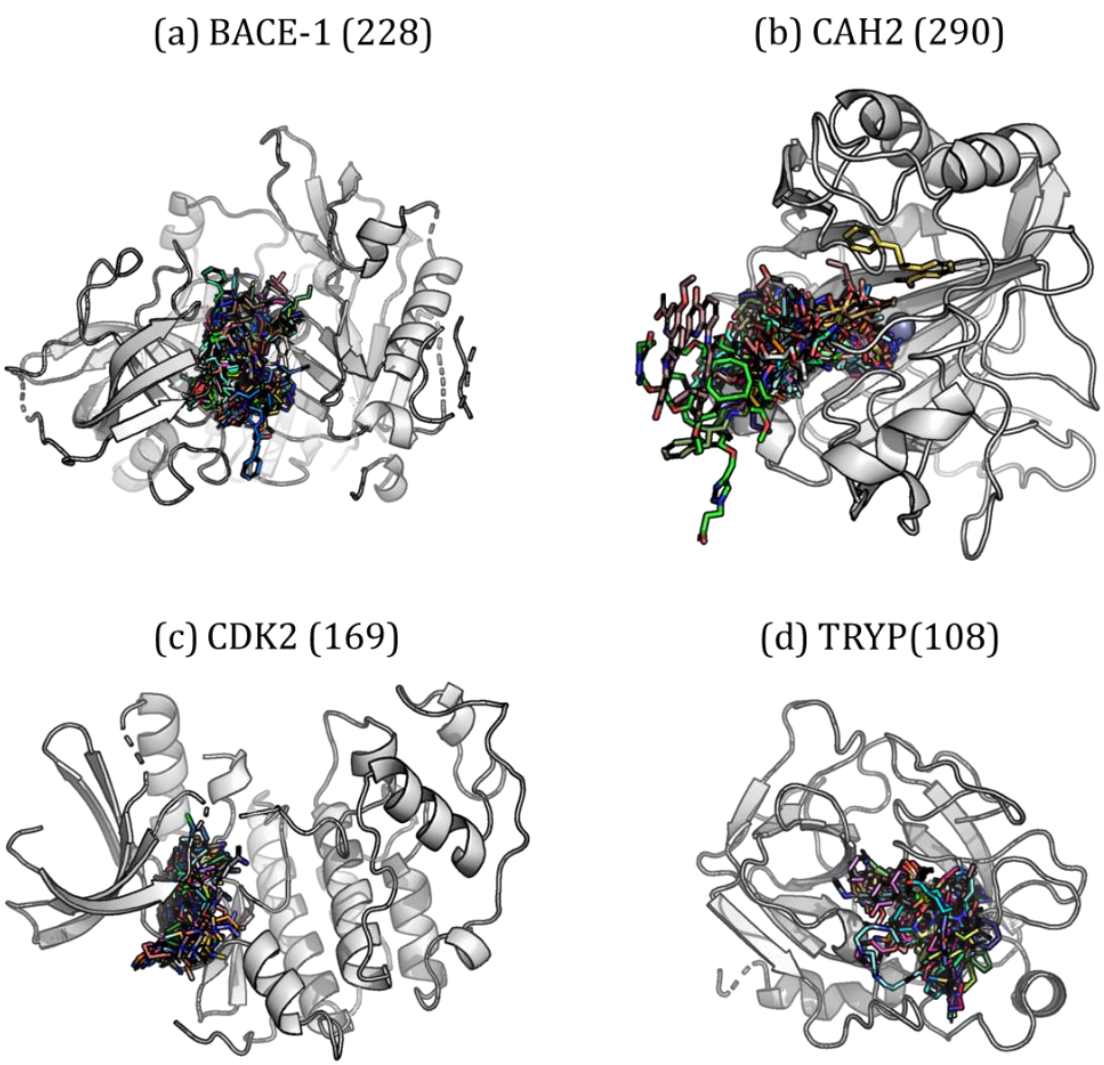


Figure 5.15. Targets used to investigate vectorised pharmacophoric features as the Bayesian optimisation search space. Numbers in brackets are the number of PDB structures used for each target. The receptor shown in the white cartoon is the representative PDB structure for the target according to the DUD-E dataset (Mysinger et al., 2012). The ligands are shown in stick representation.

The PDB structures were obtained from PDBbind (Liu et al., 2015) using the methods outlined in Section 5.2.6, to give 228, 290, 169 and 108 structures for BACE1, CAH2, CDK2 and TRYP1, respectively (Table 5-2 and Figure 5.15). The distribution of their activity data is shown in the histograms in Figure 5.16.

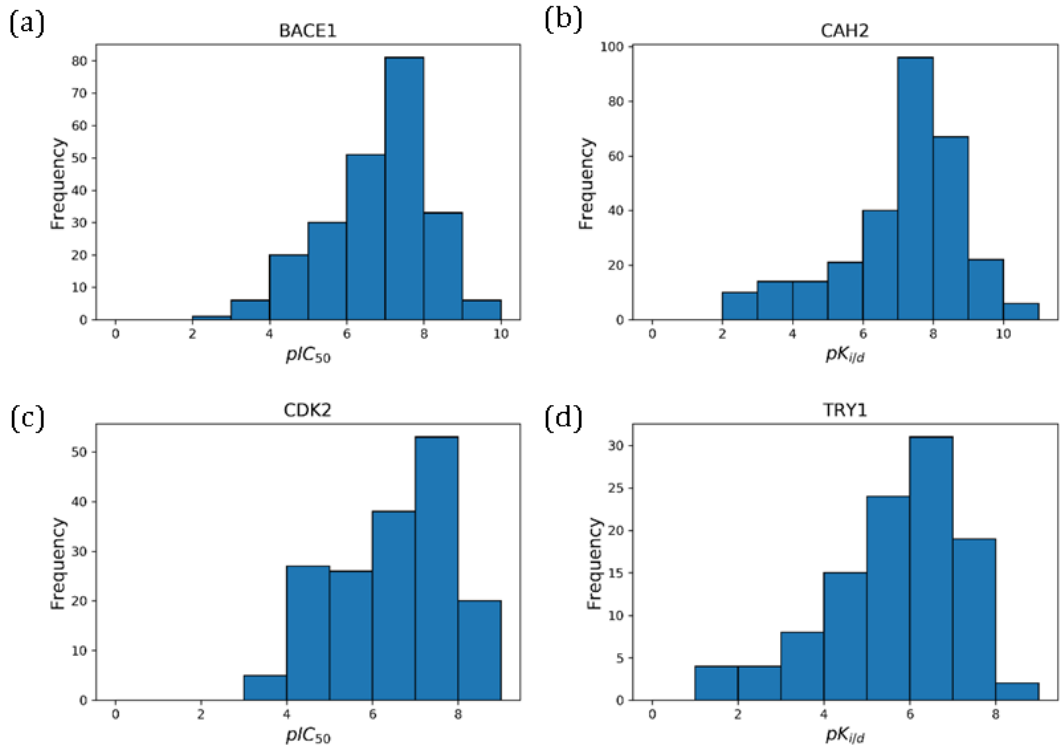


Figure 5.16. Histograms showing the distribution of activity data for the four targets; (a) BACE1; (b) CAH2; (c) CDK2; (d) TRY1; which were used to investigate Bayesian optimisation using vectorised pharmacophoric features as the search domain.

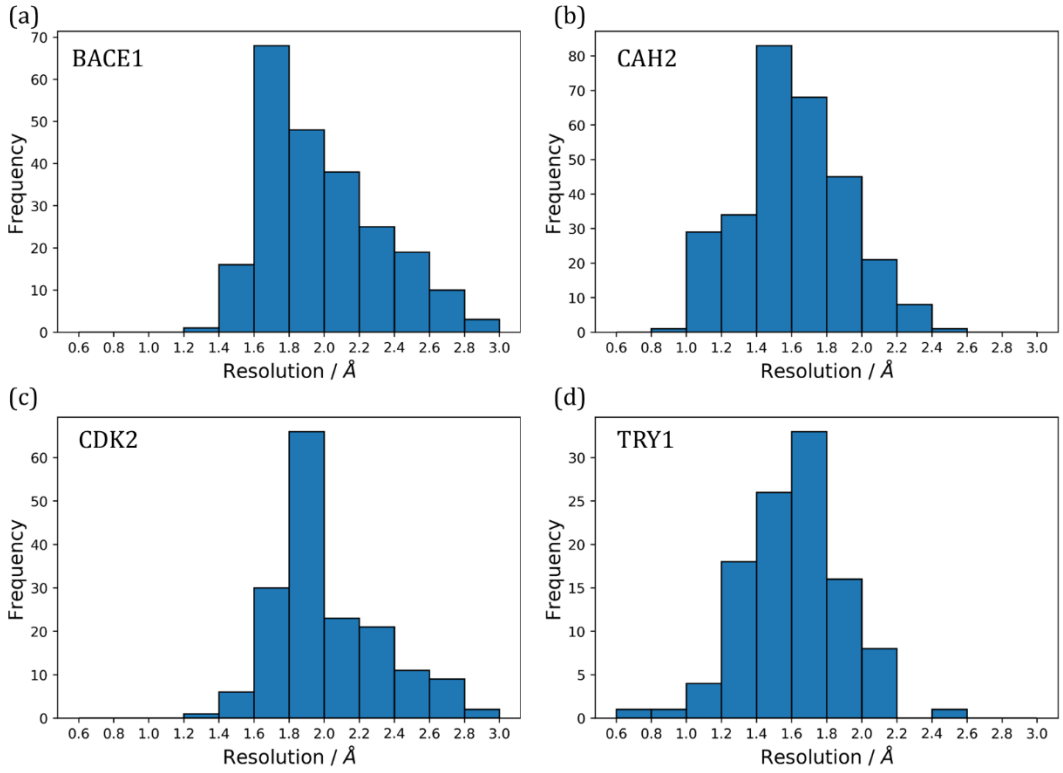


Figure 5.17. The distribution of the resolutions for the PDBs of the four targets; (a) BACE1; (b) CAH2; (c) CDK2; (d) TRY1; which were used to investigate Bayesian optimisation using vectorised pharmacophoric features as the search domain in Section 5.3.3.

The distribution of the resolutions of the PDB structures of the four different targets are shown in Figure 5.17. The proportions of PDBs with a resolution greater than 2.5 Å are 10% (22/228), <1% (1/290), 9% (15/169) and 0% (0/108) for BACE1, CAH2, CDK2 and TRY1 respectively. The relatively high proportion in the BACE1 and CDK2 dataset may be concerning; hence future work may involve repeating the analysis for that BACE1 and CDK2 and leaving out the PDBs with a resolution greater than 2.5 Å.

The following experiments were run on the four targets, studying the:

(i) effect of using the RBF kernel versus a Tanimoto kernel,

(ii) effect of using a MACCS fingerprint versus a Morgan Fingerprint,

(iii) effect of using vectorised RDKit pharmacophoric features, and

(iv) the use of two different thresholds (t=0.9 and t=0.99) for the hierarchical clustering to generate the reference pharmacophoric feature list (see Section 5.2.7).

The evolutions of the maximum activity found during all the optimisations for each target are shown in Figure 5.18. In contrast to the results for the MMP-12 and malaria dataset (Section 5.3.1-5.3.2), for BACE1 and CAH2 there is no clear advantage to performing Bayesian optimisation with most of the investigated molecular representations and kernels in comparison to random sampling. For example, MFP with the Tanimoto kernel performed very similarly to random for both targets. For CDK2, the investigated Bayesian optimisation methods performed similarly well and marginally better than random. For TRY1, all investigated Bayesian optimisation methods apart from MFP with Tanimoto kernel performed better than random. Across all four targets, the vectorised pharmacophoric features with the two different cutoffs performed similarly to each other.

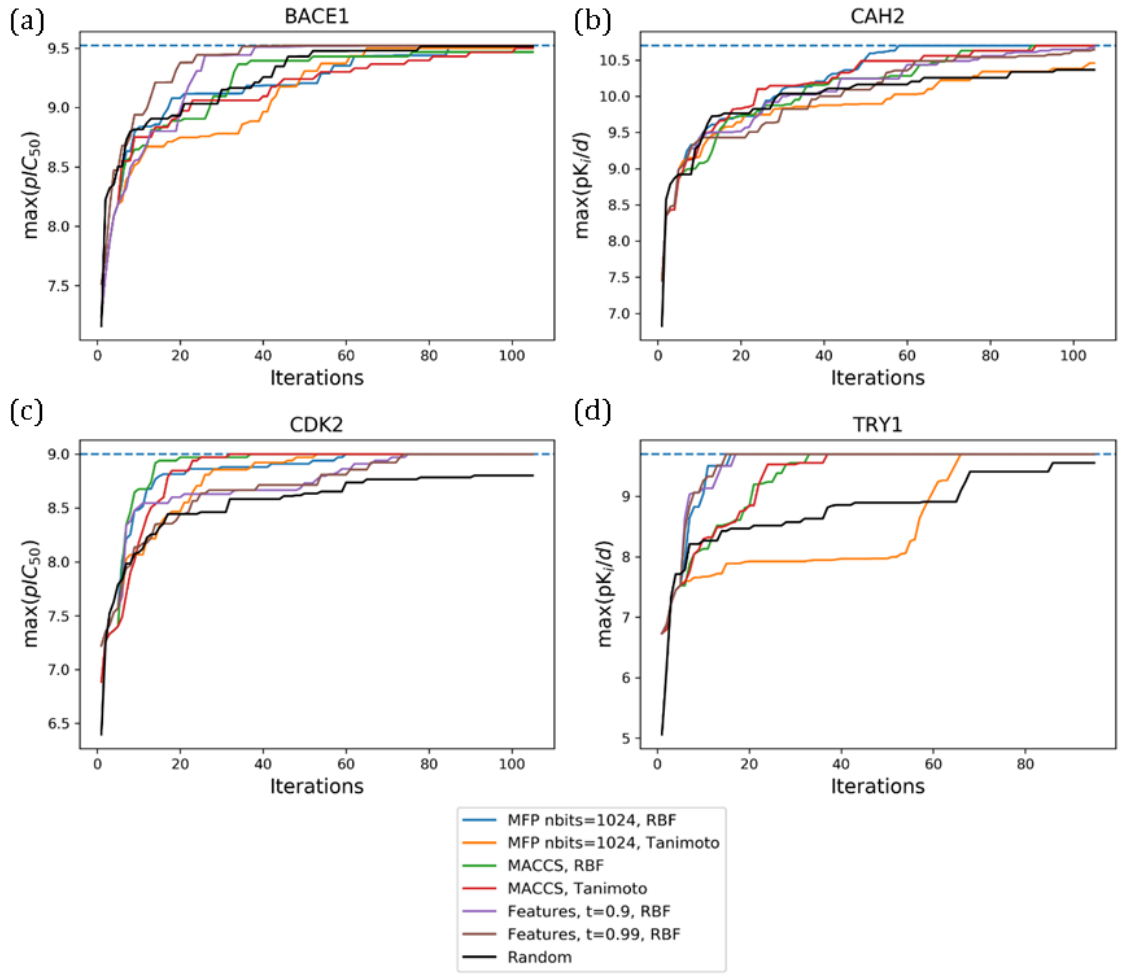


Figure 5.18. Evolution of the maximum pIC_{50} or $pK_{i/d}$ found with iteration number for the four targets, (a) BACE1; (b) CAH2; (c) CDK2; (d) TRY1; which were used to investigate Bayesian optimisation using vectorised pharmacophoric features as the search domain. The blue dotted line represents the maximum $pK_{i/d}$ or pIC_{50} for that dataset. For clarity, the ± 1 standard deviation errors from the average are not been shown (see Appendix Figure C.2 for plot with ± 1 standard deviation errors).

I also compared the different methods in terms of the recovery rate of the top decile of molecules (Figure 5.19). This analysis reveals a greater difference between the Bayesian optimisation methods and random sampling. For all targets, random sampling has the poorest recovery rate of the top decile. For BACE1, the Morgan fingerprint with either RBF kernel or Tanimoto kernel showed slightly better performance than the rest. For example, after 100 iterations, Morgan fingerprint with the RBF kernel was able to find an average of 19.5 molecules within the top 10^{th} percentile whereas using the vectorised RDKit pharmacophoric features only found an average of 15.

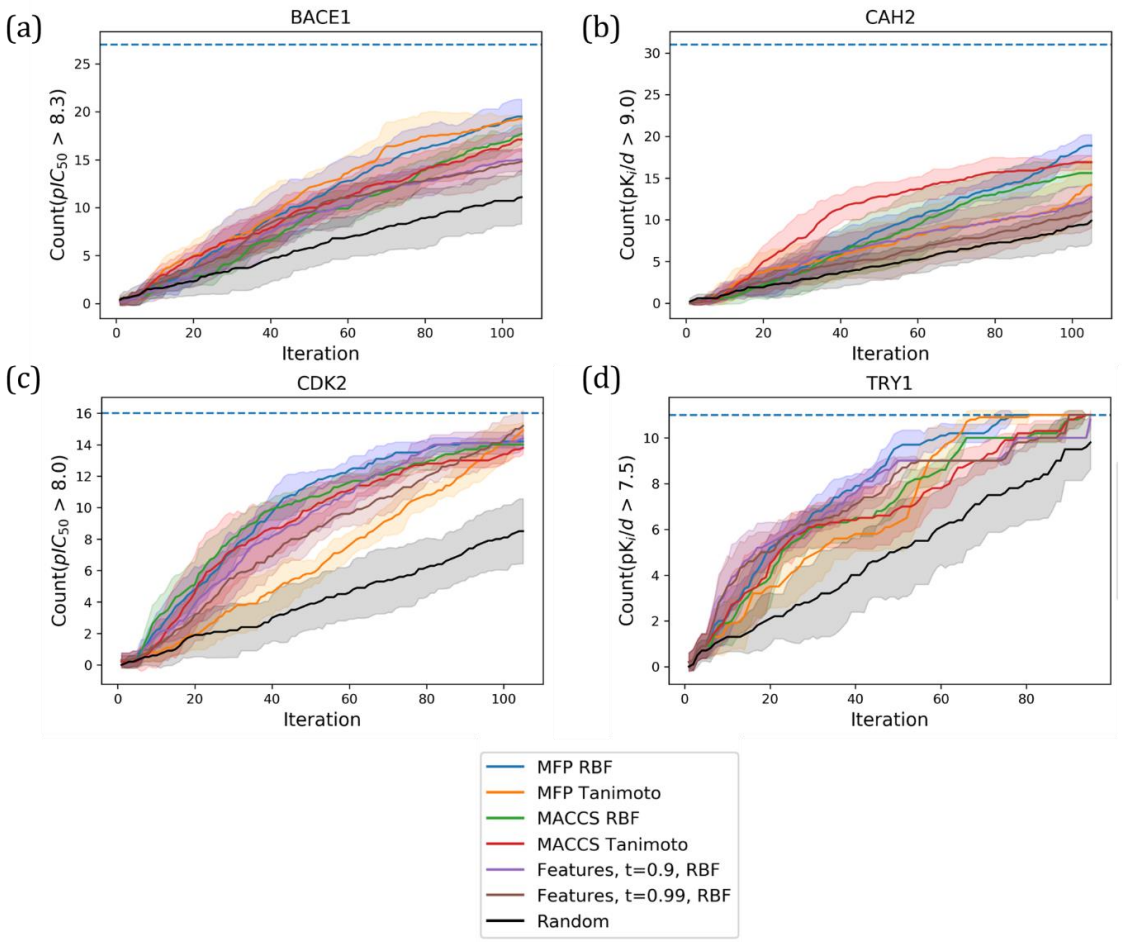


Figure 5.19. Recovery rate of top decile most potent compounds for each target: (a) BACE1; (b) CAH2; (c) CDK2; (d) TRY1. For BACE1, CAH2, CDK2, TRY1 the compounds in the top decile were those with $pIC_{50} \geq 8.3$, $pK_{i/d} \geq 9.0$, $pIC_{50} \geq 8.0$ and $pK_{i/d} \geq 7.5$, respectively. The blue dotted lines show the total number of molecules in the top decile for each target.

For CAH2, the vectorised RDKit pharmacophoric features did not perform much better than random and out of the 2D fingerprint methods, Morgan fingerprint with the RBF kernel and MACCS with the Tanimoto kernel performed slightly better, but with overlapping one standard deviation error bars. However, all the Bayesian optimisation methods were mostly similar but better than random search.

For CDK2, there is a more obvious separation between random sampling and the various Bayesian optimisation methods. Morgan fingerprints with the Tanimoto kernel performed slightly worse than the other Bayesian optimisation methods which

performed similarly well to each other. Again, all investigated Bayesian optimisation methods performed better than random sampling.

For TRY1, the dataset is smaller than the other three, hence after 90 iterations, most of the samplers were able to find all or nearly all the molecules in the top decile. Again with this target there is not much difference between the different Bayesian optimisation methods; however, again all were able to perform better than random.

In summary, Figure 5.19 shows very similar performance for the two different thresholds used to create the vectorised RDKit pharmacophoric features and in contrast to the results of the previous two ligand-based validation datasets involving inhibitors of MMP-12 (Section 5.3.1) and effective compounds against malaria (Section 5.3.2), the MFP with a Tanimoto kernel does not show consistently better performance than MFP with a RBF kernel or either of the investigated MACCS methods, across these four targets. Furthermore, there is no clear, consistently best method across the four targets for this method of evaluation.

The distributions of all points sampled for each target are shown in Figure 5.20. The distributions are shown after a different number of iterations for each target, as the size of each dataset is different. For the larger two datasets, BACE1 and CAH2, the plots show the distribution of all points sampled after 100 iterations. For the smaller datasets, CDK2 and TRY1, the plots show the distribution after 80 and 50 iterations, respectively.

For BACE1, there is enhanced sampling of the more potent molecules for all Bayesian optimisation methods, in comparison to random sampling. All Bayesian optimisation methods have similar sampling distributions, with the Morgan fingerprint methods,

using either the `RBF kernel or Tanimoto kernel, having marginally the best sampling distribution, agreeing with their marginally better performance seen in Figure 5.19.

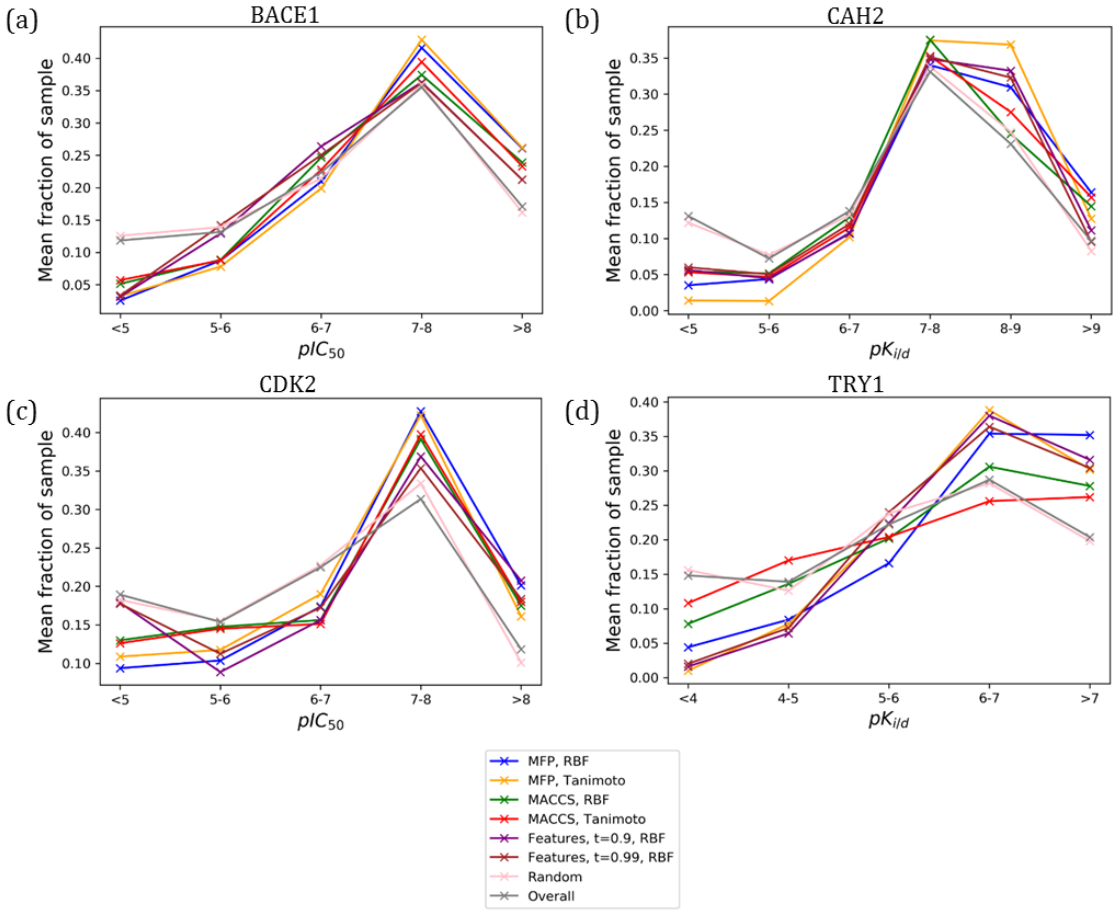


Figure 5.20. Mean distributions of all points sampled during each optimisation for (a) BACE1; (b) CAH2; (c) CDK2; and (d) TRY1. For the larger datasets, BACE1 and CAH2, the distributions are shown after 100 iterations. For the smaller datasets, CDK2 and TRY1, the distributions are shown after 80 and 50 iterations, respectively.

For CAH2, and for CDK2, all methods perform similarly well and better than random sampling in terms of enhanced sampling of more potent molecules. For CDK2, Morgan fingerprints performed marginally better than the other methods, with a greater proportion of molecules sampled with a $pIC_{50} > 7$.

For TRY1, MACCS fingerprints show the worse sampling distribution out of all Bayesian optimisation methods. Arguably, Morgan fingerprint with the RBF kernel has

the best sampling as it sampled the largest proportion of molecules with $pK_{i/d} > 7$, which also agrees with its good performance seen in Figure 5.18 and Figure 5.19.

Across the four targets, all Bayesian optimisation methods have consistently enhanced enrichment of potent molecules over random sampling. Morgan fingerprints have consistently better sampling distributions than MACCS fingerprints for all targets. The vectorised RDKit pharmacophore features made with the two different thresholds also show similar performance to each other; however, the performance of these 3D descriptors shows no consistent superiority over the 2D fingerprint methods.

In conclusion, from these four targets, no clear advantage can be seen for using vectorised RDKit pharmacophoric features over 2D fingerprint methods such as Morgan fingerprint.

5.3.4 Structure-Based Bayesian Optimisation: Using the CAH2 Dataset to Explore PLIFs and RDKit pharmacophoric Feature Maps

In this section, I investigated using protein-ligand interaction fingerprints (PLIFs) as the Bayesian optimisation exploration space. Carbonic Anhydrase II, CAH2, was chosen, as the first case study for PLIFs as it has the largest amount of structural data out of the four targets used in the previous study (Table 5-2).

The CAH2 PDB structures were downloaded and prepared as described in methods Section 5.2.6. This resulted in 265 CAH2 PDB structures, which have a distribution of $pK_{i/d}$ values as shown in Figure 5.21 and a distribution of resolutions shown in Figure 5.22. It should be noted that these 265 structures are a subset of the CAH2 dataset used to investigate RDKit pharmacophoric feature maps in Section 5.3.3, which involved 290 structures (Table 5-2).

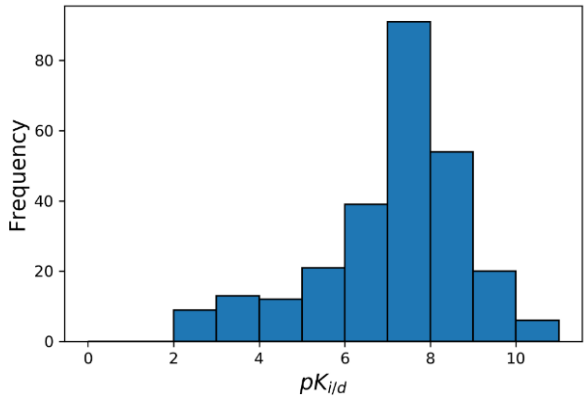


Figure 5.21. Histogram showing the distribution of the $pK_{i/d}$ values of the CAH2 target from the PDBbind general set used for the investigation of PLIFs as the Bayesian optimisation search space.

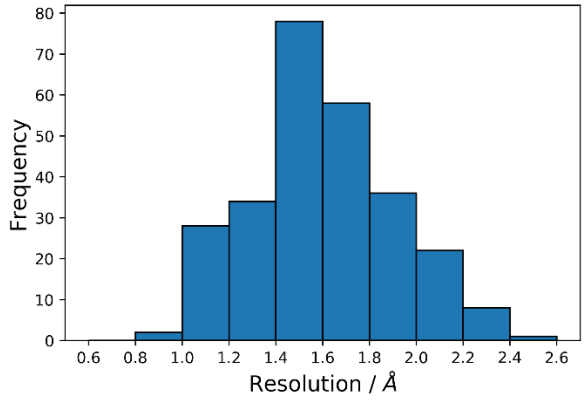


Figure 5.22. . The distribution of the resolutions for the PDBs in the CAH2 dataset set used in Section 5.3.4 for the investigation of PLIFs as the Bayesian optimisation search space.

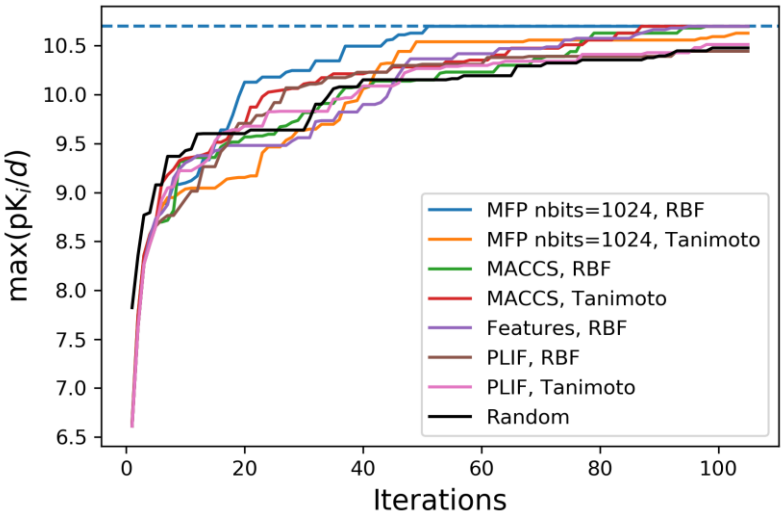


Figure 5.23. Bayesian optimisation was run with the various methods on the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space. For clarity, the ±1 standard deviation errors from the average have not been shown (see Appendix Figure C.3 for plot with ±1 standard deviation errors).

The plots of the evolution of the best $pK_{i/d}$ found show that most of the Bayesian optimisation methods investigated performed similarly to random sampling, when taking their standard deviations into account (Figure 5.23 and Appendix Figure C.3).

However, the results for the recovery rate of the top 10% most potent molecules show that random search clearly performed the worst (Figure 5.24). The difference in Figure 5.21 and Figure 5.24 could be explained by the Bayesian optimisation methods frequently choosing molecules in the top decile of the $pK_{i/d}$ distribution, but not one with a $pK_{i/d}$ that is better than what was previously sampled. This could be because the tightest binders have similar 2D molecular fingerprints to those within the top decile (*i.e.* a flat basin in the optimisation landscape). The methods that perform marginally better are MACCS fingerprints with either the Tanimoto or RBF kernel, PLIF with the Tanimoto kernel and MFP with the RBF kernel. MACCS fingerprints with the Tanimoto kernel showed the best performance up to ~60 iterations.

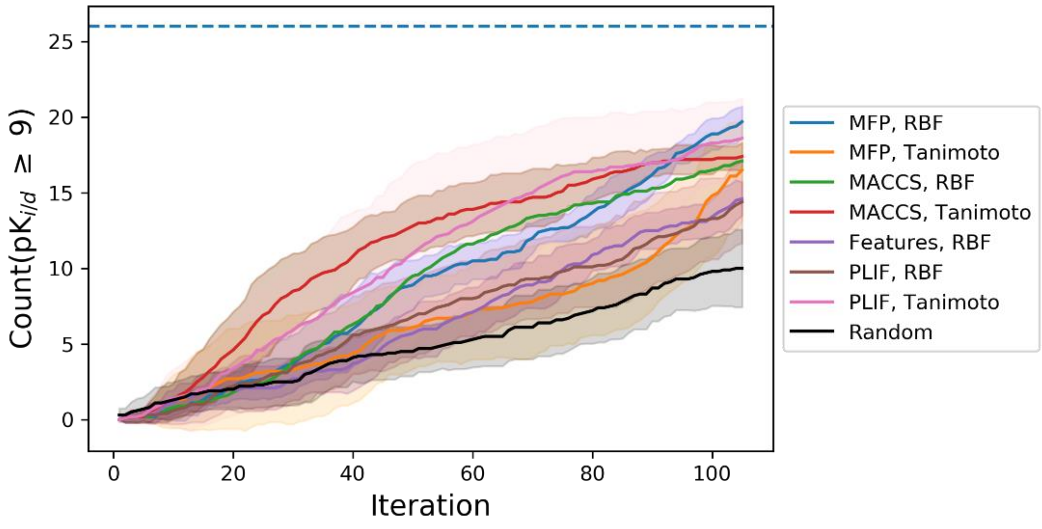


Figure 5.24. Recovery of the top decile most potent compounds ($pK_{i/d} \geq 9$) in the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space. The blue dotted line shows the total number of molecules in the top decile.

The distributions of all points sampled for all Bayesian optimisation methods show that all have enhanced sampling of the tighter binders, when compared to random sampling

or the overall distribution (Figure 5.25). The methods with marginally the best distributions are MFP with either the RBF kernel or the Tanimoto kernel. The MACCS methods perform marginally worse than the MFP methods.

In conclusion, this study on PLIFs as a potential 3D molecular descriptor for Bayesian optimisiation showed that PLIFs were not clearly any better than 2D fingerprints, such as the Morgan fingerprint. However, this is just one case study and better conclusions may be drawn if more targets were used in validation.

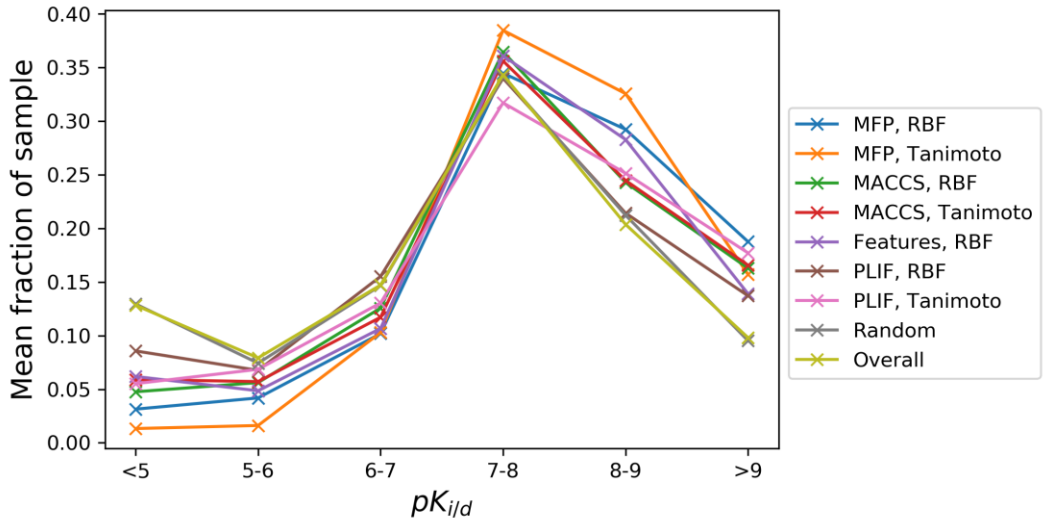


Figure 5.25. Distribution of all the $pK_{i/d}$ values found during each optimisation, in the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space. The overall distribution is shown in lime green which is similar to the distribution obtained from random sampling.

5.4 Conclusions

In this chapter, I have explored using Bayesian optimisation for ligand-based and structure-based virtual screening. My focus has been on comparing the performance of different molecular descriptors and different kernels. I measured the performance of each method by looking at the evolution of the most potent or active molecule found, the recovery rate of the top decile most potent compounds, or for the MMP-12 dataset,

the recovery rate of desirable molecules and the distribution of activities of all molecules sampled.

The first ligand-based validation dataset involved inhibitors for a single target, matrix metalloproteinase 12, MMP-12 (Pickett et al., 2011). My results showed a more efficient search performance than Pyzer-Knapp using the same molecular descriptor, same acquisition function and same kernel. One explanation could be the difference in implementation; I used GPyOpt to perform the Bayesian optimisation, whereas it was not clear how Pyzer-Knapp implemented their method. I expanded on his investigation by exploring alternative kernels and molecular descriptors and found that Morgan fingerprints with the Tanimoto kernel performed best and its performance was not significantly affected by increasing the number of bits in the fingerprint. MACCS keys showed worse performance than Morgan fingerprints and the ODDT implementation of ElectroShape performed worse than these 2D fingerprint methods for MMP-12. However, the method of conformer generation could be suboptimal; for example, future work could look at docking the candidates into the MMP-12 binding pocket and using the docked conformer for the ElectroShape calculation.

I also investigated using the malaria dataset (Spangenberg et al., 2013) as a ligand-based validation. The results agree with those from the MMP-12 dataset, where Morgan fingerprint with the Tanimoto kernel showed best performance compared to the rest. However, it should be noted that due to time constraints only a small proportion, ~5%, of the dataset was sampled and further work should either increase the number of iterations, or perform the optimisation in batches so that we can be more certain of the results.

Pyzer-Knapp also benchmarked the Bayesian optimisation methods against a greedy algorithm, which is something I could also explore in future work, in order to compare the various methods I explored against a greedy search.

For Bayesian optimisation using structural data, I studied two different 3D descriptors. One was a vectorised RDKit pharmacophoric feature fingerprint, and the second was protein-ligand interaction fingerprints (PLIFs). I devised the first vectorised representation myself and tested it on four different target datasets from the PDBbind database. However, they did not show any improvement over the 2D ligand-based fingerprint methods. This was further confirmed in my investigation of PLIFs with binders of CAH2, as PLIFs did not show any improvement over the 2D ligand-based fingerprint methods. Furthermore, in contrast to the result of the two ligand-based validation sets, for these structure-based validations, amongst the 2D fingerprint methods, the Morgan fingerprint with Tanimoto kernel did not consistently show better performance than the other 2D methods.

It is worth noting that the amount of data used in the structure-based validation datasets was much less than the amount of data present in the ligand-based validation datasets. Use of larger structure-based validation datasets would be ideal, however, the amount of publically available structure-based data with binding data annotations is limited *e.g.* the size of PDBbind. If more structural data were available, the Bayesian Optimisation experiments in the structure-based studies could be run for more iterations and a clearer separation between the different methods might be observed.

For the structure-based validations it would be interesting to compare the vectorised 3D RDKit pharmacophore features with a 3D pharmaocophoric fingerprint generated by inter-feature topological distances, such as Ligity (Ebejer et al., 2019), as my method

requires alignment of molecules in the binding site, whereas the latter does not. Moreover, investigation of a vectorised *shape* or a combined *shape and pharmacophore* molecular descriptor would be interesting, which follows on from Chapter 3.

For the investigation of PLIFs as the domain space, as I only looked at one target, validation with more targets is necessary to draw a firm conclusion. Also, the comparison of alternative PLIF generation tools would be interesting, such as the recently report Protein–Ligand Extended Connectivity, PLEC, which encodes protein-ligand interactions by pairing the ECFP environments from the ligand and the protein (Wójcikowski et al., 2019).

As mentioned in the introduction to this chapter, we envision using Bayesian optimisation to prioritise follow-up candidates following a fragment screening campaign. If 3D molecular descriptors were to be used, then a method of generating the chemical structures and their conformers would be required. In this chapter, I only looked at candidates that already have crystal structure data. Hence, the next validation could involve 3D molecular descriptors computed from the poses of docked candidates. For this case, the docked poses of DUD-E validation dataset generated by Koes *et al.* would be a useful starting point (DUD-E docked poses, 2017; Chen et al., 2019).

Eventually this work may lead to a prospective study where candidates are sequentially synthesised based on decisions made by the Bayesian optimisation method. Traditionally medicinal chemists synthesise in batches in the design-make-test-analyse cycle, as it is arguably more practical and efficient. Therefore, further work to investigate the effects of optimisation involving batches versus optimisations where a single molecule is observed at each iteration would be interesting. Optimisations using batches of a larger size should take longer to reach the maximum, as the Bayesian

response surface has relatively fewer opportunities to use previously seen information, but the extent of this effect is unclear.

One final aspect that could be explored is the choice of objective function. In this chapter, I have performed validations where the objective function is the experimentally determined potency or binding affinity or EC_{50}; however, alternatives such as absolute binding free energies computed from free-energy perturbation calculations could be investigated.

Chapter 6 Conclusions and Future Directions

A fragment screening campaign, such as those conducted at the high-throughput X-ray crystallographic screening facility (XChem) at the I04-1 beamline at Diamond Light Source, can rapidly generate multiple fragment-protein crystal structures. Currently, there is no agreed method of how to best use this data to propose what compounds to make next. Traditionally, this fragment hit-to-lead elaboration stage has been led by medicinal chemists, a process that can be highly subjective. In this thesis, I have focused on the development of computational methods to design elaboration protocols that are more objective, whilst utilising as much structural information as possible from the fragment hits, in order to prioritise follow-up elaborated fragments that are more potent than the original fragment hit(s).

One of the key challenges to the incorporation of *in silico* methods into hit-to-lead development lies in the multidiscipline nature of the problem and the proof that the *in silico* method is superior to traditional medicinal chemist-led methods. A good outcome of any useful computational tool is successful application to existing drug discovery project and experimental validation; hence, they need to be designed with the experimentalist in mind. Therefore, research efforts should not only focus on the performance of the computational method, but also on its ability to be integrated into the design-test-make-analyse cycle and how easily they can be automated. Better integration will lead to more rapid iterations of the design-test-make-analyse cycle,

which should lead to the faster development of lead structures. Therefore, all computational methods developed in this thesis have been designed with a pragmatic approach that keeps the experimentalist in mind. The methods can be easily adopted by the medicinal chemist, are intuitive to understand, *i.e.* no black box methods, and can be implemented in an iterative manner.

6.1 Summary and Future Work

In Chapter 1, I described the key challenges of drug discovery; how the high attrition rates are often avoidable if multiple objectives, including toxicity and pharmacokinetics, are considered earlier in the drug discovery pipeline. The use of computational methods can objectively aim to ensure only bioavailable, safe, drug-like molecules are considered from the beginning and can make the best use of the large amounts of activity and/or structural data that is available. Fragment-based drug discovery offers high coverage of chemical space early on in the process; however, the resulting fragment-hits are typically weakly binding and need to be elaborated to make more potent drug-like molecules. This elaboration process has traditionally been performed exclusively by medicinal chemists, and tends to be biased by their previous experience. There is a need for more objective methods, although there is no consensus what the best approach is.

In Chapter 2, I proposed a workflow that I applied in a prospective study involving designing follow-up compounds to a fragment hit of Human peroxisomal coenzyme A diphosphatase NUDT7. The workflow was pragmatic and involved reaction enumeration to generate amide follow-ups, protein-ligand docking and computational filtering of docked poses for candidates that share a conserved binding mode with the

original fragment hit. The candidates were further filtered using four hypotheses, which were based on the potential interactions made within the binding pocket and how much structural information could be gained from each. Application of the workflow prioritised 105 amides to synthesise from the original ~35k prospective candidate amides. By programming a robotic liquid handler to perform semi-automated synthesis and subsequent soaking of 78 crude reaction mixtures into crystals of NUDT7, I was able to obtain six crystal structures of my proposed follow-up ligands to NUDT7, five of which are novel and the sixth was the same as the parent fragment hit – a reassuring control result. From this prospective study, I was able to demonstrate successful application of my workflow with a success rate of 6/105 (5.7%), which is comparable to the success rate reported for the screening of fragment libraries. As part of the workflow, I used RMSD of the docked candidates to measure the conservation of binding mode with respect to the common substructure of the parent fragment binding mode. I found, however, that this measure was not optimal for molecules that were pseudosymmetric. Hence this provided the motivation for Chapter 3.

In Chapter 3, I described the development and testing of an open-source combined shape and chemical feature overlap score, namely SuCOS, for the quantification of conservation of binding modes of elaborated fragments. This follows on from one of the key assumptions of fragment-based drug discovery, that the binding mode of the original fragment-hit is structurally conserved upon synthetic elaboration. I compared SuCOS to RMSD and protein-ligand interaction fingerprint (PLIF) similarity in three studies and discussed the strengths and weaknesses of each measure. For example, if either molecule is pseudosymmetric, multiple substructure matches are present, or if there are bioisosteres, then RMSD may fail. RMSD is also size-dependent, so it is difficult to define a universal threshold unless a normalisation is used. PLIFs are

advantageous for capturing the conservation of protein-ligand interactions and are applicable to proteins with different conformations. However, there is no universally accepted definition of protein-ligand interactions and the results can differ greatly depending on which tool is used. When comparing docked poses of an elaborated molecule against its non-elaborated counterpart crystal structure and its true crystal pose, I showed that SuCOS has the highest Pearson correlation out of the three measures investigated. This result is applicable to future prospective work, where (like in Chapter 2) after a fragment screening campaign, candidate follow-ups are docked into their parent fragment crystal structure, SuCOS can be used to select candidates that have at least one docked pose with a conserved binding mode with respect to the parent fragment hit. SuCOS can also be used to determine the optimal direction of fragment growth, by favouring those candidates that overlap well with not only the parent fragment hit but also other hits that may bind in different regions of the binding pocket.

As an aside, another validation of SuCOS that also follows on from work in Chapter 2, could involve rerunning the workflow that proposes and selects compounds for the NUDT7 prospective study. However, instead of using RMSD to filter out poses with a conserved binding mode, SuCOS could be used to see how this affects the selection of compounds and hence the success hit-rate. The majority of compounds selected for synthesis should not change, with the exception of the few candidates that SuCOS deems to have conserved docked binding modes but have a MCS-RMSD greater than 2 Å.

In Chapter 4, I described how the use of SuCOS can be extended to the application of virtual screening of molecules that are not necessarily related by a common substructure. Using the DUD-E benchmark dataset, I compared the performance of ranking with the native AutoDock Vina score to rescoring with SuCOS and found that

on average, SuCOS performed better. I also explored varying the weights of the shape and chemical feature overlap components in SuCOS for each DUD-E target; however, I found that the optimal weight was target dependant, with some targets favouring a larger component of shape whereas others favoured a larger component of chemical features. However, the median optimal weighting over all targets was half shape and half chemical features. For the last study in Chapter 4, I investigated two group fusion methods for SuCOS values – cumulative and maximum ('max') – when there are multiple fragment-protein crystal structures to use as a reference. For validation, I compiled sets of reference fragment-protein crystal structures for four DUD-E targets. Alongside the two group fusion methods, I also investigated the effect of clustering the reference fragments by using Tanimoto SuCOS and selecting representative fragment structures. However, neither the cumulative nor max group fusion method performed consistently better across the four targets. Moreover, it is unclear whether clustering the reference fragments improved the performance. Nevertheless, for three of the four datasets, using the 'best' fragment as a single reference molecule performed better than both of the investigated group fusion methods, when ranked with SuCOS. However, the 'best' fragment or the most informative features are generally unknown for less-studied targets. A potential strategy could involve hypothesising which is the 'best' fragment from a fragment screen and using the other fragment-hits to target sub-pockets in the binding site. Future work could involve applying this potential strategy to a prospective study involving a less-studied target of interest, such as NUDT7.

In Chapter 5, I described the use of Bayesian optimisation for ligand-based and structure-based virtual screening. In the event of a combinatorial explosion from the proposition of elaborated candidate fragments *e.g.* after reaction enumeration with poised reactions, it is often not feasible to experimentally validate every elaboration.

Hence, I proposed using Bayesian optimisation over discrete space using a multi-armed bandit approach, to prioritise which elaborated candidate molecules to make. I expanded on Pyzer-Knapp's study by investigating the effect of different molecular descriptors and different kernels on the performance of Bayesian optimisation. Through the use of two ligand-based validation datasets, I showed that Morgan fingerprints with the Tanimoto kernel performed best amongst the investigated methods, in terms of recovering the largest number of potent molecules. To the best of our knowledge, no one has investigated the use of a 3D descriptor as the Bayesian optimisation search space; hence, I investigated the use of two descriptors: vectorised RDKit pharmacophoric feature maps, and PLIFs. Using PDBbind, I compiled datasets to validate each: the former includes four protein targets, and the latter was validated using one target. However, results from both 3D representation experiments show that there was no clear advantage to using either structure-based representation over the 2D fingerprint methods, Morgan fingerprint and MACCS. Future work could involve further validation with more targets, and/or investigation with other descriptors such as a vectorised shape descriptor or vectorised SuCOS. Also, the effect of Bayesian optimisation in batches instead of sequential/single observations could be of interest, as this is a better reflection of what a medicinal chemist would do in the design-make-test-analyse cycle.

6.2 Concluding Remarks

In this thesis, I have carried out a prospective study of structure-based fragment elaboration with experimental validation; developed an open-source combined shape and chemical feature overlap measure, SuCOS; investigated application of the SuCOS measure to virtual screening; and finally investigated using Bayesian optimisation for

prioritisation of virtual compounds. My discovery of five novel NUDT7 protein-ligand crystal structures has provided more structural information that will inspire the next generation of follow-up compounds to NUDT7. Moreover, my other work has already had an impact on the workflow of fragment elaboration after a fragment-screening campaign at XChem at the I04-1 beamline, as SuCOS has been integrated into the Fragalysis framework at XChem (https://github.com/xchem/fragalysis/) and so will Bayesian optimisation in the future.

Bibliography

Aldrich, C., Bertozzi, C., Georg, G.I., Kiessling, L., Lindsley, C., Liotta, D., Merz, K.M., Schepartz, A., and Wang, S. (2017) The Ecstasy and Agony of Assay Interference Compounds. *ACS Central Science*, 3 (3): 143–147.

Alexander, L.T., Möbitz, H., Drueckes, P., Savitsky, P., Fedorov, O., Elkins, J.M., Deane, C.M., Cowan-Jacob, S.W., and Knapp, S. (2015) Type II Inhibitors Targeting CDK2. *ACS Chem. Biol.*, 10 (9): 2116–2125.

Allen, W.J. and Rizzo, R.C. (2014) Implementation of the Hungarian Algorithm to Account for Ligand Symmetry and Similarity in Structure-Based Design. *Journal of Chemical Information and Modeling*, 54 (2): 518–529.

Alterio, V., Di Fiore, A., D'Ambrosio, K., Supuran, C.T., and De Simone, G. (2012) Multiple Binding Modes of Inhibitors to Carbonic Anhydrases: How to Design Specific Drugs Targeting 15 Different Isoforms? *Chemical Reviews*, 112 (8): 4421–4468.

Anighoro, A. and Bajorath, J. (2016a) Binding mode similarity measures for ranking of docking poses: a case study on the adenosine A2A receptor. *Journal of Computer-Aided Molecular Design*, 30 (6): 447–456.

Anighoro, A. and Bajorath, J. (2016b) Three-Dimensional Similarity in Molecular Docking: Prioritizing Ligand Poses on the Basis of Experimental Binding Modes. *Journal of Chemical Information and Modeling*, 56 (3): 580–587.

Armstrong, M.S., Morris, G.M., Finn, P.W., Sharma, R., Moretti, L., Cooper, R.I., and Richards, W.G. (2010) ElectroShape: Fast molecular similarity calculations incorporating shape, chirality and electrostatics. *Journal of Computer-Aided Molecular Design*, 24 (9): 789–801.

Arrico, L., Abbouda, A., Bianchi, S., and Malagola, R. (2002) Prediction of 'drug-likeness'. *Journal of Medical Case Reports*, 8 (1): 255–271.

Baell, J. and Walters, M.A. (2014) Chemistry: Chemical con artists foil drug discovery. *Nature*, 513 (7519): 481–483.

Baell, J.B. and Holloway, G.A. (2010) New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *Journal of Medicinal Chemistry*, 53 (7): 2719–2740.

Bajusz, D., Rácz, A., and Héberger, K. (2015) Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7 (1).

Bajusz, D., Rácz, A., and Héberger, K. (2019) Comparison of Data Fusion Methods as Consensus Scores for Ensemble Docking. *Molecules*, 24 (15): 2690.

Ballester, P.J., Westwood, I., Laurieri, N., Sim, E., and Richards, W.G. (2010) Prospective virtual screening with ultrafast shape recognition: The identification of

novel inhibitors of arylamine N-acetyltransferases. *Journal of the Royal Society Interface*, 7 (43): 335–342.

Beattie, J.F., Breault, G.A., Ellston, R.P.A., Green, S., Jewsbury, P.J., Midgley, C.J., Naven, R.T., Minshull, C.A., Pauptit, R.A., Tucker, J.A., and Pease, J.E. (2002) Cyclin-dependent kinase 4 inhibitors as a treatment for cancer. Part 1: Identification and optimisation of substituted 4,6-Bis anilino pyrimidines. *Bioorganic and Medicinal Chemistry Letters*, 13 (18): 2955–2960.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000) The Protein Data Bank. *Nucleic acids research*, 28 (1): 235–42. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10592235 (Accessed: 16 June 2017).

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2008) *KNIME: The Konstanz Information Miner*. In Springer Berlin Heidelberg. pp. 319–326.

Bessman, M.J., Frick, D.N., and O'Handley, S.F. (1996) The MutT proteins or "Nudix" hydrolases, a family of versatile, widely distributed, "housecleaning" enzymes. *The Journal of biological chemistry*, 271 (41): 25059–62. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8810257 (Accessed: 28 May 2017).

Bian, Y., Feng, Z., Yang, P., and Xie, X.-Q. (2017) Integrated In Silico Fragment-Based Drug Design: Case Study with Allosteric Modulators on Metabotropic Glutamate Receptor 5. *The AAPS Journal*, 19 (4): 1235–1248.

Böhm, H.-J. and Schneider, G. (2000) *Virtual screening for bioactive molecules*. Wiley-VCH.

Bradley, A.R. (2015) *Development of Tools to Provide Prioritisation and Guidance in the Development of Chemical Probes and Small Molecule Leads*. University of Oxford. Doctor of Philosophy.

Breiman, L. (2001) Random Forests. *Machine Learning*, 45 (1): 5–32.

Brochu, E., Cora, V.M., and de Freitas, N. (2010) *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. Available at: http://arxiv.org/abs/1012.2599 (Accessed: 17 October 2019).

Bursulaya, B.D., Totrov, M., Abagyan, R., and Brooks, C.L. (2003) Comparative study of several algorithms for flexible ligand docking. *Journal of Computer-Aided Molecular Design*, 17 (11): 755–763.

Capecchi, A., Probst, D., and Reymond, J. *One Molecular Fingerprint to Rule them All : Drugs , Biomolecules , and the Metabolome*., pp. 1–42.

Carhart, R.E., Smith, D.H., and Venkataraghavan, R. (1985) Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Modeling*, 25 (2): 64–73.

Cavasotto, C.N., Adler, N.S., and Aucar, M.G. (2018) Quantum chemical approaches in structure-based virtual screening and lead optimization. *Frontiers in Chemistry*, 6 (MAY): 1–7.

Chan, L., Hutchison, G.R., and Morris, G.M. (2019a) Bayesian Optimization for Conformer Generation. *Journal of Cheminformatics*, 11 (32).

Chan, L., Hutchison, G.R., and Morris, G.M. (2019b) BOKEI: Bayesian Optimization

Using Knowledge of Correlated Torsions and Expected Improvement for Conformer Generation. *ChemRxiv*.

Charifson, P.S., Corkery, J.J., Murcko, M.A., and Walters, W.P. (1999) Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. *Journal of Medicinal Chemistry*, 42 (25): 5100–5109.

Che, J., Wang, Z., Sheng, H., Huang, F., Dong, X., Hu, Y., Xie, X., and Hu, Y. (2018) Ligand-based pharmacophore model for the discovery of novel CXCR2 antagonists as anti-cancer metastatic agents. *Royal Society Open Science*, 5 (7): 1–11.

Chen, B., Mueller, C., and Willett, P. (2010) Combination rules for group fusion in similarity-based virtual screening. *Molecular Informatics*, 29 (6–7): 533–541.

Chen, L., Cruz, A., Ramsey, S., Dickson, C.J., Duca, J.S., Hornak, V., Koes, D.R., and Kurtzman, T. (2019) Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening Zhang, Y. (ed.). *PLOS ONE*, 14 (8): e0220113.

Coley, C.W., Jin, W., Rogers, L., Jamison, T.F., Jaakkola, T.S., Green, W.H., Barzilay, R., and Jensen, K.F. (2019a) A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical Science*, 10 (2): 370–377.

Coley, C.W., Thomas, D.A., Lummiss, J.A.M., Jaworski, J.N., Breen, C.P., Schultz, V., Hart, T., Fishman, J.S., Rogers, L., Gao, H., Hicklin, R.W., Plehiers, P.P., Byington, J., Piotti, J.S., Green, W.H., John Hart, A., Jamison, T.F., and Jensen, K.F. (2019b) A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365 (6453).

Collins, P.M., Douangamath, A., Talon, R., Dias, A., Brandao-Neto, J., Krojer, T., and von Delft, F. (2018) Achieving a Good Crystal System for Crystallographic X-Ray Fragment Screening. *Methods in Enzymology*, 610: 251–264.

Congreve, M., Carr, R., Murray, C., and Jhoti, H. (2003) A 'Rule of Three' for fragment-based lead discovery? *Drug Discovery Today*. 8 (19) pp. 876–877.

Connolly, M.L. (1985) Computation of molecular volume. *Journal of the American Chemical Society*, 107 (5): 1118–1124.

Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, 20 (3): 273–297.

Cox, O. (2016) *Design and Utilisation of a Poised Fragment Library against Epigenetic Proteins*. University of Oxford. Doctor of Philosophy.

Cox, O.B., Krojer, T., Collins, P., Monteiro, O., Talon, R., Bradley, A., Fedorov, O., Amin, J., Marsden, B.D., Spencer, J., von Delft, F., and Brennan, P.E. (2016) A poised fragment library enables rapid synthetic expansion yielding the first reported inhibitors of PHIP(2), an atypical bromodomain. *Chem. Sci.*, 7 (3): 2322–2330.

Da, C. and Kireev, D. (2014) Structural Protein–Ligand Interaction Fingerprints (SPLIF) for Structure-Based Virtual Screening: Method and Benchmark Study. *Journal of Chemical Information and Modeling*, 54 (9): 2555–2561.

Daina, A., Michielin, O., and Zoete, V. (2017) SwissADME: A free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports*, 7 (October 2016): 1–13.

Daniels, C.M., Thirawatananond, P., Ong, S.-E., Gabelli, S.B., and Leung, A.K.L.

(2015) Nudix hydrolases degrade protein-conjugated ADP-ribose. *Scientific reports*, 5: 18271.

Dean, P.M., Firth-Clark, S., Harris, W., Kirton, S.B., and Todorov, N.P. (2006) SkelGen: a general tool for structure-based de novo ligand design. *Expert opinion on drug discovery*, 1 (2): 179–189.

Dean, R.A., Cox, J.H., Bellac, C.L., Doucet, A., Starr, A.E., and Overall, C.M. (2008) Macrophage-specific metalloelastase (MMP-12) truncates and inactivates ELR+ CXC chemokines and generates CCL2, -7, -8, and -13 antagonists: potential role of the macrophage in terminating polymorphonuclear leukocyte influx. *Blood*, 112 (8): 3455–3464.

DeGoey, D.A., Chen, H.-J., Cox, P.B., and Wendt, M.D. (2018) Beyond the Rule of 5: Lessons Learned from AbbVie's Drugs and Compound Collection. *Journal of Medicinal Chemistry*, 61 (7): 2636–2651.

Deng, Z., Chuaqui, C., and Singh, J. (2004) Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein−Ligand Binding Interactions. *Journal of Medicinal Chemistry*, 47 (2): 337–344.

Densmore, D.M., Timmons, J., McCarthy, L., Ortiz, L., and Pavan, M. (2017) Automated Robotic Liquid Handling Assembly of Modular DNA Devices. *Journal of Visualized Experiments*, (130).

Desaphy, J., Raimbaud, E., Ducrot, P., and Rognan, D. (2013) Encoding protein-ligand interaction patterns in fingerprints and graphs. *Journal of Chemical Information and Modeling*, 53 (3): 623–637.

DesJarlais, R.L. (2011) "Chapter six – Using Computational Techniques in Fragment-Based Drug Discovery." In *Methods in Enzymology*. pp. 137–155.

DiMasi, J.A., Grabowski, H.G., and Hansen, R.W. (2015) The Cost of Drug Development. *New England Journal of Medicine*, 372 (20): 1972–1972.

Doak, B.C., Over, B., Giordanetto, F., and Kihlberg, J. (2014) Oral druggable space beyond the rule of 5: Insights from drugs and clinical candidates. *Chemistry and Biology*.

Dolinsky, T.J., Nielsen, J.E., McCammon, J.A., and Baker, N.A. (2004) PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic acids research*, 32 (Web Server issue): W665-7.

Dow, M., Fisher, M., James, T., Marchetti, F., and Nelson, A. (2012) Towards the systematic exploration of chemical space. *Org. Biomol. Chem.*, 10 (1): 17–28.

Drew, K.L.M., Baiman, H., Khwaounjoo, P., Yu, B., and Reynisson, J. (2012) Size estimation of chemical space: how big is it? *Journal of Pharmacy and Pharmacology*, 64 (4): 490–495.

Drwal, M.N., Bret, G., Perez, C., Jacquemard, C., Desaphy, J., and Kellenberger, E. (2018) Structural insights on fragment binding mode conservation. *Journal of Medicinal Chemistry*, 61: 5963−5973.

Drwal, M.N. and Griffith, R. (2013) Combination of ligand- and structure-based methods in virtual screening. *Drug Discovery Today: Technologies*, 10 (3): e395–e401.

Drwal, M.N., Jacquemard, C., Perez, C., Desaphy, J., and Kellenberger, E. (2017) Do Fragments and Crystallization Additives Bind Similarly to Drug-like Ligands? *Journal of Chemical Information and Modeling*, 57 (5): 1197–1209.

DUD-E docked poses (2017). Available at: http://bits.csb.pitt.edu/files/docked_dude.tar (Accessed: 2 May 2019).

Duesbury, E., Holliday, J., and Willett, P. (2018) Comparison of Maximum Common Subgraph Isomorphism Algorithms for the Alignment of 2D Chemical Structures. *ChemMedChem*, 13 (6): 588–598.

Durant, J.L., Leland, B.A., Henry, D.R., and Nourse, J.G. (2002) Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42 (6): 1273–1280.

Durrant, J.D., Amaro, R.E., and McCammon, J.A. (2009) AutoGrow: A novel algorithm for protein inhibitor design. *Chemical Biology and Drug Design*, 73 (2): 168–178.

Durrant, J.D., Lindert, S., and McCammon, J.A. (2013) AutoGrow 3.0: An improved algorithm for chemically tractable, semi-automated protein inhibitor design. *Journal of Molecular Graphics and Modelling*, 44: 104–112.

Ebejer, J.P., Finn, P.W., Wong, W.K., Deane, C.M., and Morris, G.M. (2019) Ligity: A Non-Superpositional, Knowledge-Based Approach to Virtual Screening. *Journal of Chemical Information and Modeling*, 59 (6): 2600–2616.

Egbert, M., Whitty, A., Keserű, G.M., and Vajda, S. (2019) Why Some Targets Benefit from beyond Rule of Five Drugs. *Journal of Medicinal Chemistry*.

Ehmki, E.S.R. and Kramer, C. (2017) Matched Molecular Series: Measuring SAR Similarity. *Journal of Chemical Information and Modeling*, 57 (5): 1187–1196.

Emsley, P., Lohkamp, B., Scott, W.G., and Cowtan, K. *Biological Crystallography Features and development of Coot*.

Enamine REAL Database Available at: https://enamine.net/library-synthesis/real-compounds/real-compound-libraries (Accessed: 11 December 2019).

Engkvist, O., Norrby, P.O., Selmi, N., Lam, Y. hong, Peng, Z., Sherer, E.C., Amberg, W., Erhard, T., and Smyth, L.A. (2018) Computational prediction of chemical reactions: current status and outlook. *Drug Discovery Today*, 23 (6): 1203–1218.

Englert, P. and Kovács, P. (2015) Efficient heuristics for maximum common substructure search. *Journal of Chemical Information and Modeling*, 55 (5): 941–955.

Ericksen, S.S., Wu, H., Zhang, H., Michael, L.A., Newton, M.A., Hoffmann, F.M., and Wildman, S.A. (2017) Machine Learning Consensus Scoring Improves Performance Across Targets in Structure-Based Virtual Screening. *Journal of Chemical Information and Modeling*, 57 (7): 1579–1590.

Evans, B.E., Rittle, K.E., Bock, M.G., DiPardo, R.M., Freidinger, R.M., Whitter, W.L., Lundell, G.F., Veber, D.F., Anderson, P.S., Chang, R.S.L., Lotti, V.J., Cerino, D.J., Chen, T.B., Kling, P.J., Kunkel, K.A., Springer, J.P., and Hirshfield, J. (1988) Methods for drug discovery: development of potent, selective, orally effective cholecystokinin antagonists. *Journal of Medicinal Chemistry*, 31 (12): 2235–2246.

Ewing, T.J.A., Makino, S., Skillman, A.G., and Kuntz, I.D. (2001) DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design*, 15 (5): 411–428.

Fausett, L. V. and Laurene (1994) *Fundamentals of neural networks : architectures, algorithms, and applications*. Prentice-Hall. Available at: https://dl.acm.org/citation.cfm?id=197023 (Downloaded: 27 November 2019).

Fechner, U. and Schneider, G. (2006) Flux (1): A Virtual Synthesis Scheme for Fragment-Based de Novo Design. *Journal of Chemical Information and Modeling*, 46 (2): 699–707.

Firth, N.C., Atrash, B., Brown, N., and Blagg, J. (2015) MOARF, an Integrated Workflow for Multiobjective Optimization: Implementation, Synthesis, and Biological Evaluation. *Journal of Chemical Information and Modeling*, 55 (6): 1169–1180.

Friesner, R.A., Banks, J.L., Murphy, R.B., Halgren, T.A., Klicic, J.J., Mainz, D.T., Repasky, M.P., Knoll, E.H., Shelley, M., Perry, J.K., Shaw, D.E., Francis, P., and Shenkin, P.S. (2004) Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *Journal of Medicinal Chemistry*, 47 (7): 1739–1749.

Fu, D.Y. and Meiler, J. (2018) Predictive Power of Different Types of Experimental Restraints in Small Molecule Docking: A Review. *Journal of Chemical Information and Modeling*, 58 (2): 225–233.

Gasmi, L. and McLennan, A.G. (2001) The mouse Nudt7 gene encodes a peroxisomal nudix hydrolase specific for coenzyme A and its derivatives. *Biochemical Journal*, 357 (1): 33.

Ghiandoni, G.M., Bodkin, M.J., Chen, B., Hristozov, D., Wallace, J.E.A., Webster, J., and Gillet, V. (2019) Development and Application of a Data-Driven Reaction Classification Model: Comparison of an ELN and the Medicinal Chemistry Literature. *Journal of Chemical Information and Modeling*, 59: 4167–4187.

Gimeno, A., Ojeda-Montes, M.J., Tomás-Hernández, S., Cereto-Massagué, A., Beltrán-Debón, R., Mulero, M., Pujadas, G., and Garcia-Vallvé, S. (2019) The light and dark sides of virtual screening: What is there to know? *International Journal of Molecular Sciences*, 20 (6).

Gobbi, A. and Poppinger, D. (1998) Genetic optimization of combinatorial libraries. *Biotechnology and Bioengineering*, 61 (1): 47–54.

GPy, Version 1.9.6 (2012). Available at: http://github.com/SheffieldML/GPy (Accessed: 24 May 2019).

GPyOpt, Version 1.2.5 (2016). Available at: http://github.com/SheffieldML/GPyOpt (Accessed: 24 May 2019).

Grant, J.A., Gallardo, M.A., and Pickup, B.T. (1996) A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape. *Journal of Computational Chemistry*, 17 (14): 1653–1666.

Grant, J.A. and Pickup, B.T. (1995) A Gaussian Description of Molecular Shape. *The Journal of Physical Chemistry*, 99 (11): 3503–3510.

Guvench, O., Price, D.J., and Brooks, C.L. (2005) Receptor rigidity and ligand mobility in trypsin-ligand complexes. *Proteins: Structure, Function and Genetics*, 58 (2): 407–417.

Håkansson, K. and Liljas, A. (1994) The structure of a complex between carbonic anhydrase II and a new inhibitor, trifluoromethane sulphonamide. *FEBS Letters*, 350 (2–3): 319–322.

Hall, R.J., Murray, C.W., and Verdonk, M.L. (2017) The Fragment Network: A Chemistry Recommendation Engine Built Using a Graph Database. *Journal of Medicinal Chemistry*, 60: 6440−6450.

Hann, M.M. (2011) Molecular obesity, potency and other addictions in drug discovery. *MedChemComm*, 2 (5): 349–355.

Hartenfeller, M., Eberle, M., Meier, P., Nieto-Oberhuber, C., Altmann, K.-H., Schneider, G., Jacoby, E., and Renner, S. (2011) A collection of robust organic synthesis reactions for in silico molecule design. *Journal of chemical information and modeling*, 51 (12): 3093–8.

Hartenfeller, M. and Schneider, G. (2010) "De Novo Drug Design." In *Chemoinformatics and Computational Chemical Biology*. Humana Press, Totowa, NJ. pp. 299–323.

Hartenfeller, M. and Schneider, G. (2011) Enabling future drug discovery by de novo design. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1 (5): 742–759.

Hartshorn, M.J., Murray, C.W., Cleasby, A., Frederickson, M., Tickle, I.J., and Jhoti, H. (2005) Fragment-based lead discovery using X-ray crystallography. *Journal of Medicinal Chemistry*, 48 (2): 403–413.

Hawkins, P.C.D. and Nicholls, A. (2012) Conformer generation with OMEGA: Learning from the data set and the analysis of failures. *Journal of Chemical Information and Modeling*, 52 (11): 2919–2936.

Hawkins, P.C.D., Warren, G.L., Skillman, A.G., and Nicholls, A. (2008) How to do an evaluation: Pitfalls and traps. *Journal of Computer-Aided Molecular Design*, 22 (3–4): 179–190.

Hopkins, A.L. (2008) Network pharmacology: The next paradigm in drug discovery. *Nature Chemical Biology*. 4 (11) pp. 682–690.

Hopkins, A.L., Groom, C.R., and Alex, A. (2004) Ligand efficiency: A useful metric for lead selection. *Drug Discovery Today*.

Hristozov, D., Bodkin, M., Chen, B., Patel, H., and Gillet, V.J. (2011) "Validation of Reaction Vectors for de Novo Design." In *Library Design, Search Methods, and Applications of Fragment-Based Drug Design*. pp. 29–43.

Huang, B. and Von Lilienfeld, O.A. (2016) Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity. *Journal of Chemical Physics*, 145 (16).

Huang, N., Shoichet, B.K., and Irwin, J.J. (2006) Benchmarking Sets for Molecular Docking. *Journal of Medicinal Chemistry*, 49 (23): 6789–6801.

Huang, S.Y. and Zou, X. (2010) Advances and challenges in Protein-ligand docking. *International Journal of Molecular Sciences*, 11 (8): 3016–3034.

Huey, R., Morris, G.M., Olson, A.J., and Goodsell, D.S. (2007) A semiempirical free energy force field with charge-based desolvation. *Journal of Computational Chemistry*, 28 (6): 1145–1152.

Hughes, J.P., Rees, S., Kalindjian, S.B., and Philpott, K.L. (2011) Principles of early drug discovery. *British journal of pharmacology*, 162 (6): 1239–49.

Hurst, T. (1994) Flexible 3D Searching: The Directed Tweak Technique. *Journal of Chemical Information and Computer Sciences*, 34 (1): 190–196.

Hutter, F., Hoos, H.H., and Leyton-Brown, K. (2011) *Sequential Model-Based Optimization for General Algorithm Configuration*. In Springer, Berlin, Heidelberg. pp.

507–523.

Ichihara, O., Shimada, Y., and Yoshidome, D. (2014) The importance of hydration thermodynamics in fragment-to-lead optimization. *ChemMedChem*, 9 (12): 2708–2717.

Imrie, F., Bradley, A.R., van der Schaar, M., and Deane, C.M. (2018) Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *Journal of Chemical Information and Modeling*, 58 (11): 2319–2330.

Irwin, J.J. and Shoichet, B.K. (2016) Docking Screens for Novel Ligands Conferring New Biology. *Journal of Medicinal Chemistry*, 59 (9): 4103–4120.

Jaccard, P. (1912) The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11 (2): 37–50.

Jain, A.N. (2003) Surflex: Fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of Medicinal Chemistry*, 46 (4): 499–511.

Jain, A.N. and Nicholls, A. (2008) Recommendations for evaluation of computational methods. *Journal of Computer-Aided Molecular Design*, 22 (3–4): 133.

Jasrasaria, D. and Pyzer-Knapp, E.O. (2019) Dynamic control of explore/exploit trade-off in bayesian optimization. *Advances in Intelligent Systems and Computing*, 858 (July): 1–15.

Joerger, A.C., Bauer, M.R., Wilcken, R., Baud, M.G.J., Harbrecht, H., Exner, T.E., Boeckler, F.M., Spencer, J., and Fersht, A.R. (2015) Exploiting Transient Protein States for the Design of Small-Molecule Stabilizers of Mutant p53. *Structure*, 23: 2246–2255.

Johnson, M.A. and Maggiora, G.M. (1992) Concepts and applications of molecular similarity. *Journal of Computational Chemistry*, 13 (4): 539–540.

Jones, E., Travis, O., Pearu, P., and Others (2001) *SciPy: Open source scientific tools for Python*. Available at: http://www.scipy.org/ (Accessed: 1 May 2019).

Jones, G., Willett, P., and Glen, R.C. (1995) Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology*, 245 (1): 43–53.

Jones, G., Willett, P., Glen, R.C., Leach, A.R., and Taylor, R. (1997) Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267 (3): 727–748.

Jorgensen, W.L. (2004) The Many Roles of Computation in Drug Discovery. *Science*. 303 (5665) pp. 1813–1818.

Jubb, H.C. (2014) *Tool(s) for cleaning, munging and analysing PDB files for structural bioinformatics analysis.* Available at: https://github.com/harryjubb/pdbtools (Accessed: 4 October 2017).

Jubb, H.C., Higueruelo, A.P., Ochoa-montaño, B., Pitt, W.R., Ascher, D.B., and Blundell, T.L. (2017) Arpeggio : A Web Server for Calculating and Visualising Interatomic Interactions in Protein Structures. *Journal of Molecular Biology*, 429 (3): 365–371.

Kawabata, T. (2011) Build-up algorithm for atomic correspondence between chemical structures. *Journal of Chemical Information and Modeling*, 51 (8): 1775–1782.

Kearnes, S. and Pande, V. (2016) ROCS-derived features for virtual screening. *Journal*

of Computer-Aided Molecular Design, 30 (8): 609–617.

Kenny, P.W. and Sadowski, J. (2005) *Structure Modification in Chemical Databases*. In John Wiley & Sons, Ltd. pp. 271–285.

Keserű, G.M., Erlanson, D.A., Ferenczy, G.G., Hann, M.M., Murray, C.W., and Pickett, S.D. (2016) Design Principles for Fragment Libraries: Maximizing the Value of Learnings from Pharma Fragment-Based Drug Discovery (FBDD) Programs for Use in Academia. *Journal of Medicinal Chemistry*, 59 (18): 8189–8206.

Kingma, D.P. and Welling, M. (2013) *Auto-Encoding Variational Bayes*. Available at: http://arxiv.org/abs/1312.6114 (Accessed: 17 October 2019).

Kitchen, D.B., Decornez, H., Furr, J.R., and Bajorath, J. (2004) Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nature Reviews Drug Discovery*, 3 (11): 935–949.

Koes, D.R., Baumgartner, M.P., and Camacho, C.J. (2013) Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. *Journal of Chemical Information and Modeling*, 53 (8): 1893–1904.

Kukol, A. (2011) Consensus virtual screening approaches to predict protein ligands. *European Journal of Medicinal Chemistry*, 46 (9): 4661–4664.

Kumar, A. and Zhang, K.Y.J. (2016a) A pose prediction approach based on ligand 3D shape similarity. *Journal of Computer-Aided Molecular Design*, 30 (6): 457–469.

Kumar, A. and Zhang, K.Y.J. (2016b) Application of Shape Similarity in Pose Selection and Virtual Screening in CSARdock2014 Exercise. *Journal of Chemical Information and Modeling*, 56 (6): 965–973.

Kumar, A. and Zhang, K.Y.J. (2018) Advances in the Development of Shape Similarity Methods and Their Application in Drug Discovery. *Frontiers in Chemistry*, 6 (July): 1–21.

Kutchukian, P.S., Vasilyeva, N.Y., Xu, J., Lindvall, M.K., Dillon, M.P., Glick, M., Coley, J.D., and Brooijmans, N. (2012) Inside the Mind of a Medicinal Chemist: The Role of Human Bias in Compound Prioritization during Drug Discovery. *PLoS ONE*, 7 (11).

Leach, A.R. and Gillet, V.J. (2007a) "Selecting Diverse Sets Of Compounds." In *An Introduction To Chemoinformatics*. Dordrecht: Springer Netherlands. pp. 119–139.

Leach, A.R. and Gillet, V.J. (2007b) "Virtual Screening." In *An Introduction To Chemoinformatics*. Dordrecht: Springer Netherlands. pp. 159–181.

Leeson, P.D. (2015) Molecular inflation, attrition and the rule of five. *Advanced Drug Delivery Reviews*, 101: 22–33.

Leeson, P.D. and Springthorpe, B. (2007) The influence of drug-like concepts on decision-making in medicinal chemistry. *Nature reviews. Drug discovery*, 6 (11): 881–90.

Legraverend, M., Tunnah, P., Noble, M., Ducrot, P., Ludwig, O., Grierson, D.S., Leost, M., Meijer, L., and Endicott, J. (2000) Cyclin-dependent kinase inhibition by new C-2 alkynylated purine derivatives and molecular structure of a CDK2-inhibitor complex. *Journal of Medicinal Chemistry*, 43 (7): 1282–1292.

Leung, S., Bodkin, M., von Delft, F., Brennan, P., and Morris, G.M. (2019) *SuCOS is Better than RMSD for Evaluating Fragment Elaboration and Docking Poses*.

Lewell, X.Q., Judd, D.B., Watson, S.P., and Hann, M.M. (1998) RECAP - Retrosynthetic Combinatorial Analysis Procedure: A powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of Chemical Information and Computer Sciences*, 38 (3): 511–522.

Lionta, E., Spyrou, G., Vassilatis, D., and Cournia, Z. (2014) Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. *Current Topics in Medicinal Chemistry*, 14 (16): 1923–1938.

Lipinski, C.A., Lombardo, F., Dominy, B.W., and Feeney, P.J. (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 23 (1): 3–25.

Lipkus, A.H., Yuan, Q., Lucas, K.A., Funk, S.A., Bartelt, W.F., Schenck, R.J., Trippe, A.J., and CAS Registry (2008) Structural diversity of organic chemistry. A scaffold analysis of the CAS Registry. *The Journal of organic chemistry*, 73 (12): 4443–51.

Liu, J., Su, M., Liu, Z., Li, J., Li, Y., and Wang, R. (2017) Enhance the performance of current scoring functions with the aid of 3D protein-ligand interaction fingerprints. *BMC Bioinformatics*, 18 (1): 343.

Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., and Wang, R. (2015) PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics*, 31 (3): 405–412.

Lusher, S.J., McGuire, R., Azevedo, R., Boiten, J.W., Van Schaik, R.C., and De Vlieg, J. (2011) A molecular informatics view on best practice in multi-parameter compound optimization. *Drug Discovery Today*, 16 (13–14): 555–568.

Lyu, J., Wang, S., Balius, T.E., Singh, I., Levit, A., Moroz, Y.S., O'Meara, M.J., Che, T., Algaa, E., Tolmachova, K., Tolmachev, A.A., Shoichet, B.K., Roth, B.L., and Irwin, J.J. (2019) Ultra-large library docking for discovering new chemotypes. *Nature*, 566 (7743): 224–229.

Ma, A.C., Mcnulty, M.S., Poshusta, T.L., Campbell, J.M., Martínez-Gá, G., Argue, D.P., Lee, H.B., Urban, M.D., Bullard, C.E., Blackburn, P.R., Man, T.K., Clark, K.J., and Ekker, S.C. (2016) FusX: A Rapid One-Step Transcription Activator-Like Effector Assembly System for Genome Science. *Human Gene Therapy*, 27 (6): 451–463.

Maggiora, G.M. (2006) On outliers and activity cliffs - Why QSAR often disappoints. *Journal of Chemical Information and Modeling*, 46 (4): 1535.

Mahajan, A. and Teneketzis, D. (2008) "Multi-Armed Bandit Problems." In *Foundations and Applications of Sensor Management*. Boston, MA: Springer US. pp. 121–151.

Mak, K.-K. and Pichika, M.R. (2019) Artificial intelligence in drug development: present status and future prospects. *Drug Discovery Today*, 24 (3): 773–780.

Malaria Box supporting information (2013) *Malaria Box supporting information*. Available at: https://www.mmv.org/sites/default/files/uploads/docs/RandD/Dataset.xlsx (Accessed: 29 May 2019).

Malhotra, S. and Karanicolas, J. (2017) When does chemical elaboration induce a ligand to change its binding mode? *Journal of Medicinal Chemistry*, 60 (1): 128–145.

Marcou, G. and Rognan, D. (2007) Optimizing Fragment and Scaffold Docking by Use of Molecular Interaction Fingerprints. *Journal of Chemical Information and Modeling*, 47 (1): 195–207.

Martin, A.C.R. (2005) Mapping PDB chains to UniProtKB entries. *Bioinformatics*, 21 (23): 4297–4301.

Mashalidis, E.H., Śledź, P., Lang, S., and Abell, C. (2013) A three-stage biophysical screening cascade for fragment-based drug discovery. *Nature Protocols*, 8 (11): 2309–2324.

McLennan, A.G. (2006) The Nudix hydrolase superfamily. *Cellular and Molecular Life Sciences*, 63 (2): 123–143.

Meiler, J. and Baker, D. (2006) ROSETTALIGAND: Protein–Small Molecule Docking with Full Side-Chain Flexibility. *Proteins: Structure, Function, and Bioinformatics*, 65: 538–548.

Miller, M.D., Kearsley, S.K., Underwood, D.J., and Sheridan, R.P. (1994) FLOG: a system to select "quasi-flexible" ligands complementary to a receptor of known three-dimensional structure. *Journal of computer-aided molecular design*, 8 (2): 153–74.

Mitchell, J.B.O. (2014) Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4 (5): 468–481.

Mjolsness, E. and DeCoste, D. (2001) Machine Learning for Science: State of the Art and Future Prospects. *Science*, 293 (5537): 2051–2055.

Moffat, J.G., Vincent, F., Lee, J.A., Eder, J., and Prunotto, M. (2017) Opportunities and challenges in phenotypic drug discovery: An industry perspective. *Nature Reviews Drug Discovery*, 16 (8): 531–543.

Mohs, R.C. and Greig, N.H. (2017) Drug discovery and development: Role of basic biological research. *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, 3 (4): 651–657.

Molecular Operating Environment, Chemical Computing Group, Inc, version 2014.09.

MolPort Building Blocks Database Available at: ftp://molport.com (Accessed: 1 December 2017).

MolVS: Molecule Validation and Standardization, Version 0.0.9 Available at: http://molvs.readthedocs.io/en/latest/ (Accessed: 13 June 2017).

Morgan, H.L. (1965) The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5 (2): 107–113.

Morris, G.M., Huey, R., Lindstrom, W., Sanner, M.F., Belew, R.K., Goodsell, D.S., and Olson, A.J. (2009) AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30 (16): 2785–91.

Mullard, A. (2014) New drugs cost US$2.6 billion to develop. *Nature Reviews Drug Discovery*, 13 (12): 877–877.

Murata, K. and Wolf, M. (2018) Cryo-electron microscopy for structural analysis of dynamic biological macromolecules. *Biochimica et Biophysica Acta - General Subjects*, 1862 (2): 324–334.

Murray, C.W., Callaghan, O., Chessari, G., Cleasby, A., Congreve, M., Frederickson, M., Hartshorn, M.J., McMenamin, R., Patel, S., and Wallis, N. (2007) Application of fragment screening by X-ray crystallography to β-secretase. *Journal of Medicinal Chemistry*, 50 (6): 1116–1123.

Murray, C.W. and Rees, D.C. (2009) The rise of fragment-based drug discovery.

Nature Chemistry, 1 (3): 187–192.

Murray, C.W., Verdonk, M.L., and Rees, D.C. (2012) Experiences in fragment-based drug discovery. *Trends in Pharmacological Sciences*, 33 (5): 224–232.

Murray, J.B., Roughley, S.D., Matassova, N., and Brough, P.A. (2014) Off-rate screening (ORS) by surface plasmon resonance. An efficient method to kinetically sample hit to lead chemical space from unpurified reaction products. *Journal of Medicinal Chemistry*, 57 (7): 2845–2850.

Mysinger, M.M., Carchia, M., Irwin, J.J., and Shoichet, B.K. (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, 55 (14): 6582–6594.

Nadin, A., Hattotuwagama, C., and Churcher, I. (2012) Lead-Oriented Synthesis: A New Opportunity for Synthetic Chemistry. *Angewandte Chemie International Edition*, 51 (5): 1114–1122.

Nicholls, A. (2008) What do we know and when do we know it? *Journal of computer-aided molecular design*, 22 (3–4): 239–55.

Nicholls, A., McGaughey, G.B., Sheridan, R.P., Good, A.C., Warren, G., Mathieu, M., Muchmore, S.W., Brown, S.P., Grant, J.A., Haigh, J.A., Nevins, N., Jain, A.N., and Kelley, B. (2010) Molecular Shape and Medicinal Chemistry: A Perspective. *Journal of Medicinal Chemistry*, 53 (10): 3862–3886.

Nicolaou, C.A. and Brown, N. (2013) Multi-objective optimization methods in drug design. *Drug Discovery Today: Technologies*, 10 (3): e427–e435.

Nilakantan, R., Bauman, N., Dixon, J.S., and Venkataraghavan, R. (1987) Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *Journal of Chemical Information and Computer Sciences*, 27 (2): 82–85.

Nissink, J.W.M. (2009) Simple size-independent measure of ligand efficiency. *Journal of Chemical Information and Modeling*, 49 (6): 1617–1622.

O'Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., and Hutchison, G.R. (2011) Open Babel: An Open chemical toolbox. *Journal of Cheminformatics*, 3 (10): 1–14.

Olsson, M.H.M., Søndergaard, C.R., Rostkowski, M., and Jensen, J.H. (2011) PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. *Journal of Chemical Theory and Computation*, 7 (2): 525–537.

Onodera, K., Satou, K., and Hirota, H. (2007) Evaluations of molecular docking programs for virtual screening. *Journal of Chemical Information and Modeling*, 47 (4): 1609–1618.

OpenEye Shape Toolkit, OpenEye Scientific Software Available at: http://www.eyesopen.com.

Patel, D., Bauman, J.D., and Arnold, E. (2014) Advantages of crystallographic fragment screening: Functional and mechanistic insights from a powerful platform for efficient drug discovery. *Progress in Biophysics and Molecular Biology*, 116 (2–3): 92–100.

Patel, H., Bodkin, M.J., Chen, B., and Gillet, V.J. (2009) Knowledge-Based Approach to *de Novo* Design Using Reaction Vectors. *Journal of Chemical Information and Modeling*, 49 (5): 1163–1184.

PDB Data Distribution by Experimental Method and Molecular Type Available at:

https://www.rcsb.org/stats/summary (Accessed: 21 November 2019).

Pearce, N.M., Krojer, T., Bradley, A.R., Collins, P., Nowak, R.P., Talon, R., Marsden, B.D., Kelm, S., Shi, J., Deane, C.M., and von Delft, F. (2017) A multi-crystal method for extracting obscured crystallographic states from conventionally uninterpretable electron density. *Nature Communications*, 8: 15123.

Peltason, L. and Bajorath, J. (2007) Molecular Similarity Analysis Uncovers Heterogeneous Structure-Activity Relationships and Variable Activity Landscapes. *Chemistry and Biology*, 14 (5): 489–497.

Pickett, S.D., Green, D.V.S., Hunt, D.L., Pardoe, D.A., and Hughes, I. (2011) Automated lead optimization of MMP-12 inhibitors using a genetic algorithm. *ACS Medicinal Chemistry Letters*, 2 (1): 28–33.

Plewczynski, D., Łaźniewski, M., Augustyniak, R., and Ginalski, K. (2011) Can we trust docking results? Evaluation of seven commonly used programs on PDBbind database. *Journal of Computational Chemistry*, 32 (4): 742–755.

Pyzer-Knapp, E.O. (2018) Bayesian optimization for accelerated drug discovery. *IBM Journal of Research and Development*, 62 (6): 2:1-2:7.

Radifar, M., Yuniarti, N., and Istyastono, E.P. (2013) PyPLIF : Python-based Protein-Ligand Interaction Fingerprinting Abstract : Background : Methodology : *Bioinformation*, 9 (6): 325–328.

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., and Koes, D.R. (2017) Protein–Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, 57 (4): 942–957.

Ramírez, D. and Caballero, J. (2018) Is It Reliable to Take the Molecular Docking Top Scoring Position as the Best Solution without Considering Available Structural Data? *Molecules*, 23 (5): 1–17.

Rataj, K., Czarnecki, W., Podlewska, S., Pocha, A., and Bojarski, A.J. (2018) Substructural connectivity fingerprint and extreme entropy machines—a new method of compound representation and analysis. *Molecules*, 23 (6).

Raymond, J.W., Gardiner, E.J., and Willett, P. (2002) Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. *Journal of Chemical Information and Computer Sciences*, 42 (2): 305–316.

RDKit, Version 2017.03.1 (2017). Available at: http://www.rdkit.org (Accessed: 14 September 2017).

RDKit, Version 2018.03.1 (2018). Available at: http://www.rdkit.org (Accessed: 1 May 2018).

RDKit, Version 2019.03.1 (2019). Available at: http://www.rdkit.org (Accessed: 18 April 2019).

Reddy, G.S.K.K., Ali, A., Nalam, M.N.L., Anjum, S.G., Cao, H., Nathans, R.S., Schiffer, C.A., and Rana, T.M. (2007) Design and synthesis of HIV-1 protease inhibitors incorporating oxazolidinones as P2/P2′ ligands in pseudosymmetric dipeptide isosteres. *Journal of Medicinal Chemistry*, 50 (18): 4316–4328.

Renaud, J.P., Chung, C.W., Danielson, U.H., Egner, U., Hennig, M., Hubbard, R.E., and Nar, H. (2016) Biophysics in drug discovery: Impact, challenges and opportunities. *Nature Reviews Drug Discovery*, 15 (10): 679–698.

Reynolds, C.H., Tounge, B.A., and Bembenek, S.D. (2008) Ligand binding efficiency: Trends, physical basis, and implications. *Journal of Medicinal Chemistry*, 51 (8): 2432–2438.

Riniker, S. and Landrum, G.A. (2013) Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of Cheminformatics*, 5 (5).

Riniker, S. and Landrum, G.A. (2015) Better Informed Distance Geometry: Using What We Know to Improve Conformation Generation. *Journal of Chemical Information and Modeling*, 55 (12): 2562–2574.

Ripphausen, P., Nisius, B., Peltason, L., and Bajorath, J. (2010) Quo vadis, virtual screening? A comprehensive survey of prospective applications. *Journal of Medicinal Chemistry*, 53 (24): 8461–8467.

Rishton, G.M. (1997) Reactive compounds and in vitro false positives in HTS. *Drug Discovery Today*, 2 (9): 382–384.

Roberts, R.A., Kavanagh, S.L., Mellor, H.R., Pollard, C.E., Robinson, S., and Platz, S.J. (2014) Reducing attrition in drug development: Smart loading preclinical safety assessment. *Drug Discovery Today*. 19 (3) pp. 341–347.

ROCS, version 3.2.0.3 (2015). Available at: www.eyesopen.com.

Rogers, D. and Hahn, M. (2010) Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50 (5): 742–754.

Rokach, L. and Maimon, O. (2005) "Clustering Methods." In Maimon, O. and Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US. pp. 321–352.

Rossum, G. Van (1995) *Python Software Foundation. Python Language Reference, version 2.7*. Available at: http://www.python.org (Accessed: 1 September 2016).

Roughley, S.D. and Jordan, A.M. (2011) The Medicinal Chemist's Toolbox: An Analysis of Reactions Used in the Pursuit of Drug Candidates. *Journal of Medicinal Chemistry*, 54 (10): 3451–3479.

Satyanarayana, A. and Kaldis, P. (2009) A dual role of Cdk2 in DNA damage response. *Cell Division*, 4: 2–5.

Saubern, S., Guha, R., and Baell, J.B. (2011) KNIME Workflow to Assess PAINS Filters in SMARTS Format. Comparison of RDKit and Indigo Cheminformatics Libraries. *Molecular Informatics*, 30 (10): 847–850.

Schmidt, A., Jelsch, C., Østergaard, P., Rypniewski, W., and Lamzin, V.S. (2003) Trypsin Revisited: Crystallography at (sub) atomic resolution and quantum chemistry revealing details of catalysis. *Journal of Biological Chemistry*, 278 (44): 43357–43362.

Schreyer, A.M. and Blundell, T. (2012) USRCAT: Real-time ultrafast shape recognition with pharmacophoric constraints. *Journal of Cheminformatics*, 4 (11): 1.

Schrödinger, LLC., N. *The PyMOL Molecular Graphics System, Version 2.1.0*.

Scior, T., Bender, A., Tresadern, G., Medina-Franco, J.L., Martínez-Mayorga, K., Langer, T., Cuanalo-Contreras, K., and Agrafiotis, D.K. (2012) Recognizing pitfalls in virtual screening: A critical review. *Journal of Chemical Information and Modeling*, 52 (4): 867–881.

Scior, T., Medina-Franco, J., Do, Q.-T., Martinez-Mayorga, K., Yunes Rojas, J., and Bernard, P. (2009) How to Recognize and Workaround Pitfalls in QSAR Studies: A

Critical Review. *Current Medicinal Chemistry*, 16 (32): 4297–4313.

Scott, D.E., Coyne, A.G., Hudson, S.A., and Abell, C. (2012) Fragment-Based Approaches in Drug Discovery and Chemical Biology. *Biochemistry*, 51 (25): 4990–5003.

Seidel, T., Ibis, G., Bendix, F., and Wolber, G. (2010) Strategies for 3D pharmacophore-based virtual screening. *Drug Discovery Today: Technologies*, 7 (4).

Shah, G.N., Bonapace, G., Hu, P.Y., Strisciuglio, P., and Sly, W.S. (2004) Carbonic anhydrase II deficiency syndrome (osteopetrosis with renal tubular acidosis and brain calcification): Novel mutations in CA2 identified by direct sequencing expand the opportunity for genotype-phenotype correlation. *Human Mutation*, 24 (3): 272–272.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and de Freitas, N. (2016) Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104 (1): 148–175.

Shen, C., Ding, J., Wang, Z., Cao, D., Ding, X., and Hou, T. (2019) From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, (May 2019): 1–23.

Sherawat, M., Kaur, P., Perbandt, M., Betzel, C., Slusarchyk, W.A., Bisacchi, G.S., Chang, C., Jacobson, B.L., Einspahr, H.M., and Singh, T.P. (2007) Structure of the complex of trypsin with a highly potent synthetic inhibitor at 0.97 Å resolution. *Acta Crystallographica Section D: Biological Crystallography*, 63 (4): 500–507.

Shimizu, H., Tosaki, A., Kaneko, K., Hisano, T., Sakurai, T., and Nukina, N. (2008) Crystal Structure of an Active Form of BACE1, an Enzyme Responsible for Amyloid Protein Production. *Molecular and Cellular Biology*, 28 (11): 3663–3671.

Sliwoski, G., Kothiwale, S., Meiler, J., and Lowe, E.W. (2014) Computational Methods in Drug Discovery. *Pharmacological Reviews*, 66 (1): 334–395.

Smyth, M.S. and Martin, J.H.J. (2000) x Ray crystallography. *Journal of Clinical Pathology - Molecular Pathology*. 53 (1) pp. 8–14.

Snoek, J., Larochelle, H., and Adams, R.P. (2012) Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 4: 2951–2959.

Søndergaard, C.R., Olsson, M.H.M., Rostkowski, M., and Jensen, J.H. (2011) Improved Treatment of Ligands and Coupling Effects in Empirical Calculation and Rationalization of pKa Values. *Journal of Chemical Theory and Computation*, 7 (7): 2284–2295.

Sousa, S.F., Fernandes, P.A., and Ramos, M.J. (2006) Protein–Ligand Docking: Current Status and Future Challenges. *Proteins: Structure, Function, and Bioinformatics*, 26: 15–26.

Spangenberg, T., Burrows, J.N., Kowalczyk, P., McDonald, S., Wells, T.N.C., and Willis, P. (2013) The Open Access Malaria Box: A Drug Discovery Catalyst for Neglected Diseases. *PLoS ONE*, 8 (6).

Spencer, J., Patel, H., Amin, J., Callear, S.K., Coles, S.J., Deadman, J.J., Furman, C., Mansouri, R., Chavatte, P., and Millet, R. (2012) Microwave-mediated synthesis and manipulation of a 2-substituted-5-aminooxazole-4-carbonitrile library. *Tetrahedron Letters*, 53 (13): 1656–1659.

Spurlino, J.C. (2011) "Fragment screening purely with protein crystallography." In

Methods in Enzymology. pp. 321–356.

Sridhar, A., Ross, G.A., and Biggin, P.C. (2017) Waterdock 2.0: Water placement prediction for Holo-structures with a pymol plugin. *PLoS ONE*, 12 (2): e0172743.

Stubbs, M.T. (1998) Structural and functional analyses of benzamidine-based inhibitors in complex with trypsin: Implications for the inhibition of factor Xa, tPA, and urokinase. *Journal of Medicinal Chemistry*, 41 (27): 5445–5456.

Tanimoto, T.T. (1957) "An Elementary Mathematical theory of Classification and Prediction." In *Internal IBM Technical Report*. New York, 1957.

Temml, V., Voss, C. V., Dirsch, V.M., and Schuster, D. (2014) Discovery of New Liver X Receptor Agonists by Pharmacophore Modeling and Shape-Based Virtual Screening. *Journal of Chemical Information and Modeling*, 54 (2): 367–371.

Tian, S., Wang, J., Li, Y., Li, D., Xu, L., and Hou, T. (2015) The application of in silico drug-likeness predictions in pharmaceutical research. *Advanced Drug Delivery Reviews*.

Trott, O. and Olson, A.J. (2010) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading Oleg. *Journal of Computational Chemistry*, 31 (2): 455–461.

Tversky, A. (1977) Features of Similarity. *Psychological Review*, 84 (4): 327–352.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., and Zhao, S. (2019) Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18 (6): 463–477.

Velupillai, S., Sáez, L.D., Krojer, T., Bennett, J., Ruda, G.F., Szommer, T., Straub, V., Alonso, G.N., Siejka, P., Bradley, A., Talon, R., Fairhead, M., Elkins, J., Delft, F. von, Fedorov, O., Brennan, P., and Huber, K. (2018) Human Peroxisomal Coenzyme A Diphosphatase NUDT7(NUDT7); A Target Enabling Package (TEP). *Zenodo*.

Verdonk, M.L., Ludlow, R.F., Giangreco, I., and Rathi, P.C. (2016) Protein-ligand informatics force field (PLIff): Toward a fully knowledge driven "force field" for biomolecular interactions. *Journal of Medicinal Chemistry*, 59 (14): 6891–6902.

Virtanen, P., Gommers, R., Oliphant, T.E., et al. (2019) SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, p. arXiv:1907.10121.

Walters, W.P. and Murcko, M.A. (2002) Prediction of "drug-likeness." *Advanced Drug Delivery Reviews*, 54: 255–271.

Walters, W.P., Stahl, M.T., and Murcko, M.A. (1998) Virtual screening—an overview. *Drug Discovery Today*, 3 (4): 160–178.

Wang, R., Fang, X., Lu, Y., and Wang, S. (2004) The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry*, 47 (12): 2977–2980.

Wang, X., Han, W., Yan, X., Zhang, J., Yang, M., and Jiang, P. (2019) Pharmacophore features for machine learning in pharmaceutical virtual screening. *Molecular Diversity*, pp. 1–6.

Warren, G.L., Andrews, C.W., Capelli, A.M., Clarke, B., LaLonde, J., Lambert, M.H., Lindvall, M., Nevins, N., Semus, S.F., Senger, S., Tedesco, G., Wall, I.D., Woolven, J.M., Peishoff, C.E., and Head, M.S. (2006) A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry*, 49 (20): 5912–5931.

Wermuth, C.G., Ganellin, C.R., Lindberg, P., and Mitscher, L.A. (1998) Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998). *Pure and Applied Chemistry*, 70 (5): 1129–1143.

Willett, P. (2013) Fusing similarity rankings in ligand-based virtual screening. *Computational and Structural Biotechnology Journal*, 5 (6): e201302002.

Winter, R., Montanari, F., Noé, F., and Clevert, D.A. (2019) Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical Science*, 10 (6): 1692–1701.

Wójcikowski, M., Kukiełka, M., Stepniewska-Dziubinska, M.M., and Siedlecki, P. (2019) Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. *Bioinformatics*, 35 (8): 1334–1341.

Wójcikowski, M., Zielenkiewicz, P., and Siedlecki, P. (2015) Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of Cheminformatics*, 7 (1): 26.

Wojdyr, M., Keegan, R., Winter, G., and Ashton, A. (2013) DIMPLE - a pipeline for the rapid generation of difference maps from protein crystals with putatively bound ligands. *Acta Crystallographica Section A Foundations of Crystallography*, 69 (a1): s299–s299.

Yang, J.-M. and Chen, C.-C. (2004) GEMDOCK: A generic evolutionary method for molecular docking. *Proteins: Structure, Function, and Bioinformatics*, 55 (2): 288–304.

Young, R.J. and Leeson, P.D. (2018) Mapping the Efficiency and Physicochemical Trajectories of Successful Optimizations. *Journal of Medicinal Chemistry*, 61 (15): 6421–6467.

Yuriev, E., Holien, J., and Ramsland, P.A. (2015) Improvements, trends, and new ideas in molecular docking: 2012-2013 in review. *Journal of Molecular Recognition*, 28 (10): 581–604.

Zaliani, A., Boda, K., Seidel, T., Herwig, A., Schwab, C.H., Gasteiger, J., Claußen, H., Lemmen, C., Degen, J., Pärn, J., and Rarey, M. (2009) Second-generation de novo design: A view from a medicinal chemist perspective. *Journal of Computer-Aided Molecular Design*, 23 (8): 593–602.

Zhang, S. (2011) "Computer-Aided Drug Discovery and Development." In *Drug Design and Discovery. Methods in Molecular Biology (Methods and Protocols)*. Humana Press. pp. 23–38.

Zhang, Y. and Skolnick, J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research*, 33 (7): 2302–9.

Zhu, Z., Sun, Z.Y., Ye, Y., et al. (2010) Discovery of cyclic acylguanidines as highly potent and selective β-site amyloid cleaving enzyme (BACE) inhibitors: Part I - Inhibitor design and validation. *Journal of Medicinal Chemistry*, 53 (3): 951–965.

Appendix A : Chapter 2

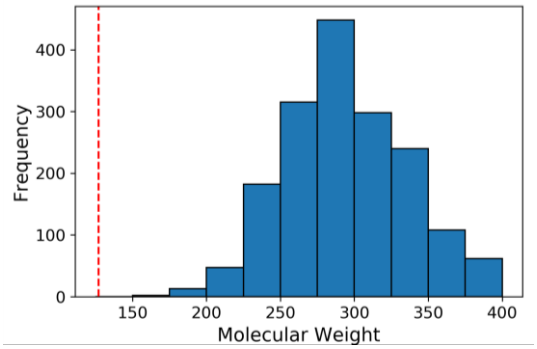


Figure A.1. The distribution of molecular weights of the prospective amide candidates that were docked into NUDT7.The red dotted vertical line (at molecular weight=127) represents the molecular weight of 4-chloroaniline..

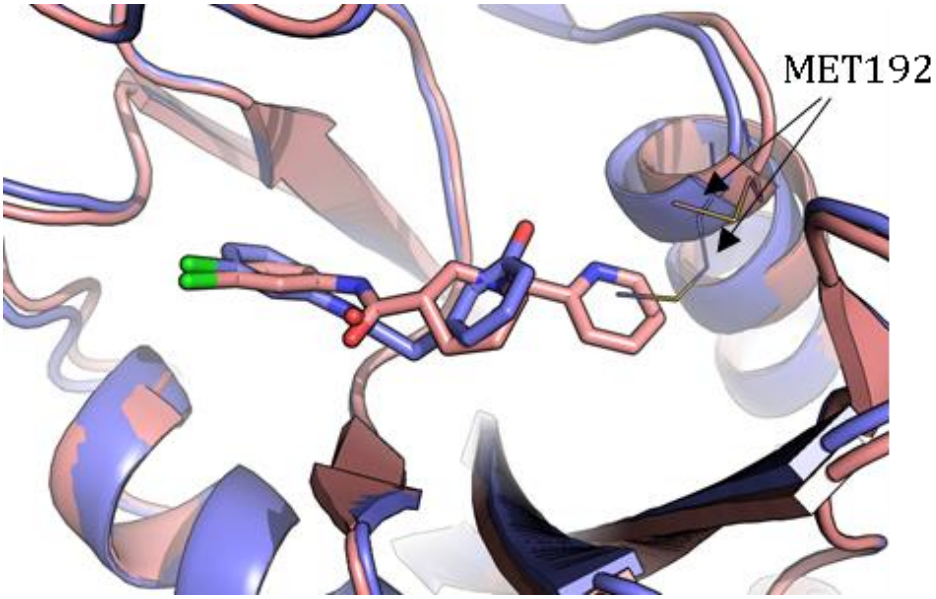


Figure A.2 The predicted docked pose for x0090 had a RMSD > 2 Å with respect to the crystal pose. The experimental X-ray crystal structures of NUDT7-x1237 (blue) and NUDT7-x0090 (pink) are shown, with residue MET192 shown in lines. MET192 of NUDT7-x1237 clashes with ligand x0090, hence this conformational change in protein, which was not modelled in docking, could explain the unsuccessful pose prediction of x0090. This figure was produced using PyMOL (Schrödinger, LLC.).

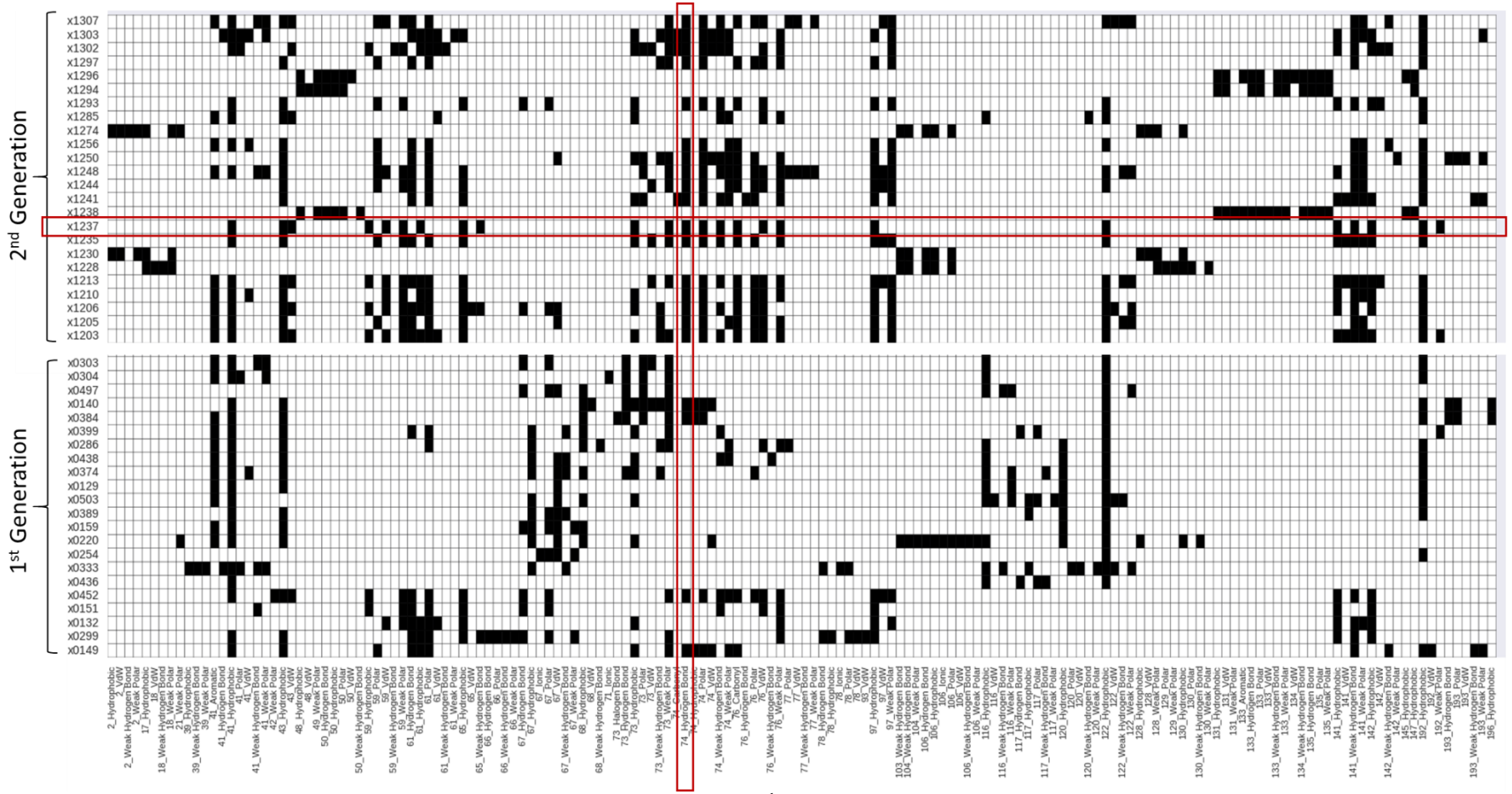


Figure A.3. Protein Ligand Interaction Fingerprint (PLIF) heatmap for the 1st and 2nd generation hits for NUDT7. The PLIFs were computed using Arpeggio (Jubb et al., 2017). Fragment hits are listed on the *y*-axis and interactions with the amino acids of the protein are listed on the *x*-axis. The notation for the PLIFs along the *x*-axis is the residue number and the interaction type*: e.g,* 2_Hydrophobic refers to a hydrophobic interaction with residue 2. The presence of an interaction is shown by a black square. The horizontal red box highlights the hit x1237 for which I proposed follow-up compounds in Chapter 2. The vertical red box highlights one of the interactions frequently seen in the 2nd generation hits and is one of the criterion for filtering out docking poses in Section 2.3.1.

A.1 Code for synthesis protocol for Opentrons

Here are the classes I defined in all my protocols:

```
class Vector(object):
    def tolist(self):
        return list(self.input_list)

    def astype(self, input_type):
        if input_type == int:
            return Vector([int(float(x)) for x in self.input_list])
        return Vector([input_type(x) for x in self.input_list])

    def __init__(self, input_list):
        self.input_list = input_list

class DataFrame(object):
    def __len__(self):
        return self.length

    def __getitem__(self, value):
        return Vector(self.dict_input[value])

    def __init__(self, dict_input, length):
        self.dict_input = dict_input
        self.length = length

def read_csv(input_file):
    lines = open(input_file).readlines()
    header = lines[0].rstrip().split(",")
    out_d = {}
    for head in header:
        out_d[head] = []
    for line in lines[1:]:
        spl_line = line.rstrip().split(",")
        for i, head in enumerate(header):
            out_d[head].append(spl_line[i])
    df = DataFrame(out_d, len(lines[1:]))
    return df
```

The following code sets the speed of movement by Opentrons and sets up the deck by specifying the location of the tip rack, solvent rack and destination rack:

```
from opentrons import robot, containers, instruments

robot.head_speed(x=18000, y=18000, z=5000, a=700, b=700)

# Deck setup
tiprack = containers.load("tiprack-1000ul-H", "B3")
source = containers.load('trough-12row', 'C2') # DMA rack
```

```
destination = containers.load("FluidX_96_small", "A1") # acids tray
to make stock with
trash = containers.load("point", 'C3')

# Pipettes SetUp
p1000 = instruments.Pipette(
    name='eppendorf1000',
    axis='b',
    trash_container=trash,
    tip_racks=[tiprack],
    max_volume=1000,
    min_volume=30,
    channels=1,
)
```

The following code makes up stock solutions for the solid reagents:

```
pos_source = [source.wells('A2').bottom()] # where to aspirate
solvent

acids_df =
read_csv(r"C:\Users\opentrons\Desktop\susans_protocol\susans_protoco
l\csvs\AcOH_platemap.csv")

def run_custom_protocol(df):
    p1000.pick_up_tip()
    # transfer one-by-one
    for i, x in enumerate(df["Rack position"].tolist()):
        if i > 38:
            rack_pos = x
            vol = round(float(df["DMA"].tolist()[i]))
            if vol != 0:
                p1000.transfer([vol], pos_source,
destination.wells(rack_pos).top(5), new_tip='never')
    p1000.drop_tip()
    return

run_custom_protocol(acids_df)
```

The following code mixes all reagents to start the coupling reaction:

```
# get quantities and locations from csv file
others_df =
read_csv(r"C:\Users\opentrons\Desktop\susans_protocol\susans_protoco
l\csvs\Others_Coupling_susan.csv")
acids_df =
read_csv(r"C:\Users\opentrons\Desktop\susans_protocol\susans_protoco
l\csvs\AcOH_platemap.csv")

def run_custom_protocol(others_df, acids_df):
    amine = "3-chloroaniline"
    base = "NME"
    coupling_agent = "T3P"

    # transfer 0.8M amine
    row_nb = [i for i, x in enumerate(others_df["CPD ID"].tolist())
if x == amine][0]
    vol = others_df["Volume per reaction (uL)"].tolist()[row_nb]
    rack_pos = others_df["Rack position"].tolist()[row_nb]
    p300_multi.distribute([vol], trough.wells(rack_pos), [x.top(-12)
for x in destination.rows()], blow_out= True)

    # transfer neat base
    row_nb = [i for i, x in enumerate(others_df["CPD ID"].tolist())
if x == base][0]
    vol = others_df["Volume per reaction (uL)"].tolist()[row_nb]
    rack_pos = others_df["Rack position"].tolist()[row_nb]
    p300_multi.distribute([vol], trough.wells(rack_pos), [x.top(-12)
for x in destination.rows()])

    # transfer 50% solution coupling agent T3P
    row_nb = [i for i, x in enumerate(others_df["CPD ID"].tolist())
if x == coupling_agent][0]
    vol = others_df["Volume per reaction (uL)"].tolist()[row_nb]
    rack_pos = others_df["Rack position"].tolist()[row_nb]
    p300_multi.distribute([vol], trough.wells(rack_pos), [x.top(-12)
for x in destination.rows()])

    p300_multi.transfer(['78.75'], [x.bottom(2) for x in
source_row.rows()], destination.rows(), new_tip='always')

    return

run_custom_protocol(others_df, acids_df)
```

The following code takes aliquots of the reaction mixture for quality control:

```
#CSV file data
others_df =
read_csv(r"C:\Users\opentrons\Desktop\susans_protocol\csvs\Others_Co
upling_susan.csv")

def run_custom_protocol(others_df):
    cpd_id = "CPD ID"
    solvent = "MeOH"
    location_header = "Rack position"
    volume_header = "Volume per reaction (uL)"

    # Transfer 15uL from rxn well to 96 qc plate
    source_QC = [well.bottom(2) for well in reaction_rack.rows()]
    destination_QC = [well.top() for well in plate_QC.rows()]

    # # Dispense 50uL from trough containing MeOH or MeCN to 96
wellplate
    for i, x in enumerate(others_df[cpd_id].tolist()):
        if x == solvent:
            source_multi = others_df[location_header].tolist()[i]
            volume_multi =
round(float(others_df[volume_header].tolist()[i]))
            p300_multi.distribute([volume_multi],
source.wells(source_multi), destination_QC)

    return

run_custom_protocol(others_df)
```

The following code to transfers aliquots of the crude reactions from the 96 well reaction plate into a 384 well plate which is required for soaking of the protein crystals with the LabCyte Echo liquid handler:

```
# CSV file data
others_df =
read_csv(r"C:\Users\opentrons\Desktop\susans_protocol\csvs\Others_Co
upling_susan.csv")

def run_custom_protocol(others_df):
    source_transfer = [well.bottom(1) for well in
reaction_rack.rows(0, to=9)]
    destination_384 = []
    for row in transfer_rack.rows(0, to=9):
        destination_384.append(row.wells('B', length=8,
step=2).bottom(1))
    p300_multi.transfer(20, source_transfer, destination_384,
blow_out=True, new_tip='always')

run_custom_protocol(others_df)
```

A.2 Tables listing the reagent SMILES, hypotheses and well locations for the 105 amide-forming reactions

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
CCOC(=O)Cl	AcCl	CCOC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide0_PEBreagent	199.0400062	108.521	A1	A1
COC(=O)Cl	AcCl	COC(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide3_PEBreagent	185.0243562	94.494	B1	N/A
CC(C)C(=O)Cl	AcCl	CC(C)C(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide6_PEBreagent	197.0607417	106.549	C1	C1
CC(C)CC(=O)Cl	AcCl	CC(C)CC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide9_PEBreagent	211.0763917	120.576	D1	D1
CCC(C)C(=O)Cl	AcCl	CC[C@@H](C)C(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide12_PEBreagent	211.0763917	120.576	E1	E1
O=C(Cl)C1CCCCC1	AcCl	O=C(Nc1cccc(Cl)c1)C1CCCCC1	Greasy pocket	Amide18_PEBreagent	237.0920418	146.614	F1	F1
O=C(Cl)c1ccc(F)cc1F	AcCl	O=C(Nc1cccc(Cl)c1)c1ccc(F)cc1F	73 polar / H-bond	Amide21_PEBreagent	267.026248	176.547	G1	G1
O=C(Cl)Oc1ccccc1	AcCl	O=C(Nc1cccc(Cl)c1)Oc1ccccc1	Greasy pocket	Amide23_PEBreagent	247.0400062	156.565	H1	N/A
O=C(Cl)Cc1ccc(F)cc1	AcCl	O=C(Cc1ccc(F)cc1)Nc1cccc(Cl)c1	Greasy pocket	Amide24_PEBreagent	263.0513199	172.583	A2	A2
O=C(Cl)COc1ccccc1	AcCl	O=C(COc1ccccc1)Nc1cccc(Cl)c1	Greasy pocket	Amide25_PEBreagent	261.0556563	170.592	B2	B2
Cc1ccccc1C(=O)Cl	AcCl	Cc1ccccc1C(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide27_PEBreagent	245.0607417	154.593	C2	C2
COc1ccc(CC(=O)Cl)cc1	AcCl	COc1ccc(CC(=O)Nc2cccc(Cl)c2)cc1	Greasy pocket	Amide30_PEBreagent	275.0713064	184.619	D2	D2

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
COc1cccc(CC(=O)Cl)c1	AcCl	COc1cccc(CC(=O)Nc2cccc(Cl)c2)c1	Greasy pocket	Amide32_PEBreagent	275.0713064	184.619	E2	E2
CC(=O)Cl	AcCl	CC(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide34_PEBreagent	169.0294416	78.495	F2	F2
COCC(=O)Cl	AcCl	COCC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide37_PEBreagent	199.0400062	108.521	G2	G2
O=C(Cl)Cc1ccccc1	AcCl	O=C(Cc1ccccc1)Nc1cccc(Cl)c1	Greasy pocket	Amide39_PEBreagent	245.0607417	154.593	H2	H2
Cc1ccc(CC(=O)Cl)cc1	AcCl	Cc1ccc(CC(=O)Nc2cccc(Cl)c2)cc1	Greasy pocket	Amide46_PEBreagent	259.0763917	168.62	A3	A3
O=C(Cl)Cc1ccccc1F	AcCl	O=C(Cc1ccccc1F)Nc1cccc(Cl)c1	Greasy pocket	Amide47_PEBreagent	263.0513199	172.583	B3	B3
O=C(Cl)Cc1cccc(F)c1	AcCl	O=C(Cc1cccc(F)c1)Nc1cccc(Cl)c1	Greasy pocket	Amide48_PEBreagent	263.0513199	172.583	C3	C3
O=C(Cl)Cc1ccc2c(c1)OCO2	AcCl	O=C(Cc1ccc2c(c1)OCO2)Nc1cccc(Cl)c1	Greasy pocket	Amide49_PEBreagent	289.0505709	198.602	D3	D3
CCC[C@@H](C)C(=O)Cl	AcCl	CCCC(C)C(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide243_Molport AcCl	225.092042	134.0498	E3	E3
COc1cccc(C(=O)Cl)c1	AcCl	COc1cccc(C(=O)Nc2cccc(Cl)c2)c1	73 polar / H-bond	Amide232_Molport AcCl	261.055656	170.0135	F3	F3
O=C(Cl)c1ccnc(Cl)c1	AcCl	O=C(Nc1cccc(Cl)c1)c1ccnc(Cl)c1	Greasy pocket	Amide157_Molport AcCl	266.001368	174.9592	G3	G3
COCCC(=O)Cl	AcCl	COCCC(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide238_Molport AcCl	213.055656	122.0135	H3	H3
CC(C)CCC(=O)Cl	AcCl	CC(C)CCC(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide510_Molport AcCl	225.092042	134.0498	A4	A4

Table A-1. Summary of the 25 acyl chlorides used in the acylation reactions.

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
CN(C)CC(=O)O.Cl	AcOH	CN(C)CC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide64_PEBreagent	212.0716407	139.579	A1	N/A
O=C(O)C1CC1	AcOH	O=C(Nc1cccc(Cl)c1)C1CC1	Best Pareto and diverse	Amide68_PEBreagent	195.0450916	86.09	B1	B01
O=C(O)C1CCCC1	AcOH	O=C(Nc1cccc(Cl)c1)C1CCCC1	Worst Pareto and diverse	Amide74_PEBreagent	223.0763917	114.144	C1	C01
O=C(O)CC1CCCC1	AcOH	O=C(CC1CCCC1)Nc1cccc(Cl)c1	Greasy pocket	Amide80_PEBreagent	237.0920418	128.171	D1	D01
O=C(O)Cc1ccc(F)cc1F	AcOH	O=C(Cc1ccc(F)cc1F)Nc1cccc(Cl)c1	Greasy pocket	Amide89_PEBreagent	281.0418981	172.131	E1	E01
Cc1cccc(CC(=O)O)c1	AcOH	Cc1cccc(CC(=O)Nc2cccc(Cl)c2)c1	Greasy pocket	Amide90_PEBreagent	259.0763917	150.177	F1	F01
O=C(O)c1ccc(O)cc1	AcOH	O=C(Nc1cccc(Cl)c1)c1ccc(O)cc1	Greasy pocket	Amide96_PEBreagent	247.0400062	138.122	G1	N/A
O=C(O)c1cncc(Br)c1	AcOH	O=C(Nc1cccc(Cl)c1)c1cncc(Br)c1	Greasy pocket	Amide102_PEBreagent	309.9508527	202.007	H1	H01
O=C(O)c1ccccc1O	AcOH	O=C(Nc1cccc(Cl)c1)c1ccccc1O	Greasy pocket	Amide105_PEBreagent	247.0400062	138.122	A2	A02
CC(C)(C)OC(=O)NC(CO)C(=O)O	AcOH	CC(C)(C)OC(=O)N[C@H](CO)C(=O)Nc1ccccc(Cl)c1	Greasy pocket	Amide53_PEBreagent	314.1033348	205.21	B2	B02
O=C(O)Cc1ccc(Cl)nc1	AcOH	O=C(Cc1ccc(Cl)nc1)Nc1cccc(Cl)c1	Greasy pocket	Amide106_PEBreagent	280.0170183	171.58	C2	C02
CC(=O)NCC(=O)O	AcOH	CC(=O)NCC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide54_PEBreagent	226.0509053	117.104	D2	N/A
O=C(O)C1CCOCC1	AcOH	O=C(Nc1cccc(Cl)c1)C1CCOCC1	Worst Pareto and diverse	Amide107_PEBreagent	239.0713064	130.143	E2	E02
O=C(O)c1ccc(F)cc1	AcOH	O=C(Nc1cccc(Cl)c1)c1ccc(F)cc1	Greasy pocket	Amide110_PEBreagent	249.0356698	140.113	F2	F02
N#Cc1cccc(C(=O)O)c1	AcOH	N#Cc1cccc(C(=O)Nc2cccc(Cl)c2)c1	Greasy pocket	Amide111_PEBreagent	256.0403406	147.133	G2	N/A
O=C(O)c1ccccc1F	AcOH	O=C(Nc1cccc(Cl)c1)c1ccccc1F	73 polar / H-bond	Amide112_PEBreagent	249.0356698	140.113	H2	H02
CC(C)(C)OC(=O)NCCC(=O)O	AcOH	CC(C)(C)OC(=O)NCCC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide55_PEBreagent	298.1084201	189.211	A3	A03
O=C(O)c1cccc(-c2nnn[nH]2)c1	AcOH	O=C(Nc1cccc(Cl)c1)c1cccc(-c2nnn[nH]2)c1	73 polar / H-bond	Amide125_PEBreagent	299.0573876	190.162	B3	B03

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
Cn1cncc1C(=O)O	AcOH	Cn1cncc1C(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide127_PEBReagent	235.0512396	126.115	C3	C03
O=C(O)C1CCCOC1	AcOH	O=C(Nc1cccc(Cl)c1)[C@H]1CCCOC1	Worst Pareto and diverse	Amide129_PEBReagent	239.0713064	130.143	D3	D03
O=C(O)c1c(F)cccc1F	AcOH	O=C(Nc1cccc(Cl)c1)c1c(F)cccc1F	73 polar / H-bond	Amide137_PEBReagent	267.026248	158.104	E3	E03
CC(C)(C)OC(=O)N1CC(O)CC1C(=O)O	AcOH	CC(C)(C)OC(=O)N1C[C@H](O)C[C@H]1C(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide140_PEBReagent	340.1189848	231.248	F3	F03
Cc1ncccc1C(=O)O	AcOH	Cc1ncccc1C(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide141_PEBReagent	246.0559907	137.138	G3	G03
Cl.O=C(O)Cc1ccncc1	AcOH	O=C(Cc1ccncc1)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide144_PEBReagent	246.0559907	173.596	H3	H03
CC(=O)Nc1cccc(C(=O)O)c1	AcOH	CC(=O)Nc1cccc(C(=O)Nc2cccc(Cl)c2)c1	Greasy pocket	Amide60_PEBreagent	288.0665553	179.175	A4	A04
CC(C)(C)C(=O)O	AcOH	CC(C)(C)C(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide149_PEBReagent	211.0763917	102.133	B4	B04
N#CCC(=O)O	AcOH	N#CCC(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide153_PEBReagent	194.0246905	85.062	C4	N/A
N#Cc1cccc(CC(=O)O)c1	AcOH	N#Cc1cccc(CC(=O)Nc2cccc(Cl)c2)c1	Greasy pocket	Amide156_PEBReagent	270.0559907	161.16	D4	D04
O=C(O)Cc1cnccn1	AcOH	O=C(Cc1cnccn1)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide158_PEBReagent	247.0512396	138.126	E4	N/A
Cc1ccc(CC(=O)O)cn1	AcOH	Cc1ccc(CC(=O)Nc2cccc(Cl)c2)cn1	Greasy pocket	Amide159_PEBReagent	260.0716407	151.165	F4	F04
O=C(O)Cc1cccnc1	AcOH	O=C(Cc1cccnc1)Nc1cccc(Cl)c1	Greasy pocket	Amide160_PEBReagent	246.0559907	137.138	G4	N/A
N#Cc1cc(CC(=O)O)ccn1	AcOH	N#Cc1cc(CC(=O)Nc2cccc(Cl)c2)ccn1	Worst Pareto and diverse	Amide161_PEBReagent	271.0512396	162.148	H4	H04
CC(C(=O)O)c1cccc(C#N)c1	AcOH	C[C@@H](C(=O)Nc1cccc(Cl)c1)c1cccc(C#N)c1	Greasy pocket	Amide168_PEBReagent	284.0716407	175.187	A5	N/A
CN1CCN(CC(=O)O)CC1	AcOH	C[N@]1CC[N@@](CC(=O)Nc2cccc(Cl)c2)CC1	Best Pareto and diverse	Amide170_PEBReagent	267.1138399	158.201	B5	N/A
CN1CCN(CCC(=O)O)CC1.Cl.Cl	AcOH	C[N@]1CC[N@](CCC(=O)Nc2cccc(Cl)c2)CC1	Worst Pareto and diverse	Amide172_PEBReagent	281.1294899	245.144	C5	N/A
O=C(O)Cc1ccc(O)cc1	AcOH	O=C(Cc1ccc(O)cc1)Nc1cccc(Cl)c1	Greasy pocket	Amide173_PEBReagent	261.0556563	206.12	D5	D05

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
O=C(O)Cc1cccc(O)c1	AcOH	O=C(Cc1cccc(O)c1)Nc1cccc(Cl)c1	Greasy pocket	Amide174_PEBr eagent	261.055 6563	152.149	E5	E05
Cc1ccccc1CC(=O)O	AcOH	Cc1ccccc1CC(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide175_PEBr eagent	259.076 3917	150.177	F5	F05
O=C(O)Cc1cccc2[nH]ncc12	AcOH	O=C(Cc1cccc2[nH]ncc12)Nc1cccc(Cl)c1	Greasy pocket	Amide177_PEBr eagent	285.066 8897	176.175	G5	G05
O=C(O)Cc1ccc2[nH]ncc2c1	AcOH	O=C(Cc1ccc2[nH]ncc2c1)Nc1cccc(Cl)c1	Greasy pocket	Amide178_PEBr eagent	285.066 8897	176.175	H5	H05
O=C(O)c1cn[nH]c1	AcOH	O=C(Nc1cccc(Cl)c1)c1cn[nH]c1	Worst Pareto and diverse	Amide179_PEBr eagent	221.035 5896	112.088	A6	A06
Cl.O=C(O)CN1CCCCC1	AcOH	O=C(CN1CCCCC1)Nc1cccc(Cl)c1	Greasy pocket	Amide183_PEBr eagent	252.102 9408	179.644	B6	B06
CC(C)(C)OC(=O)NCC(=O)O	AcOH	CC(C)(C)OC(=O)NCC(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide61_PEBre agent	284.092 7701	175.184	C6	C06
O=C(O)C[C@@H]1OC(=O)c2ccccc21	AcOH	O=C(CC1OC(=O)c2ccccc21)Nc1cccc(Cl)c1	73polarHbond	Amide494_Molp ort AcOH	301.050 571	192.0423	D6	D06
CC(C)(F)C(=O)O	AcOH	CC(C)(F)C(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide436_Molp ort AcOH	215.051 32	106.043	E6	E06
C#CCCCC(=O)O	AcOH	C#CCCCC(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide662_Molp ort AcOH	221.060 742	112.0524	F6	F06
CO[C@H]1CC[C@@H](C(=O)O)CC1	AcOH	COC1CCC(C(=O)Nc2cccc(Cl)c2)CC1	topdiverse	Amide659_Molp ort AcOH	267.102 606	158.0943	G6	N/A
C#CC[C@H](NC(=O)OC(C)(C)C)C(=O)O	AcOH	C#CCC(NC(=O)OC(C)(C)C)C(=O)Nc1cccc(Cl)c1	Greasy pocket	Amide150_Molp ort AcOH	322.108 42	213.1001	H6	H06
O=C1C[C@H](C(=O)O)CN1	AcOH	O=C1CC(C(=O)Nc2cccc(Cl)c2)CN1	Best Pareto and diverse	Amide3_Molport AcOH	238.050 905	129.0426	A7	A07
CCOC(=O)N1CSC[C@H]1C(=O)O	AcOH	CCOC(=O)N1CSCC1C(=O)Nc1cccc(Cl)c1	73 polar / H-bond	Amide992_Molp ort AcOH	314.049 191	205.0409	B7	B07
O=C(O)CNC(=O)c1ccc(O)cc1	AcOH	O=C(CNC(=O)c1ccc(O)cc1)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide13_Molpo rt AcOH	304.061 47	195.0532	C7	C07
O=C(O)Cn1cc2ccccc2n1	AcOH	O=C(Cn1cc2ccccc2n1)Nc1cccc(Cl)c1	Greasy pocket	Amide1114_Mol port AcOH	285.066 89	176.0586	D7	D07
CN1CCN(c2cc(C(=O)O)ccn2)CC1	AcOH	CN1CCN(c2cc(C(=O)Nc3cccc(Cl)c3)ccn2)CC1	73 polar / H-bond	Amide697_Molp ort AcOH	330.124 739	221.1164	E7	E07
CC(C)(C)OC(=O)N1CC[C@@](C)(C(=O)O)C1	AcOH	CC(C)(C)OC(=O)N1CCC(C)(C(=O)Nc2cccc(Cl)c2)C1	73 polar / H-bond	Amide1328_Mol port AcOH	338.139 72	229.1314	F7	N/A

AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
O=C(O)C1(O)CCCC1	AcOH	O=C(Nc1cccc(Cl)c1)C1(O)CCCC1	Best Pareto and diverse	Amide231_Molport AcOH	239.071306	130.063	G7	N/A
C[C@H](Cn1cnc2ccccc21)C(=O)O	AcOH	CC(Cn1cnc2ccccc21)C(=O)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide683_Molport AcOH	313.09819	204.0899	H7	H07
O=C(O)CN1C(=O)CSc2ccccc21	AcOH	O=C(CN1C(=O)CSc2ccccc21)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide578_Molport AcOH	332.038626	223.0303	A8	N/A
O=C(O)Cc1ccco1	AcOH	O=C(Cc1ccco1)Nc1cccc(Cl)c1	Best Pareto and diverse	Amide524_Molport AcOH	235.040006	126.0317	B8	B08
CC(C)(C)OC(=O)N1CC(F)(F)C[C@H]1C(=O)O	AcOH	CC(C)(C)OC(=O)N1CC(F)(F)CC1C(=O)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide1439_Molport AcOH	360.105227	251.0969	C8	C08
CC(C)(C(=O)O)c1ccc(O)cc1	AcOH	CC(C)(C(=O)Nc1cccc(Cl)c1)c1ccc(O)cc1	Best Pareto and diverse	Amide1437_Molport AcOH	289.086956	180.0786	D8	N/A
Cc1cc(=O)c(C(=O)O)nn1-c1ccccc1F	AcOH	Cc1cc(=O)c(C(=O)Nc2cccc(Cl)c2)nn1-c1ccccc1F	Greasy pocket	Amide781_Molport AcOH	357.068033	248.0597	E8	E08
O=C(O)CNC(=O)c1ccc2c(c1)OCO2	AcOH	O=C(CNC(=O)c1ccc2c(c1)OCO2)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide46_Molport AcOH	332.056385	223.0481	F8	F08
C[C@H](C(=O)O)n1c(=O)oc2ccccc21	AcOH	CC(C(=O)Nc1cccc(Cl)c1)n1c(=O)oc2ccccc21	Greasy pocket	Amide694_Molport AcOH	316.06147	207.0532	G8	N/A
O=C(O)c1cccc(-c2nnco2)c1	AcOH	O=C(Nc1cccc(Cl)c1)c1cccc(-c2nnco2)c1	73 polar / H-bond	Amide1215_Molport AcOH	299.046154	190.0378	H8	N/A
O=C(O)[C@H]1CS[C@@H](Cc2ccccc2F)C(=O)N1	AcOH	O=C(Nc1cccc(Cl)c1)C1CSC(Cc2ccccc2F)C(=O)N1	Greasy pocket	Amide106_Molport AcOH	378.060505	269.0522	A9	N/A
O=C(O)c1noc(-c2ccc(F)cc2)n1	AcOH	O=C(Nc1cccc(Cl)c1)c1noc(-c2ccc(F)cc2)n1	Worst Pareto and diverse	Amide852_Molport AcOH	317.036732	208.0284	B9	N/A
O=C(O)CCC(=O)N1CC(=O)Nc2ccccc21	AcOH	O=C(CCC(=O)N1CC(=O)Nc2ccccc21)Nc1cccc(Cl)c1	73 polar / H-bond	Amide1_Molport AcOH	357.088019	248.0797	C9	C09
O=C(O)[C@@H]1CCCN(C(=O)c2ccccn2)C1	AcOH	O=C(Nc1cccc(Cl)c1)C1CCCN(C(=O)c2ccccn2)C1	Worst Pareto and diverse	Amide959_Molport AcOH	343.108754	234.1004	D9	D09
O=C(O)[C@@H]1CC(=O)N(C2CCCCCC2)C1	AcOH	O=C(Nc1cccc(Cl)c1)C1CC(=O)N(C2CCCCCC2)C1	Worst Pareto and diverse	Amide1205_Molport AcOH	334.144806	225.1365	E9	E09
Cc1ccc(C(=O)NCC(=O)O)s1	AcOH	Cc1ccc(C(=O)NCC(=O)Nc2cccc(Cl)c2)s1	Best Pareto and diverse	Amide44_Molport AcOH	308.038626	199.0303	F9	F09
O=C(O)[C@@H]1CCCN(C(=O)Cc2ccc(F)cc2)C1	AcOH	O=C(Nc1cccc(Cl)c1)C1CCCN(C(=O)Cc2ccc(F)cc2)C1	73 polar / H-bond	Amide888_Molport AcOH	374.119734	265.1114	G9	G09
O=C(O)CCn1c(=O)[nH]c(=O)c2ccccc21	AcOH	O=C(CCn1c(=O)[nH]c(=O)c2ccccc21)Nc1ccccc(Cl)c1	Worst Pareto and diverse	Amide676_Molport AcOH	343.072369	234.0641	H9	N/A

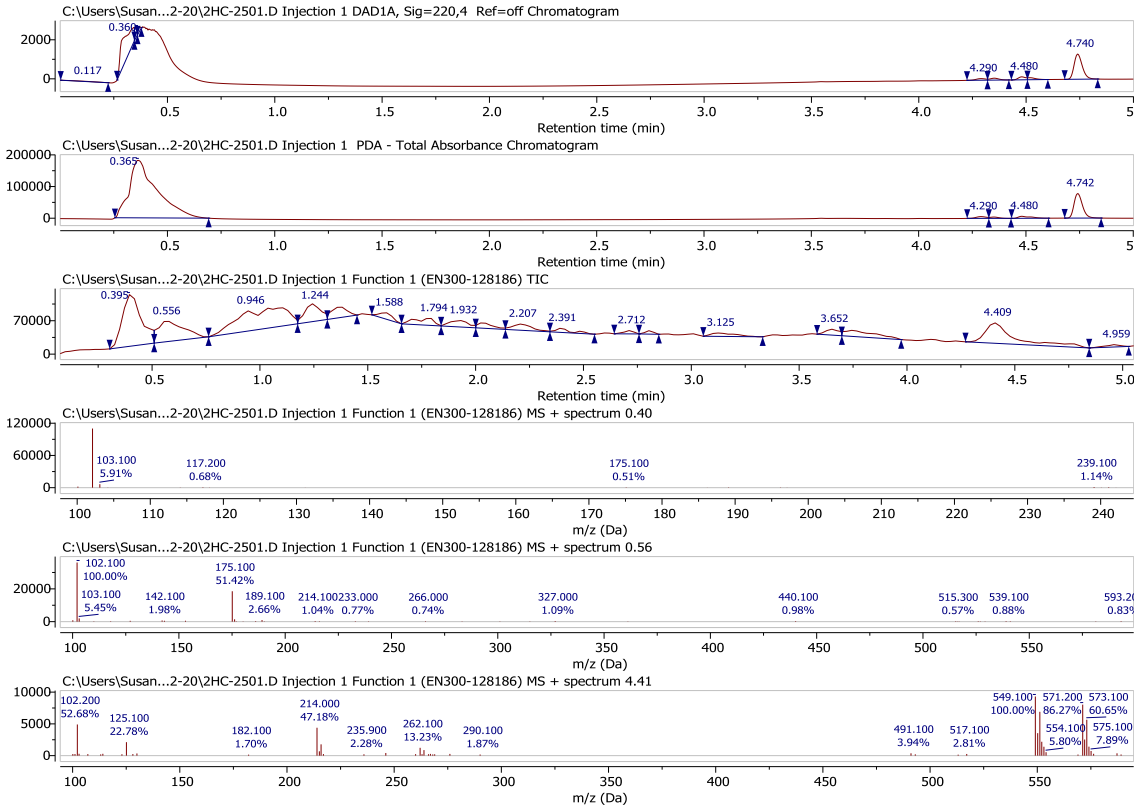
AcX SMILES	AcX type	Amide SMILES	Hypothesis	ID	Amide MW	Reagent MW	Reaction well location	Soaked well location
O=C(O)c1cccn1C1CC1	AcOH	O=C(Nc1cccc(Cl)c1)c1cccn1C1CC1	Greasy pocket	Amide1180_Molport AcOH	260.071641	151.0633	A10	N/A
Cc1nc(C[N@]2CC[C@H](C(=O)O)CC2)cs1	AcOH	Cc1nc(CN2CCC(C(=O)Nc3cccc(Cl)c3)CC2)cs1	Greasy pocket	Amide801_Molport AcOH	349.101561	240.0932	B10	B10
O=C(O)CC[C@@H]1Cc2ccccc2NC1=O	AcOH	O=C(CCC1Cc2ccccc2NC1=O)Nc1cccc(Cl)c1	Greasy pocket	Amide57_Molport AcOH	328.097855	219.0895	C10	C10
O=C(O)CN1C(=O)CCCc2sccc21	AcOH	O=C(CN1C(=O)CCCc2sccc21)Nc1cccc(Cl)c1	Greasy pocket	Amide1366_Molport AcOH	334.054276	225.046	D10	N/A
O=C(O)[C@H]1Cc2cc(Cl)ccc2O1	AcOH	O=C(Nc1cccc(Cl)c1)C1Cc2cc(Cl)ccc2O1	Greasy pocket	Amide705_Molport AcOH	307.016684	198.0084	E10	E10
O=C(O)CNC(=O)c1ccccn1	AcOH	O=C(CNC(=O)c1ccccn1)Nc1cccc(Cl)c1	Worst Pareto and diverse	Amide61_Molport AcOH	289.061804	180.0535	F10	N/A
O=C(O)[C@H]1CCCC[C@H]1C(=O)N[C@@H]1CCS(=O)(=O)C1	AcOH	O=C(Nc1cccc(Cl)c1)C1CCCCC1C(=O)NC1CCS(=O)(=O)C1	73 polar / H-bond	Amide76_Molport AcOH	398.106706	289.0984	G10	N/A
O=C(O)CNC(=O)NC1CCCCC1	AcOH	O=C(CNC(=O)NC1CCCCC1)Nc1cccc(Cl)c1	Greasy pocket	Amide63_Molport AcOH	309.124405	200.1161	H10	N/A

Table A-2. Summary of the 80 carboxylic acids used in the coupling reactions.

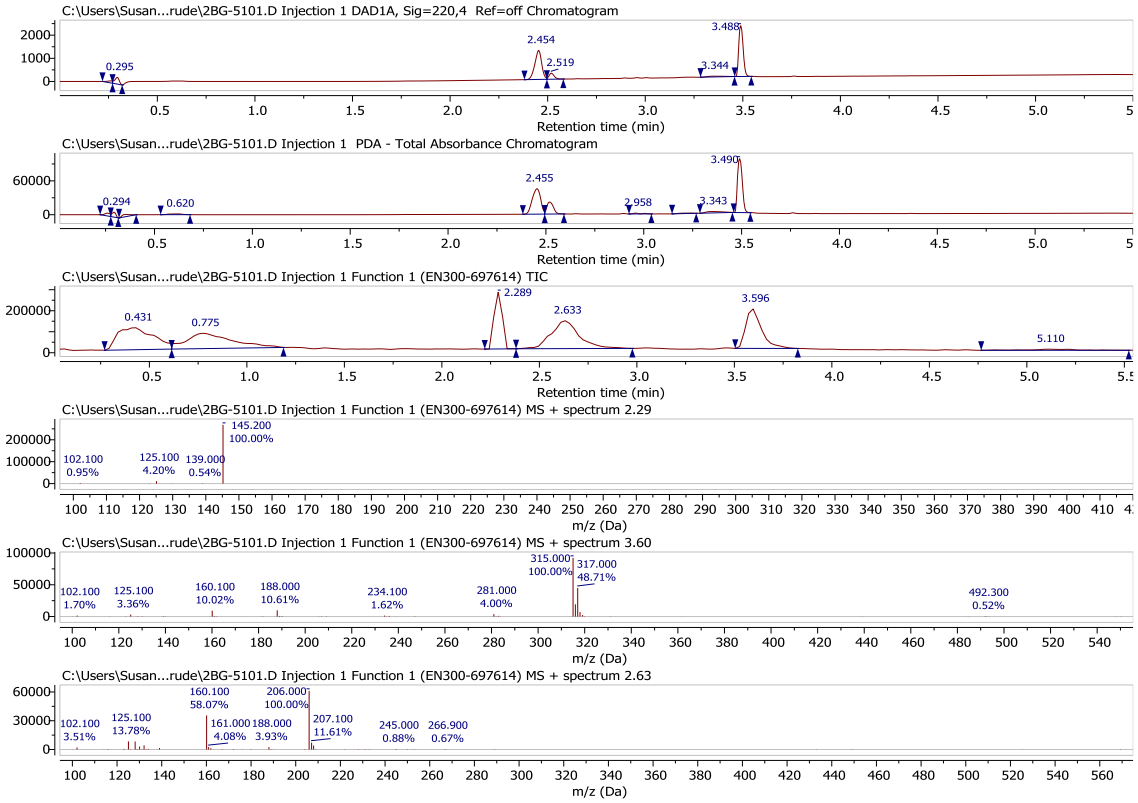
A.3 LCMS traces for the six amide follow-up hits

These are the LCMS traces for the six crude reaction mixtures, which when soaked into NUDT7 crystals, resulted in the X-ray crystal structures of the product amide in the NUDT7 binding site (Figure 2.7b – g). For each crude reaction, a table is shown for the molecular weight (MW) and retention time (RT) for the starting material and product, if seen in the LCMS traces. The results of the LCMS are shown as six panels, which are: (1) the DAD chromatogram; (2) the PDA-Total Absorption Chromatogram; (3) the Total Ion Current (TIC) chromatogram; (4)-(6) the mass spectrum for the three highest peaks.

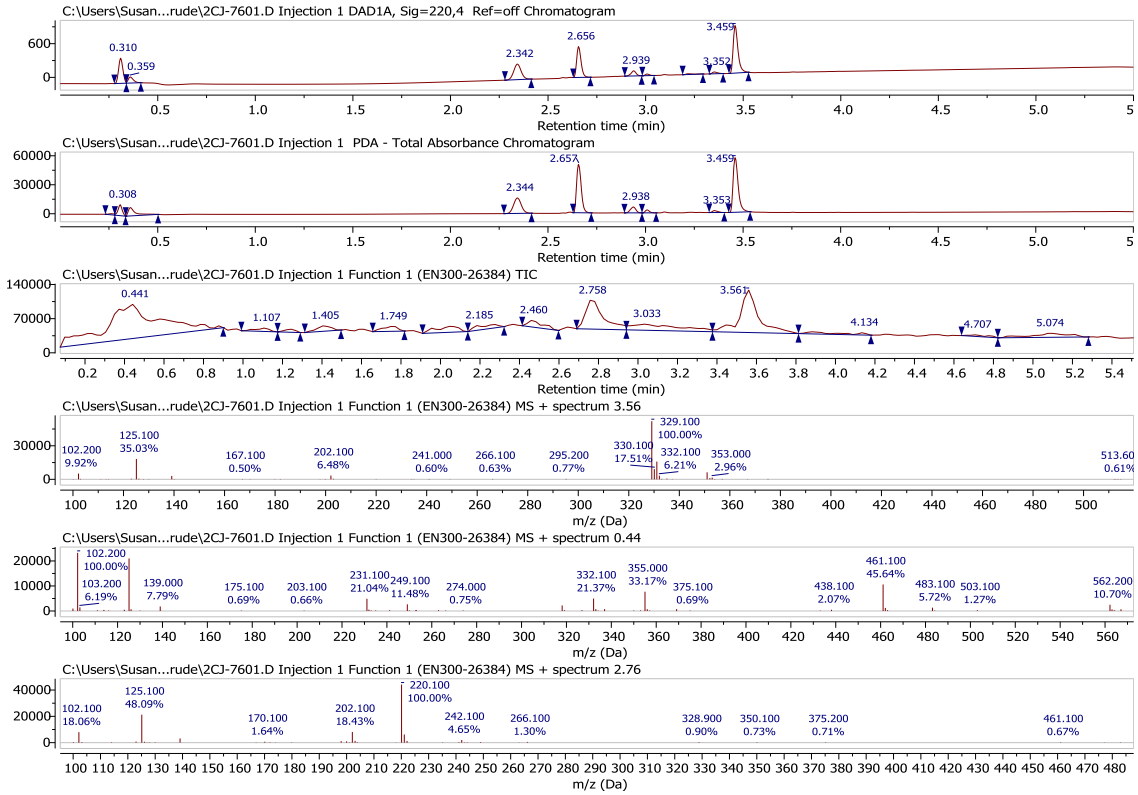
Crystal: x0062	MW	RT
Starting material	122 or 104 for acid	Not seen
Product	213	4.41



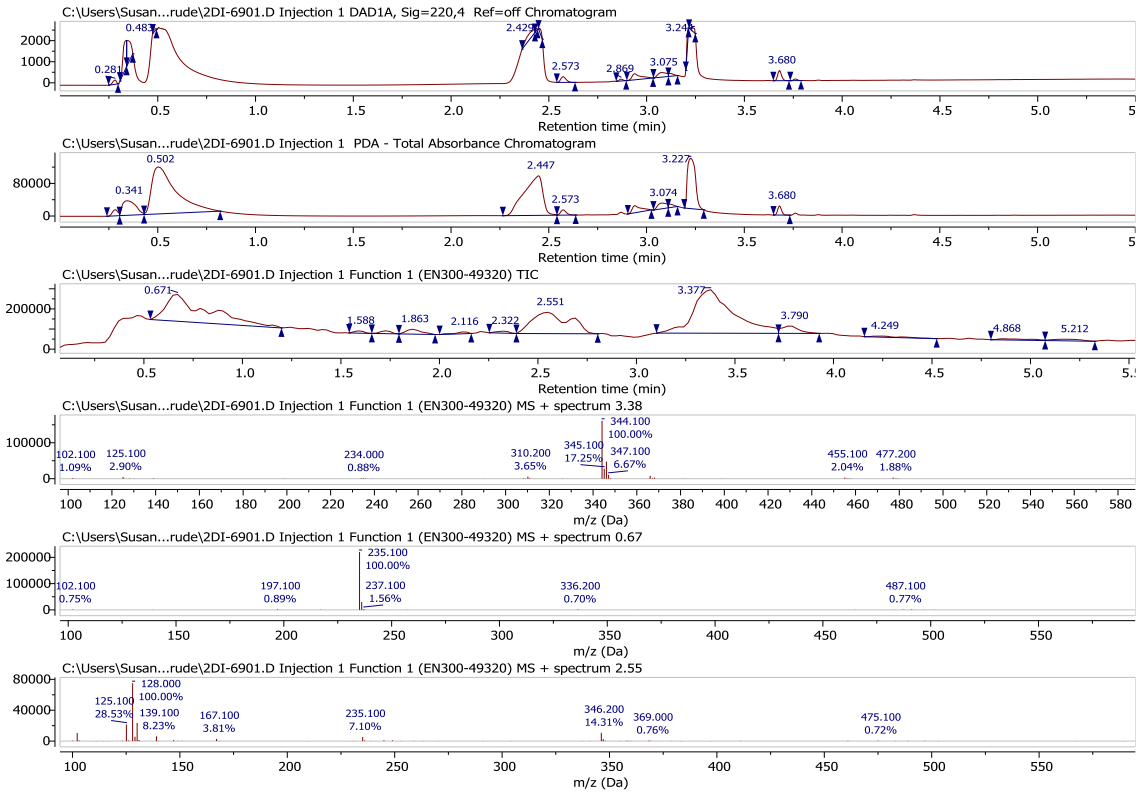
Crystal: x0073	MW	RT
Starting material	205	2.63
Product	314	3.60



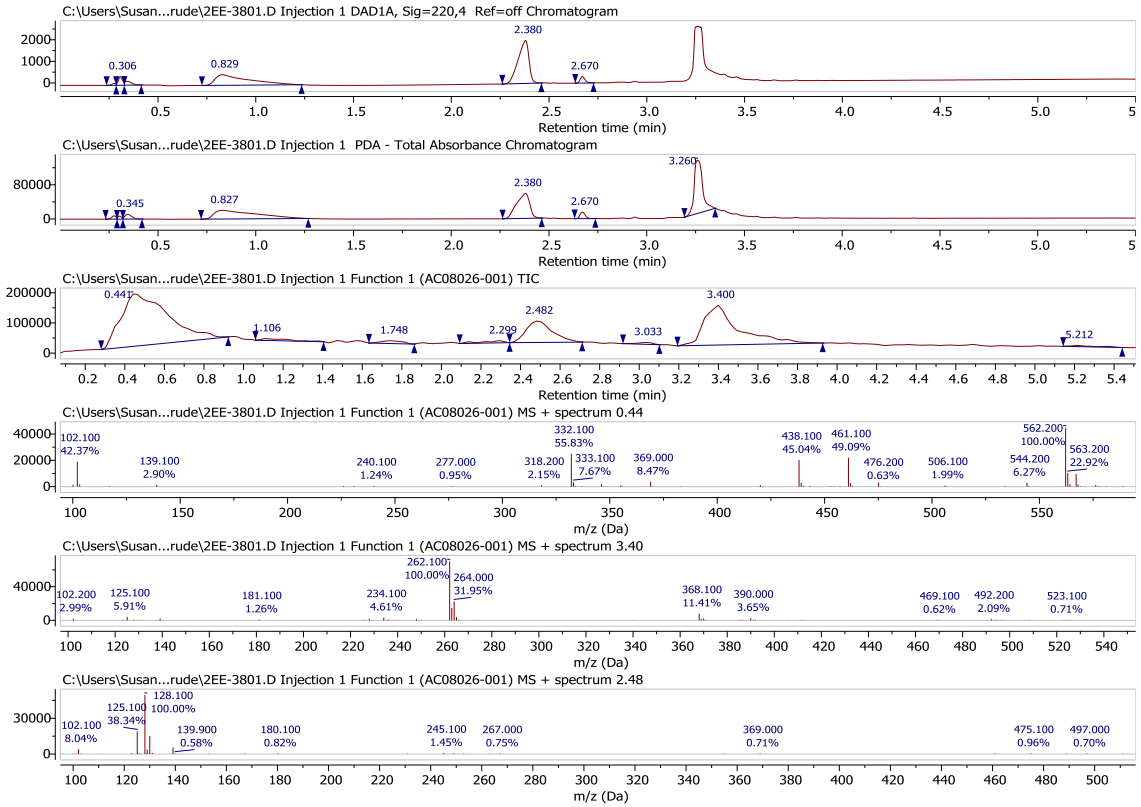
Crystal: x0083	MW	RT
Starting material	219	2.73
Product	328	3.56



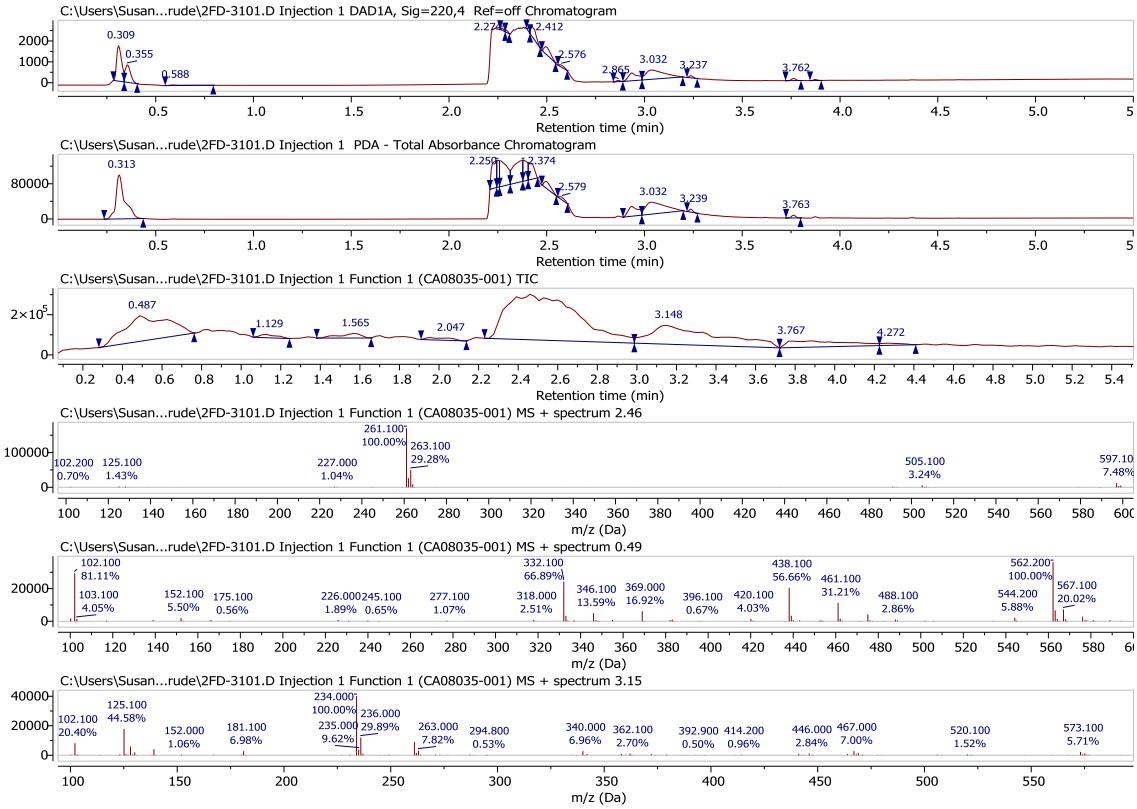
Crystal: x0090	MW	RT
Starting material	234	0.67
Product	343	3.38



Crystal: x0094	MW	RT
Starting material	152	Not seen
Product	261	3.40



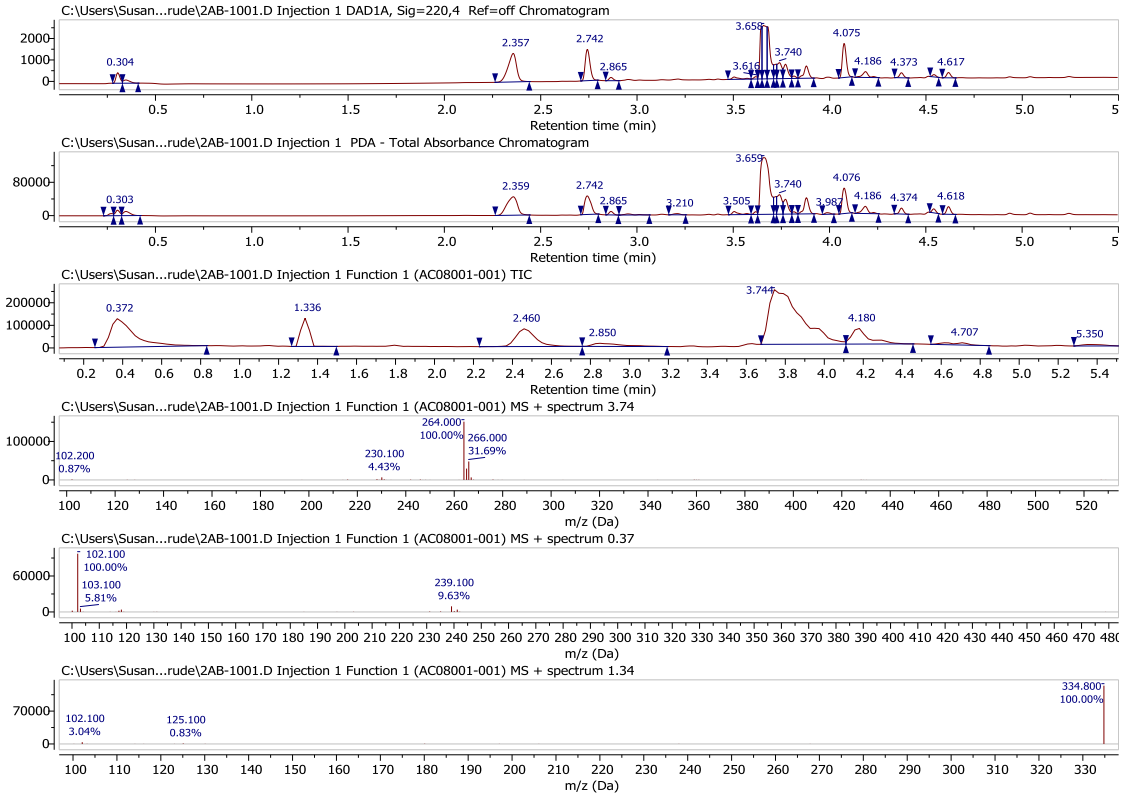
Crystal: x0102	MW	RT
Starting material	151	0.49 (little seen)
Product	260	2.46



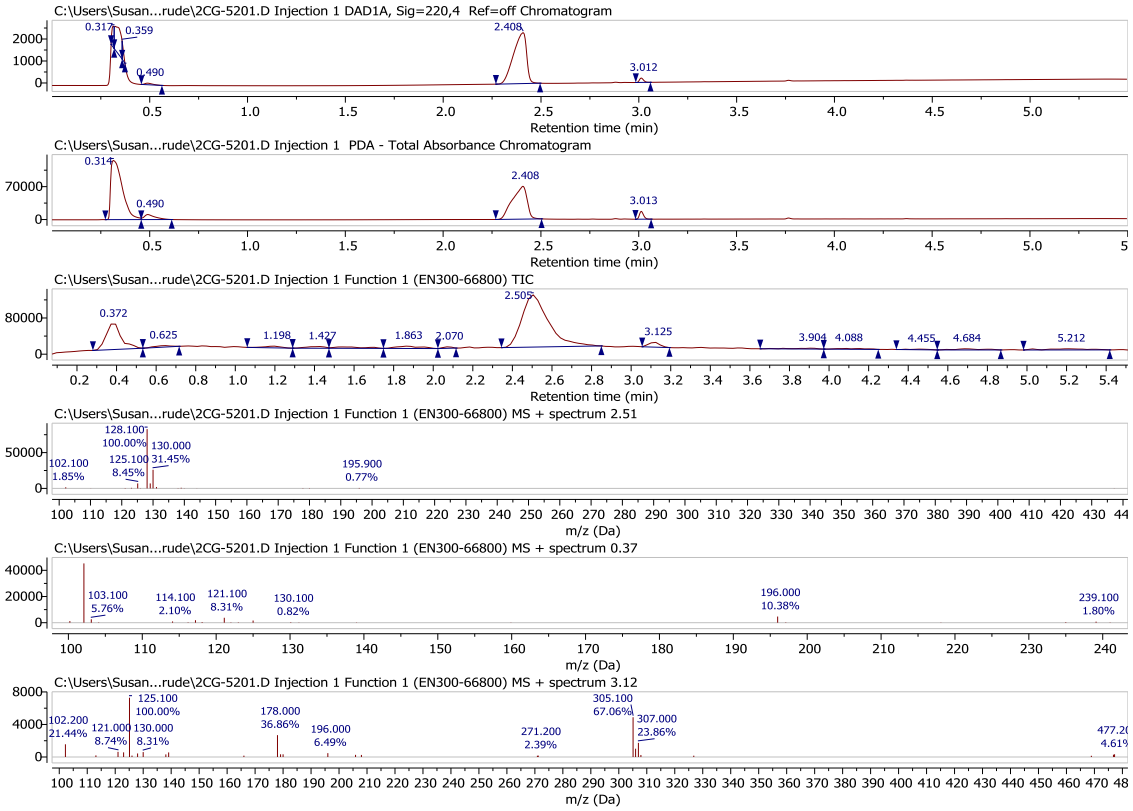
A.4 LCMS traces for the three starting material hits

These are the LCMS traces for the three crude reaction mixtures, which when soaked into NUDT7 crystals, resulted in the X-ray crystal structures of the starting material in the NUDT7 binding site (Figure 2.8). For each crude reaction, a table is shown for the molecular weight (MW) and retention time (RT) for the starting material and product, if seen in the LCMS traces. The results of the LCMS are shown as six panels, which are: (1) the DAD chromatogram; (2) the PDA-Total Absorption Chromatogram; (3) the Total Ion Current (TIC) chromatogram; (4)-(6) the mass spectrum for the three highest peaks.

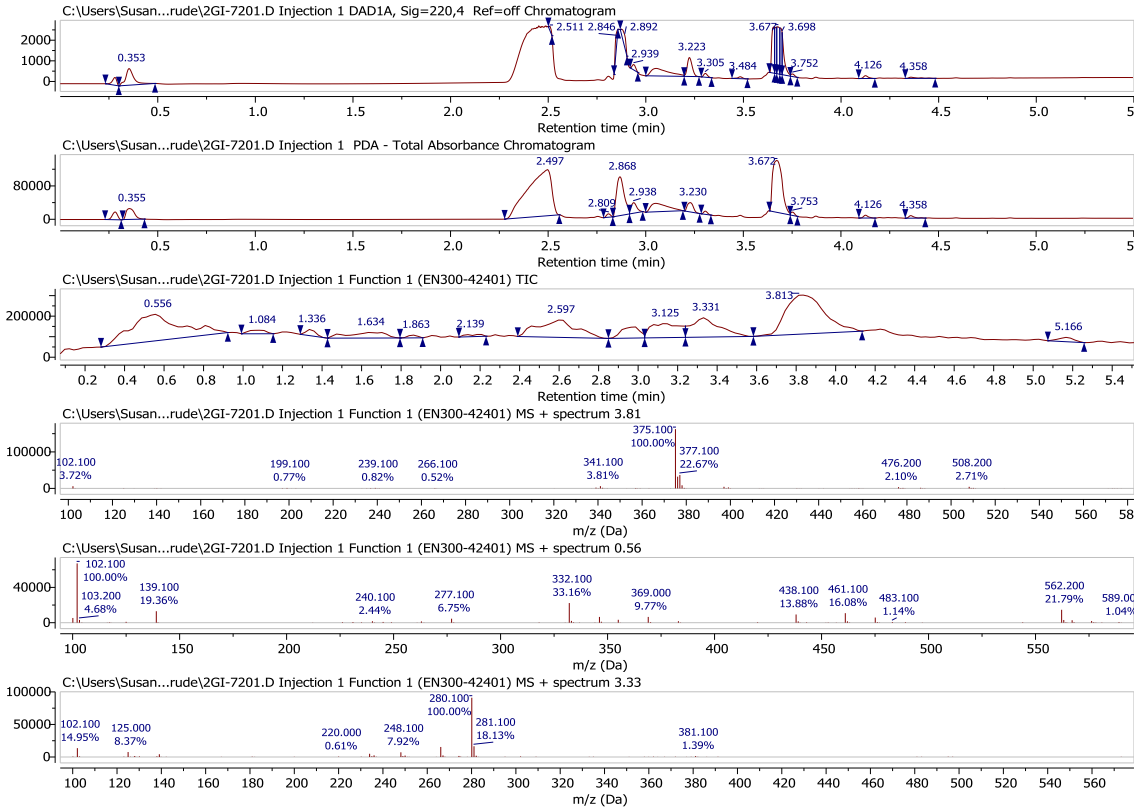
Crystal: x0041	MW	RT
Starting material	154	Not seen
Product	263	3.70



Crystal: x0080	MW	RT
Starting material	195	0.37
Product	305	3.12



Crystal: x0107	MW	RT
Starting material	265	2.80
Product	375	3.81



Appendix B : Chapter 4

B.1 Common crystallographic additives that were excluded

1PE, 3CO, ACE, ACT, ACY, AML, ARF, ARJ, AZI, BCN, BCT, BEZ, BME, BOG, BR, CA, CCN, CIT, CL, CNN, CO, CO2, CO3, CU, DMF, DMS, DTT, DTU, EDO, EOH, FMT, GOL, H2S, HG, IMD, IOD, IPA, IR, K, KCS, LCP, MES, MG, MGF, MLA, MMC, MN, MRD, NAN, NI, NO3, OCS, OH, ONM, OXY, PEG, PG4, PGE, PO4, RU, SAR, SCN, SEP, SGM, SO3, SO4, SPK, TAR, TLA, TMO, TPO, TRS, URE, XE, Y1, ZN

Target	Clustered Ligands	Matched Decoys	Docked Actives Parsed	Proportion of Docked Actives Parsed	Docked Decoys Parsed	Proportion of Docked Decoys Parsed
AA2AR	482	31550	482	1.00	31489	1.00
ABL1	182	10750	181	0.99	10740	1.00
ACE	282	16900	280	0.99	16855	1.00
ACES	453	26250	453	1.00	26220	1.00
ADA	93	5450	93	1.00	5439	1.00
ADA17	532	35900	532	1.00	33601	0.94
ADRB1	247	15850	247	1.00	15836	1.00
ADRB2	231	15000	231	1.00	11254	0.75
AKT1	293	16450	293	1.00	16415	1.00
AKT2	117	6900	117	1.00	6886	1.00
ALDR	159	9000	159	1.00	8983	1.00
AMPC	48	2850	48	1.00	2823	0.99
ANDR	269	14350	269	1.00	14336	1.00
AOFB	122	6900	122	1.00	6889	1.00
BACE1	283	18100	283	1.00	18072	1.00
BRAF	152	9950	152	1.00	9937	1.00
CAH2	492	31172	492	1.00	31126	1.00
CASP3	199	10700	198	0.99	10685	1.00
CDK2	474	27850	473	1.00	27824	1.00
COMT	41	3850	41	1.00	3837	1.00
CP2C9	120	7450	120	1.00	7442	1.00
CP3A4	170	11800	167	0.98	11785	1.00
CSF1R	166	12150	166	1.00	12132	1.00
CXCR4	40	3406	40	1.00	3395	1.00
DEF	102	5700	102	1.00	5687	1.00
DHI1	387	19350	330	0.85	19329	1.00
DPP4	533	40950	533	1.00	40910	1.00
DRD3	480	34050	478	1.00	34016	1.00
DYR	231	17200	231	1.00	17165	1.00
EGFR	542	35050	542	1.00	35011	1.00
ESR1	383	20685	383	1.00	20653	1.00
ESR2	367	20199	367	1.00	20178	1.00
FA10	537	28325	537	1.00	20015	0.71
FA7	114	6250	114	1.00	6236	1.00
FABP4	47	2750	47	1.00	2740	1.00
FAK1	100	5350	100	1.00	5343	1.00
FGFR1	139	8700	139	1.00	8691	1.00
FKB1A	111	5800	111	1.00	5792	1.00
FNTA	592	51500	592	1.00	51418	1.00
FPPS	85	8850	85	1.00	8812	1.00
GCR	258	15000	258	1.00	14984	1.00
GLCM	54	3800	54	1.00	3794	1.00
GRIA2	158	11845	158	1.00	11826	1.00
GRIK1	101	6550	100	0.99	6537	1.00
HDAC2	185	10300	185	1.00	10291	1.00
HDAC8	170	10450	170	1.00	10436	1.00
HIVINT	100	6650	100	1.00	6632	1.00
HIVPR	536	35750	536	1.00	35678	1.00
HIVRT	338	18891	337	1.00	18870	1.00
HMDH	170	8750	170	1.00	8734	1.00
HS90A	88	4850	88	1.00	4842	1.00
HXK4	92	4700	92	1.00	4686	1.00
IGF1R	148	9300	148	1.00	9282	1.00
INHA	44	2300	43	0.98	2293	1.00
ITAL	138	8500	137	0.99	8482	1.00
JAK2	130	6500	107	0.82	6487	1.00
KIF11	116	6850	116	1.00	6838	1.00

Target	Clustered Ligands	Matched Decoys	Docked Actives Parsed	Proportion of Docked Actives Parsed	Docked Decoys Parsed	Proportion of Docked Decoys Parsed
KIT	166	10450	166	1.00	10440	1.00
KITH	57	2850	57	1.00	2839	1.00
KPCB	135	8700	135	1.00	8684	1.00
LCK	420	27400	420	1.00	27368	1.00
LKHA4	171	9450	170	0.99	9443	1.00
MAPK2	101	6150	101	1.00	6139	1.00
MCR	94	5150	94	1.00	5138	1.00
MET	166	11250	166	1.00	11232	1.00
MK01	79	4550	79	1.00	4541	1.00
MK10	104	6600	104	1.00	6591	1.00
MK14	578	35850	578	1.00	35801	1.00
MMP13	572	37200	572	1.00	37117	1.00
MP2K1	121	8150	120	0.99	8138	1.00
NOS1	100	8050	100	1.00	8044	1.00
NRAM	98	6200	98	1.00	6194	1.00
PA2GA	99	5150	98	0.99	5138	1.00
PARP1	508	30050	508	1.00	30022	1.00
PDE5A	398	27550	398	1.00	27513	1.00
PGH1	195	10800	195	1.00	10788	1.00
PGH2	435	23150	435	1.00	23127	1.00
PLK1	107	6800	106	0.99	6789	1.00
PNPH	103	6950	102	0.99	6944	1.00
PPARA	373	19399	373	1.00	19349	1.00
PPARD	240	12250	240	1.00	12219	1.00
PPARG	484	25300	484	1.00	25250	1.00
PRGR	293	15650	293	1.00	15632	1.00
PTN1	130	7250	130	1.00	7237	1.00
PUR2	50	2700	50	1.00	2683	0.99
PYGM	77	3950	77	1.00	3936	1.00
PYRD	111	6450	111	1.00	6436	1.00
RENI	104	6958	103	0.99	6950	1.00
ROCK1	100	6300	100	1.00	6288	1.00
RXRA	131	6950	131	1.00	6926	1.00
SAHH	63	3450	63	1.00	3445	1.00
SRC	524	34500	524	1.00	34441	1.00
TGFR1	133	8500	133	1.00	8488	1.00
THB	103	7450	103	1.00	7432	1.00
THRB	461	27004	461	1.00	26941	1.00
TRY1	449	25980	449	1.00	25906	1.00
TRYB1	148	7650	148	1.00	7636	1.00
TYSY	109	6750	109	1.00	6729	1.00
UROK	162	9850	162	1.00	9836	1.00
VGFR2	409	24950	409	1.00	24921	1.00
WEE1	102	6150	101	0.99	6142	1.00
XIAP	100	5150	99	0.99	5137	1.00

Table B-1. Number of molecules successfully parsed by RDKit for each DUD-E target.

Target	AutoDock Vina	SuCOS
AA2AR	0.652	**0.805**
ABL1	**0.777**	0.687
ACE	0.564	**0.565**
ACES	**0.776**	0.563
ADA	0.573	**0.864**
ADA17	0.712	**0.901**
ADRB1	0.731	**0.850**
ADRB2	0.714	**0.840**
AKT1	**0.764**	0.699
AKT2	**0.788**	0.733
ALDR	**0.737**	0.684
AMPC	0.608	**0.731**
ANDR	0.627	**0.829**
AOFB	**0.783**	0.563
BACE1	0.724	**0.803**
BRAF	**0.865**	0.770
CAH2	0.587	**0.865**
CASP3	**0.697**	0.679
CDK2	0.718	**0.796**
COMT	0.631	**0.990**
CP2C9	0.623	**0.636**
CP3A4	**0.603**	0.543
CSF1R	0.685	**0.742**
CXCR4	0.596	**0.875**
DEF	0.764	**0.961**
DHI1	**0.772**	0.699
DPP4	0.625	**0.711**
DRD3	**0.749**	0.644
DYR	0.770	**0.812**
EGFR	0.642	**0.779**
ESR1	**0.828**	0.772
ESR2	**0.796**	0.783
FA10	0.833	**0.850**
FA7	0.910	**0.935**
FABP4	**0.783**	0.662
FAK1	0.809	**0.825**
FGFR1	**0.677**	0.672
FKB1A	0.770	**0.787**
FNTA	0.654	**0.682**
FPPS	0.286	**0.983**
GCR	0.639	**0.763**
GLCM	0.491	**0.783**
GRIA2	0.746	**0.833**
GRIK1	0.593	**0.864**
HDAC2	**0.849**	0.814
HDAC8	**0.823**	0.802
HIVINT	**0.707**	0.629
HIVPR	0.716	**0.883**
HIVRT	0.679	**0.705**
HMDH	0.787	**0.937**
HS90A	0.262	**0.779**

Target	AutoDock Vina	SuCOS
HXK4	0.571	**0.697**
IGF1R	**0.834**	0.730
INHA	**0.715**	0.625
ITAL	**0.600**	0.573
JAK2	0.775	**0.821**
KIF11	**0.854**	0.784
KIT	**0.779**	0.662
KITH	0.737	**0.832**
KPCB	0.774	**0.837**
LCK	**0.797**	0.544
LKHA4	0.894	**0.930**
MAPK2	**0.886**	0.865
MCR	0.543	**0.767**
MET	0.811	**0.867**
MK01	**0.857**	0.654
MK10	**0.749**	0.677
MK14	**0.740**	0.649
MMP13	0.656	**0.884**
MP2K1	0.537	**0.648**
NOS1	0.588	**0.731**
NRAM	0.541	**0.900**
PA2GA	0.620	**0.810**
PARP1	0.857	**0.938**
PDE5A	0.664	**0.765**
PGH1	**0.643**	0.634
PGH2	0.772	**0.789**
PLK1	0.645	**0.698**
PNPH	0.882	**0.952**
PPARA	**0.863**	0.845
PPARD	**0.758**	0.710
PPARG	0.795	**0.829**
PRGR	**0.670**	0.564
PTN1	**0.834**	0.782
PUR2	0.907	**0.980**
PYGM	**0.600**	**0.600**
PYRD	**0.833**	0.814
RENI	0.664	**0.675**
ROCK1	0.719	**0.791**
RXRA	0.807	**0.831**
SAHH	0.803	**0.945**
SRC	0.648	**0.725**
TGFR1	0.903	**0.957**
THB	0.823	**0.901**
THRB	0.766	**0.815**
TRY1	**0.798**	0.772
TRYB1	**0.707**	0.646
TYSY	**0.868**	0.852
UROK	0.771	**0.889**
VGFR2	**0.767**	0.748
WEE1	0.957	**0.994**
XIAP	0.730	**0.941**

Table B-2. DUD-E AUC ROC for AutoDock Vina and SuCOS.

Target	AutoDock Vina	SuCOS
AA2AR	0.652	**0.805**
ABL1	**0.777**	0.687
ACE	0.564	**0.565**
ACES	**0.776**	0.563
ADA	0.573	**0.864**
ADA17	0.712	**0.901**
ADRB1	0.731	**0.850**
ADRB2	0.714	**0.840**
AKT1	**0.764**	0.699
AKT2	**0.788**	0.733
ALDR	**0.737**	0.684
AMPC	0.608	**0.731**
ANDR	0.627	**0.829**
AOFB	**0.783**	0.563
BACE1	0.724	**0.803**
BRAF	**0.865**	0.770
CAH2	0.587	**0.865**
CASP3	**0.697**	0.679
CDK2	0.718	**0.796**
COMT	0.631	**0.990**
CP2C9	0.623	**0.636**
CP3A4	**0.603**	0.543
CSF1R	0.685	**0.742**
CXCR4	0.596	**0.875**
DEF	0.764	**0.961**
DHI1	**0.772**	0.699
DPP4	0.625	**0.711**
DRD3	**0.749**	0.644
DYR	0.770	**0.812**
EGFR	0.642	**0.779**
ESR1	**0.828**	0.772
ESR2	**0.796**	0.783
FA10	0.833	**0.850**
FA7	0.910	**0.935**
FABP4	**0.783**	0.662
FAK1	0.809	**0.825**
FGFR1	**0.677**	0.672
FKB1A	0.770	**0.787**
FNTA	0.654	**0.682**
FPPS	0.286	**0.983**
GCR	0.639	**0.763**
GLCM	0.491	**0.783**
GRIA2	0.746	**0.833**
GRIK1	0.593	**0.864**
HDAC2	**0.849**	0.814
HDAC8	**0.823**	0.802
HIVINT	**0.707**	0.629
HIVPR	0.716	**0.883**
HIVRT	0.679	**0.705**
HMDH	0.787	**0.937**
HS90A	0.262	**0.779**

Target	AutoDock Vina	SuCOS
HXK4	0.571	**0.697**
IGF1R	**0.834**	0.730
INHA	**0.715**	0.625
ITAL	**0.600**	0.573
JAK2	0.775	**0.821**
KIF11	**0.854**	0.784
KIT	**0.779**	0.662
KITH	0.737	**0.832**
KPCB	0.774	**0.837**
LCK	**0.797**	0.544
LKHA4	0.894	**0.930**
MAPK2	**0.886**	0.865
MCR	0.543	**0.767**
MET	0.811	**0.867**
MK01	**0.857**	0.654
MK10	**0.749**	0.677
MK14	**0.740**	0.649
MMP13	0.656	**0.884**
MP2K1	0.537	**0.648**
NOS1	0.588	**0.731**
NRAM	0.541	**0.900**
PA2GA	0.620	**0.810**
PARP1	0.857	**0.938**
PDE5A	0.664	**0.765**
PGH1	**0.643**	0.634
PGH2	0.772	**0.789**
PLK1	0.645	**0.698**
PNPH	0.882	**0.952**
PPARA	**0.863**	0.845
PPARD	**0.758**	0.710
PPARG	0.795	**0.829**
PRGR	**0.670**	0.564
PTN1	**0.834**	0.782
PUR2	0.907	**0.980**
PYGM	**0.600**	**0.600**
PYRD	**0.833**	0.814
RENI	0.664	**0.675**
ROCK1	0.719	**0.791**
RXRA	0.807	**0.831**
SAHH	0.803	**0.945**
SRC	0.648	**0.725**
TGFR1	0.903	**0.957**
THB	0.823	**0.901**
THRB	0.766	**0.815**
TRY1	**0.798**	0.772
TRYB1	**0.707**	0.646
TYSY	**0.868**	0.852
UROK	0.771	**0.889**
VGFR2	**0.767**	0.748
WEE1	0.957	**0.994**
XIAP	0.730	**0.941**

Table B-3. DUD-E ROC enrichment at 0.5% for AutoDock Vina and SuCOS.

Target	AutoDock Vina	SuCOS
AA2AR	2.490	**21.162**
ABL1	**13.812**	12.707
ACE	2.857	**5.000**
ACES	**16.998**	2.649
ADA	1.075	**40.860**
ADA17	31.015	**65.977**
ADRB1	4.049	**29.960**
ADRB2	3.463	**16.017**
AKT1	5.461	**5.802**
AKT2	**29.060**	10.256
ALDR	11.321	**16.352**
AMPC	0.000	**10.417**
ANDR	20.446	**26.022**
AOFB	**7.377**	1.639
BACE1	4.947	**14.841**
BRAF	19.079	**27.632**
CAH2	0.000	**4.472**
CASP3	1.515	**7.071**
CDK2	9.937	**16.279**
COMT	4.878	**92.683**
CP2C9	3.333	**4.167**
CP3A4	1.796	**3.593**
CSF1R	1.205	**11.446**
CXCR4	0.000	**42.500**
DEF	11.765	**73.529**
DHI1	4.242	**13.030**
DPP4	0.563	**17.448**
DRD3	**5.649**	2.092
DYR	7.792	**39.827**
EGFR	5.904	**31.365**
ESR1	22.715	**36.292**
ESR2	16.894	**36.785**
FA10	20.670	**26.257**
FA7	14.035	**57.018**
FABP4	31.915	**34.043**
FAK1	21.000	**34.000**
FGFR1	**8.633**	5.755
FKB1A	7.207	**9.009**
FNTA	**2.703**	1.351
FPPS	0.000	**63.529**
GCR	15.891	**22.868**
GLCM	0.000	**25.926**
GRIA2	12.658	**53.797**
GRIK1	5.000	**47.000**
HDAC2	14.054	**17.838**
HDAC8	25.882	**47.647**
HIVINT	**2.000**	**2.000**
HIVPR	4.664	**31.530**
HIVRT	4.154	**9.792**
HMDH	4.706	**74.118**
HS90A	0.000	**44.318**

Target	AutoDock Vina	SuCOS
HXK4	3.261	**28.261**
IGF1R	**17.568**	15.541
INHA	**9.302**	**9.302**
ITAL	0.000	**8.759**
JAK2	15.888	**33.645**
KIF11	**36.207**	33.621
KIT	**4.819**	3.012
KITH	33.333	**63.158**
KPCB	34.074	**53.333**
LCK	**10.714**	7.619
LKHA4	17.647	**50.000**
MAPK2	17.822	**46.535**
MCR	7.447	**18.085**
MET	12.048	**41.566**
MK01	3.797	**26.582**
MK10	**7.692**	0.962
MK14	**6.920**	**6.920**
MMP13	4.021	**52.098**
MP2K1	0.000	**10.000**
NOS1	2.000	**11.000**
NRAM	0.000	**36.735**
PA2GA	**1.020**	**1.020**
PARP1	17.913	**49.803**
PDE5A	11.809	**23.618**
PGH1	**6.667**	2.051
PGH2	31.954	**39.080**
PLK1	0.000	**9.434**
PNPH	13.725	**70.588**
PPARA	6.971	**23.324**
PPARD	1.250	**7.083**
PPARG	6.198	**31.405**
PRGR	**13.993**	9.215
PTN1	**31.538**	16.154
PUR2	4.000	**94.000**
PYGM	**3.896**	0.000
PYRD	22.523	**42.342**
RENI	4.854	**18.447**
ROCK1	8.000	**9.000**
RXRA	**36.641**	9.160
SAHH	26.984	**90.476**
SRC	5.534	**8.779**
TGFR1	11.278	**48.872**
THB	33.981	**48.544**
THRB	1.735	**10.195**
TRY1	2.895	**11.136**
TRYB1	**8.108**	3.378
TYSY	**25.688**	23.853
UROK	8.642	**44.444**
VGFR2	**19.560**	8.802
WEE1	72.277	**98.020**
XIAP	11.111	**62.626**

Table B-4. DUD-E ROC enrichment at 1% for AutoDock Vina and SuCOS.

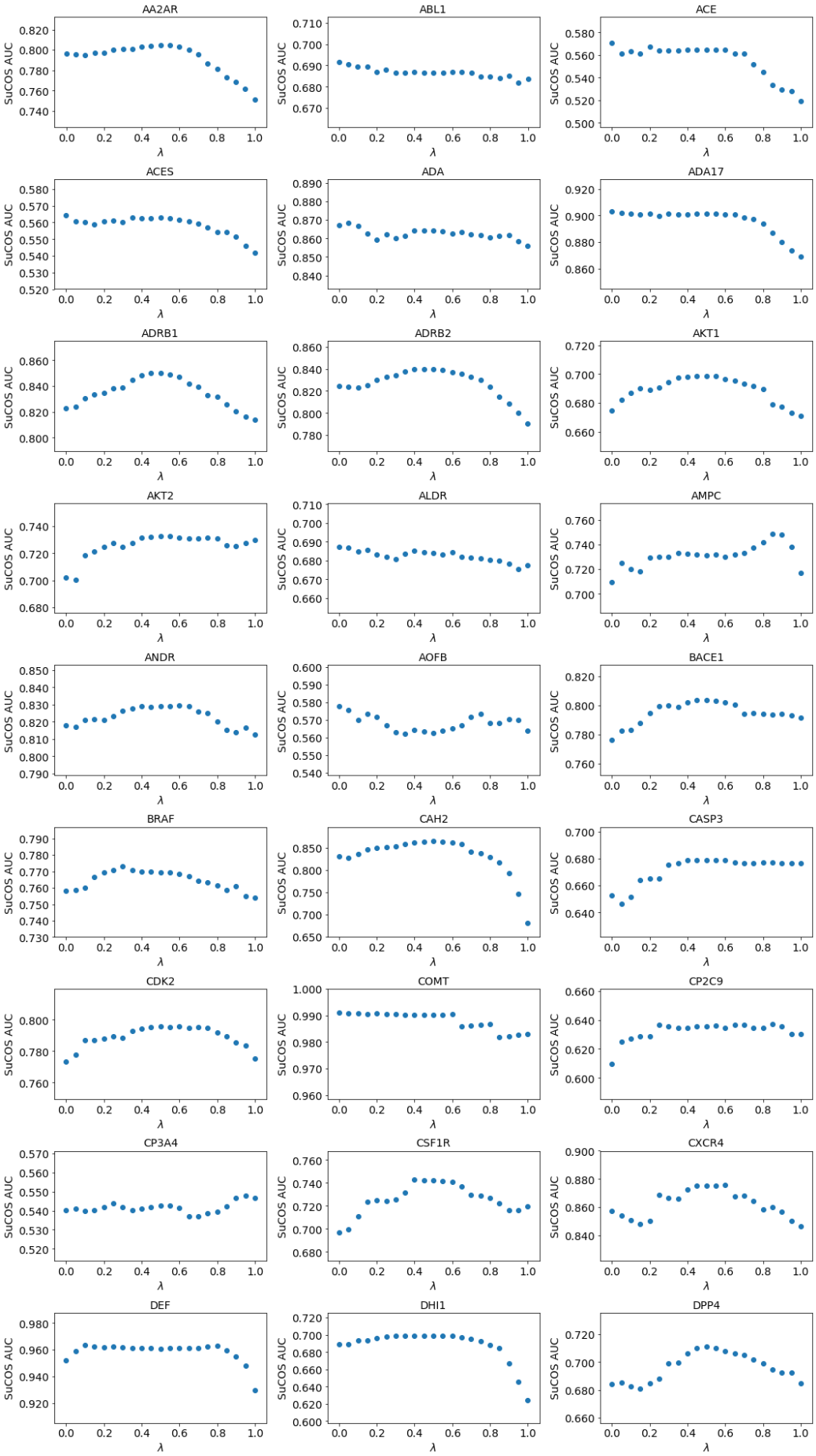
Target	AutoDock Vina	SuCOS
AA2AR	3.423	**14.108**
ABL1	**12.431**	7.182
ACE	2.143	**2.857**
ACES	**11.589**	1.987
ADA	0.538	**23.118**
ADA17	16.823	**34.586**
ADRB1	4.049	**17.814**
ADRB2	3.680	**10.606**
AKT1	**6.826**	3.925
AKT2	**14.957**	6.838
ALDR	6.918	**9.748**
AMPC	1.042	**7.292**
ANDR	12.454	**16.729**
AOFB	**7.787**	1.639
BACE1	4.770	**11.484**
BRAF	13.158	**14.145**
CAH2	0.305	**3.049**
CASP3	2.778	**6.313**
CDK2	6.237	**14.165**
COMT	2.439	**48.780**
CP2C9	2.083	**3.750**
CP3A4	1.796	**3.593**
CSF1R	3.313	**7.229**
CXCR4	0.000	**27.500**
DEF	7.843	**41.176**
DHI1	4.394	**8.182**
DPP4	1.313	**10.507**
DRD3	**4.393**	1.464
DYR	5.844	**21.861**
EGFR	4.705	**17.620**
ESR1	13.969	**20.235**
ESR2	12.125	**21.117**
FA10	13.780	**17.039**
FA7	14.035	**35.088**
FABP4	15.957	**17.021**
FAK1	10.500	**17.500**
FGFR1	**8.633**	7.914
FKB1A	**5.856**	**5.856**
FNTA	**2.534**	1.774
FPPS	0.000	**37.647**
GCR	10.078	**15.310**
GLCM	0.000	**16.667**
GRIA2	8.544	**27.848**
GRIK1	3.500	**26.500**
HDAC2	**11.892**	**11.892**
HDAC8	17.941	**25.000**
HIVINT	1.000	**2.000**
HIVPR	4.478	**20.243**
HIVRT	3.709	**6.380**
HMDH	4.118	**37.059**
HS90A	0.000	**24.432**
HXK4	3.261	**14.130**
IGF1R	**10.811**	9.797
INHA	**9.302**	4.651
ITAL	0.365	**4.745**
JAK2	9.813	**17.757**
KIF11	**21.983**	17.672
KIT	**5.120**	2.711
KITH	18.421	**31.579**
KPCB	21.111	**27.778**
LCK	**7.857**	5.000
LKHA4	16.471	**30.882**
MAPK2	13.366	**24.257**
MCR	4.787	**13.830**
MET	10.542	**23.795**
MK01	4.430	**13.291**
MK10	**6.250**	1.923
MK14	**6.228**	4.844
MMP13	4.458	**28.584**
MP2K1	0.833	**5.000**
NOS1	1.000	**9.000**
NRAM	0.000	**21.939**
PA2GA	0.510	**2.551**
PARP1	13.091	**29.626**
PDE5A	9.799	**12.563**
PGH1	**5.128**	1.282
PGH2	20.460	**22.759**
PLK1	1.887	**7.547**
PNPH	8.824	**35.784**
PPARA	6.568	**14.343**
PPARD	1.875	**5.000**
PPARG	5.579	**19.215**
PRGR	**9.727**	6.314
PTN1	**19.231**	11.154
PUR2	7.000	**47.000**
PYGM	**1.948**	0.000
PYRD	13.063	**23.423**
RENI	4.854	**11.165**
ROCK1	4.500	**6.500**
RXRA	**20.992**	10.687
SAHH	19.048	**45.238**
SRC	4.485	**6.584**
TGFR1	13.158	**31.579**
THB	22.816	**28.641**
THRB	3.905	**7.809**
TRY1	2.673	**7.350**
TRYB1	**5.743**	4.054
TYSY	**16.514**	16.055
UROK	7.716	**24.691**
VGFR2	**11.736**	5.990
WEE1	38.119	**49.010**
XIAP	7.071	**35.859**

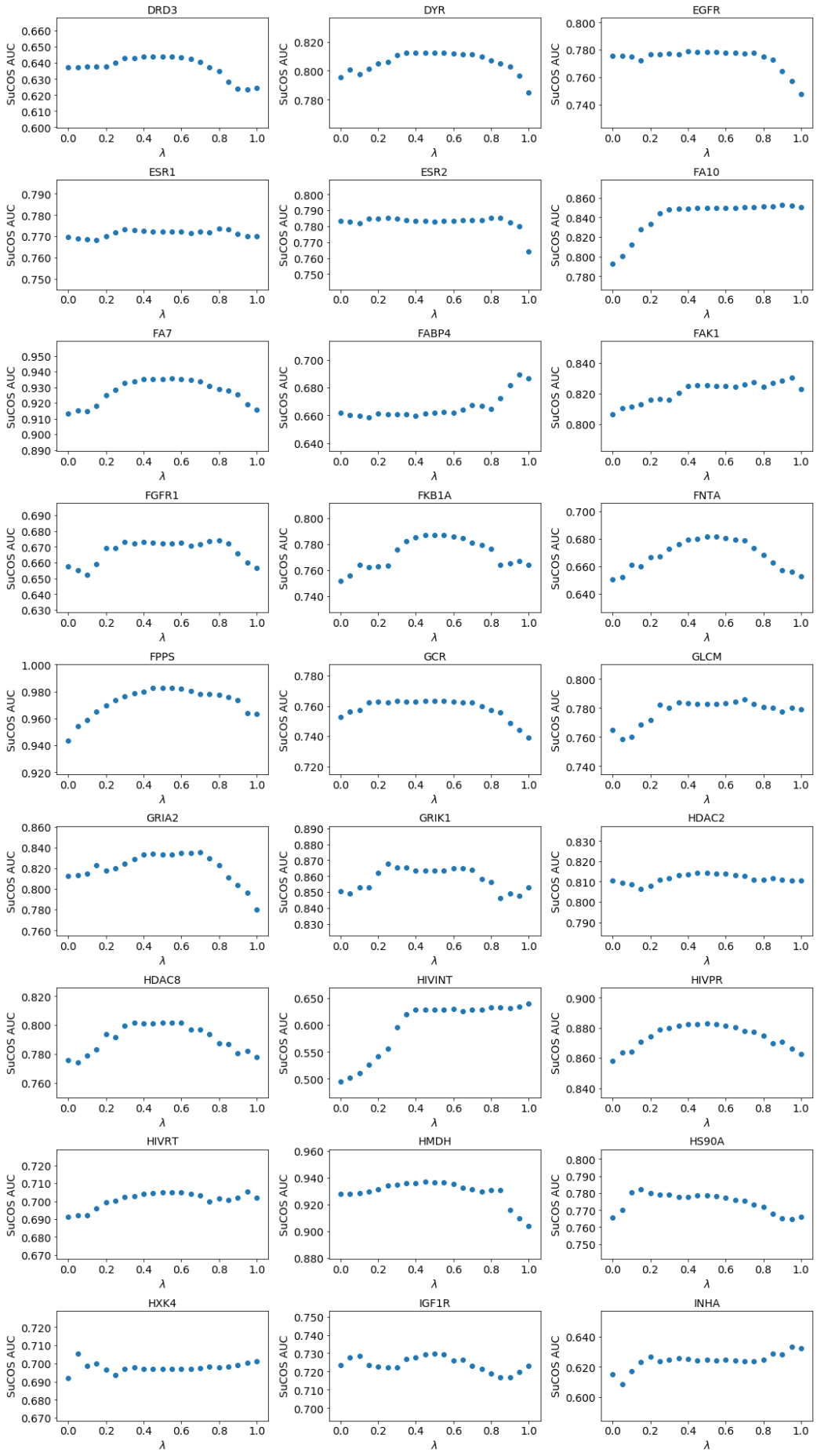
Table B-5. DUD-E ROC enrichment at 2% for AutoDock Vina and SuCOS.

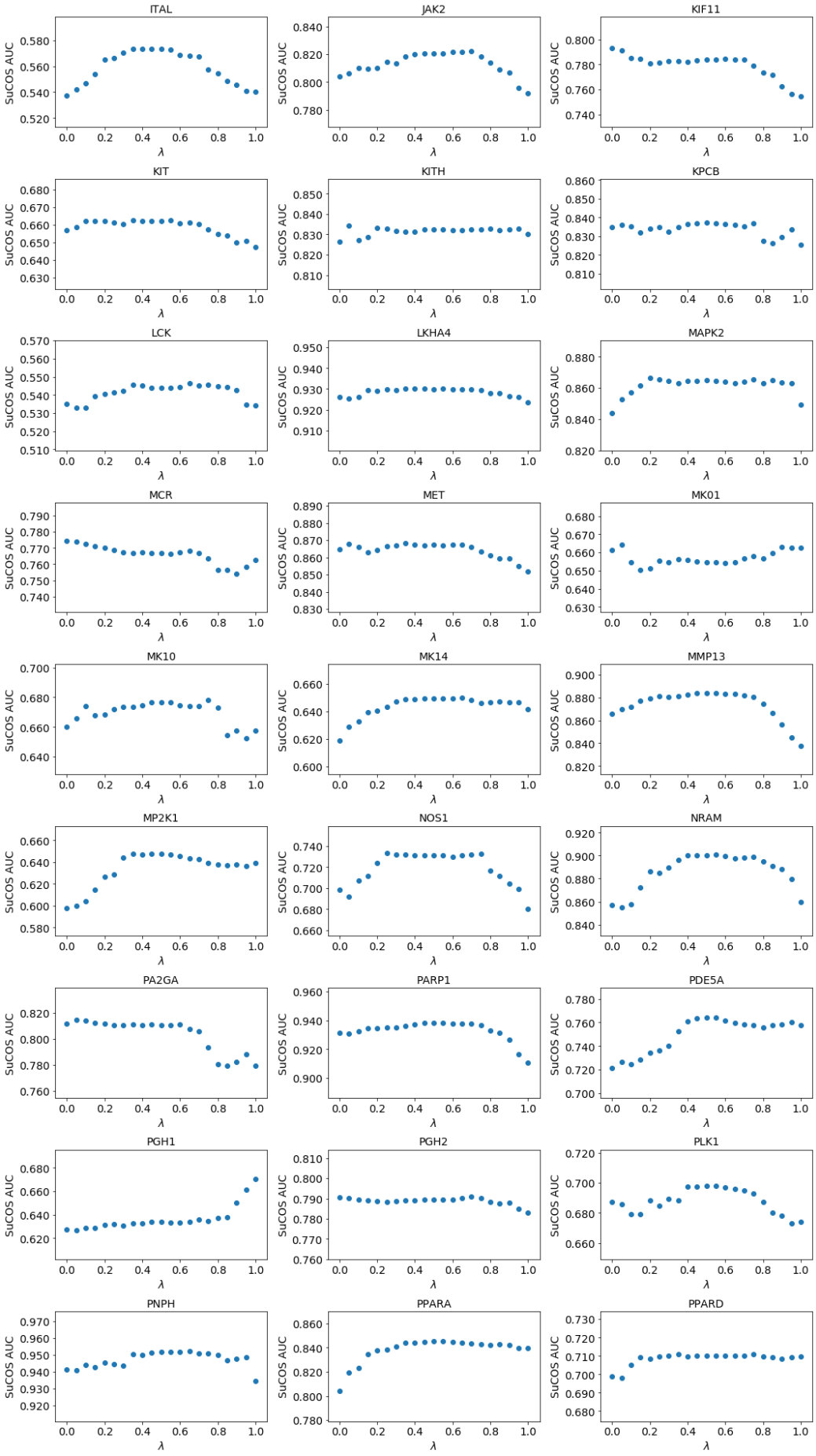
Target	AutoDock Vina	SuCOS
AA2AR	2.905	**8.050**
ABL1	**8.066**	3.204
ACE	1.143	**1.643**
ACES	**6.843**	1.987
ADA	0.860	**11.398**
ADA17	7.406	**15.075**
ADRB1	3.563	**9.231**
ADRB2	3.550	**7.359**
AKT1	**5.051**	3.413
AKT2	**7.863**	4.957
ALDR	5.157	**5.660**
AMPC	1.667	**7.500**
ANDR	6.097	**9.591**
AOFB	**5.738**	1.475
BACE1	3.322	**7.350**
BRAF	**8.158**	6.711
CAH2	0.691	**9.024**
CASP3	2.727	**3.636**
CDK2	4.693	**7.442**
COMT	1.951	**19.512**
CP2C9	**3.000**	2.500
CP3A4	2.395	**2.515**
CSF1R	2.651	**5.060**
CXCR4	0.500	**14.500**
DEF	6.471	**17.451**
DHI1	3.636	**4.606**
DPP4	1.013	**5.779**
DRD3	**3.808**	1.548
DYR	4.242	**10.130**
EGFR	3.247	**8.524**
ESR1	7.572	**9.452**
ESR2	8.229	**10.463**
FA10	8.045	**9.385**
FA7	12.281	**15.965**
FABP4	6.383	**6.809**
FAK1	6.200	**7.600**
FGFR1	4.317	**5.324**
FKB1A	3.784	**4.685**
FNTA	**2.399**	2.365
FPPS	0.000	**18.588**
GCR	5.736	**8.605**
GLCM	0.000	**10.741**
GRIA2	5.823	**11.772**
GRIK1	3.200	**11.000**
HDAC2	**8.757**	7.351
HDAC8	9.294	**10.471**
HIVINT	**2.800**	1.400
HIVPR	4.179	**11.455**
HIVRT	2.967	**3.501**
HMDH	3.647	**15.412**
HS90A	0.000	**11.136**

Target	AutoDock Vina	SuCOS
HXK4	1.522	**7.609**
IGF1R	**6.892**	5.676
INHA	**4.651**	1.860
ITAL	0.876	**2.482**
JAK2	6.355	**8.785**
KIF11	**11.724**	8.103
KIT	**3.494**	1.687
KITH	9.474	**13.333**
KPCB	9.630	**11.852**
LCK	**5.286**	2.619
LKHA4	11.059	**14.353**
MAPK2	10.297	**10.693**
MCR	2.553	**7.872**
MET	7.108	**11.325**
MK01	**8.354**	6.582
MK10	**4.423**	1.154
MK14	**4.256**	3.183
MMP13	3.671	**13.217**
MP2K1	0.333	**2.667**
NOS1	2.000	**6.000**
NRAM	0.204	**12.245**
PA2GA	0.612	**3.061**
PARP1	7.480	**14.882**
PDE5A	5.075	**5.729**
PGH1	**3.282**	1.538
PGH2	9.793	**10.437**
PLK1	1.887	**5.283**
PNPH	7.843	**15.686**
PPARA	7.346	**8.418**
PPARD	2.750	**4.167**
PPARG	4.793	**9.339**
PRGR	**6.075**	2.799
PTN1	**9.692**	6.308
PUR2	10.000	**18.800**
PYGM	**1.558**	0.000
PYRD	8.108	**10.090**
RENI	3.689	**5.243**
ROCK1	2.600	**6.600**
RXRA	**10.076**	8.397
SAHH	10.159	**18.413**
SRC	3.053	**4.237**
TGFR1	10.526	**16.241**
THB	11.068	**13.010**
THRB	4.078	**5.727**
TRY1	3.608	**5.657**
TRYB1	**4.054**	3.378
TYSY	**11.376**	8.440
UROK	6.049	**12.099**
VGFR2	**6.944**	4.205
WEE1	16.040	**19.604**
XIAP	4.444	**16.970**

Table B-6. DUD-E ROC enrichment at 5% for AutoDock Vina and SuCOS.


AA2AR
ABL1
ACE
ACES
ADA
ADA17
ADRB1
ADRB2
AKT1
AKT2
ALDR
AMPC
ANDR
AOFB
BACE1
BRAF
CAH2
CASP3
CDK2
COMT
CP2C9
CP3A4
CSF1R
CXCR4
DEF
DHI1
DPP4
SuCOS AUC
λ


DRD3
DYR
EGFR
ESR1
ESR2
FA10
FA7
FABP4
FAK1
FGFR1
FKB1A
FNTA
FPPS
GCR
GLCM
GRIA2
GRIK1
HDAC2
HDAC8
HIVINT
HIVPR
HIVRT
HMDH
HS90A
HXK4
IGF1R
INHA
SuCOS AUC
λ


ITAL
JAK2
KIF11
KIT
KITH
KPCB
LCK
LKHA4
MAPK2
MCR
MET
MK01
MK10
MK14
MMP13
MP2K1
NOS1
NRAM
PA2GA
PARP1
PDE5A
PGH1
PGH2
PLK1
PNPH
PPARA
PPARD
SuCOS AUC
λ

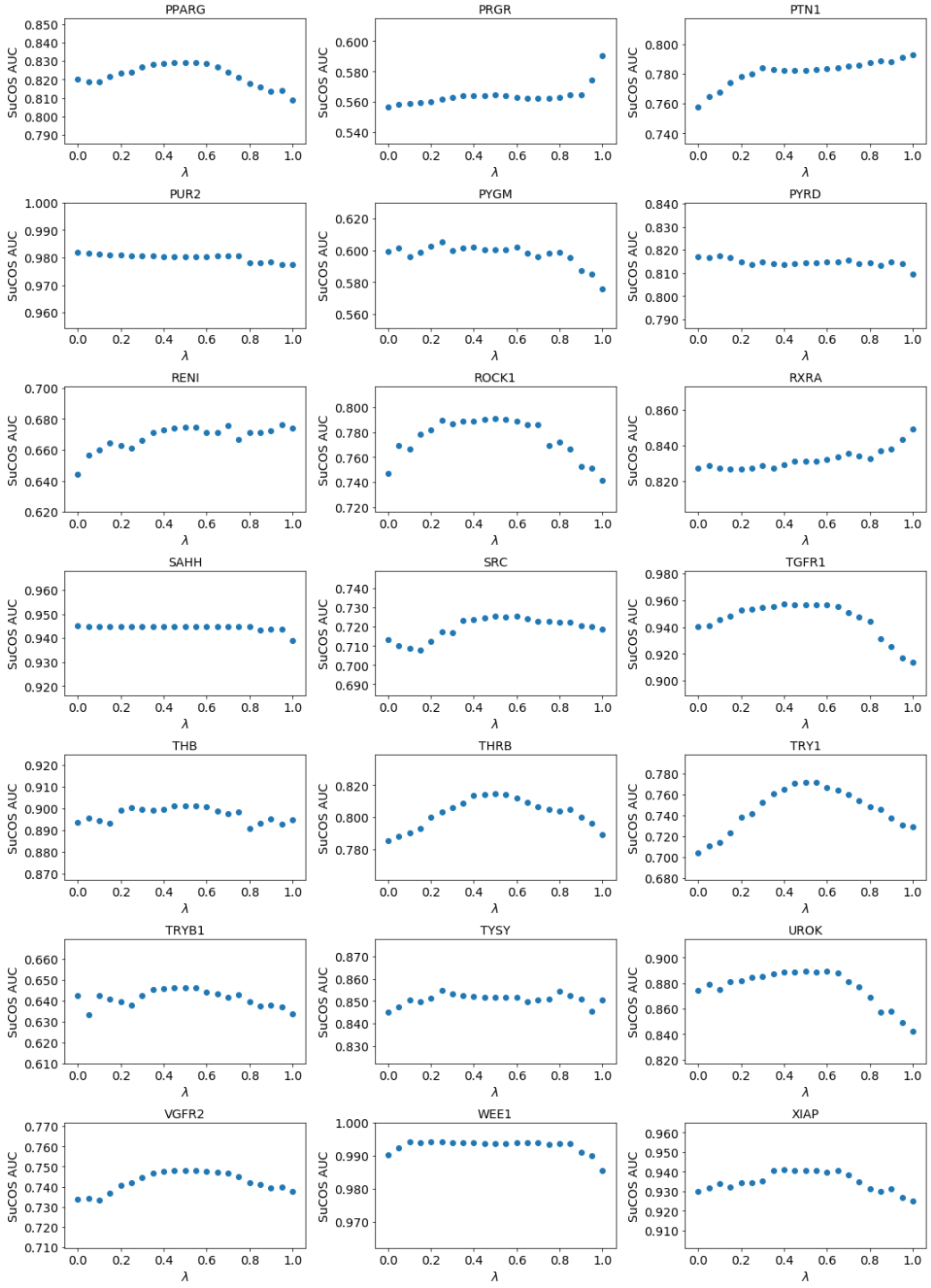


Figure B.1. Investigating how the ROC AUC changes with varying the weight of the components of SuCOS for each DUD-E target.

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
3WB5	0B4	C[C@]1(CC(=O)N(C(=N1)N)C)[C@@H]2C[C@H]2c3ccccc3	0.818	0.762	0.786	1
3WB4	0B3	C[C@]1(CC(=O)N(C(=N1)N)C)CCc2ccccc2	0.805	0.749	0.774	2
2OHM	8AP	c1ccc(cc1)CNc2cccnc2N	0.803	0.726	0.807	3
3HVG	EV0	CCCC1=CC(=O)NC(=N1)N	0.802	0.764	0.788	4
3VV6	B00	CN1C(=O)C=C(N=C1N)[C@H]2C[C@H]2c3ccccc3	0.794	0.757	0.773	5
2OHR	8IP	c1cc(cc(c1)c2cccnc2)CNc3cccnc3N	0.791	0.730	0.767	6
3IGB	454	c1ccc(cc1)C2(C3=NCCCN3C(=N2)N)c4ccccc4	0.773	0.748	0.771	7
4X2L*	3WP	C[C@]1(CCSC(=N1)N)c2ccc(cc2F)F	0.770	0.700	0.775	8
2OF0	CMZ	Cc1ccc(c(c1)OC[C@H](CN2CCOCC2)O)C	0.763	0.717	0.722	9
4L7G	1W0	c1cc(cc(c1)[C@]23CCOC[C@H]2OC(=N3)N)c4cncnc4	0.754	0.690	0.718	10
2OHK	1SQ	c1ccc2c(c1)ccnc2N	0.747	0.641	0.790	11
2ZJH*	F1H	c1ccc(cc1)CN2CCC(CC2)NC(=O)CCCS	0.747	0.716	0.727	12
2OHP	6IP	c1cc(nc(c1)N)CCc2ccc3cc[nH]c3c2	0.747	0.695	0.739	13
4ZSM	4RW	C1CC[C@@H]2[C@H](C1)CSC(=N2)N	0.746	0.711	0.784	14
3BRA	AEF	c1cc(ccc1CCN)O	0.741	0.743	0.639	15
3UDH	91	c1ccc2c(c1)[C@]3(CCNC3)C(=O)N2	0.739	0.682	0.739	16
3KMY	D8Y	c1cc(cc(c1)Cl)CCc2cccnc2N	0.737	0.628	0.764	17
3RSX	RSV	c1cc(nc2c1cc(cc2)c3ccsc3)N	0.736	0.646	0.725	18
2OHN	4FP	c1cc(ccc1CC2CCNCC2)F	0.734	0.703	0.716	19
3L5B	BDO	N=C\1/N[C@](C(=O)N1Cc2cccc(c2)Cl)(C)CC(C)C	0.732	0.726	0.683	20
2OHL	2AQ	c1ccc2c(c1)ccc(n2)N	0.727	0.647	0.754	21
4DJU	0KK	N=C/1\NC(C(=O)N1C)(c2ccccc2)c3ccccc3	0.726	0.730	0.726	22
3HW1	EV2	c1ccc2c(c1)nc(c(n2)N3CCCC3)N	0.725	0.746	0.664	23
2ZJJ*	F1J	c1cc(ccc1C[N@@]2CCN[C@@H](C2)C(=O)NCCS)F	0.724	0.670	0.721	24
3UDJ	92	COC(=O)[C@H]1C[C@@]2(CN1)c3ccccc3NC2=O	0.717	0.684	0.680	25
3BUH	AED	c1cc(c(cc1CCN)C2CCCCC2)O	0.715	0.715	0.606	26
3BUG*	AEH	CCc1cc(ccc1O)CCN	0.709	0.702	0.662	27
3L59	BDJ	N=C\1/NC(C(=O)N1Cc2cccc(c2)Cl)(C)C	0.707	0.638	0.695	28
3KMX	G00	N=C(/N)\SCc1ccc(c(c1)Cl)OCCCC	0.702	0.616	0.718	29
4BEK	XK0	C[C@]1(CCSC(=N1)N)c2ccc(cc2)OC	0.699	0.640	0.764	30
3MSJ*	EV3	c1cc2c(cc1Cl)nc(n2CCCO)N	0.699	0.741	0.660	31
4WY1	3VO	c1cc(c(cc1F)F)[C@]23COCC[C@H]2CSC(=N3)N	0.698	0.645	0.723	32
3BUF	AEG	C[C@H](Cc1ccc(cc1)O)N	0.679	0.696	0.623	33
5MXD	III	CN(C)c1c2ccccc2nc(n1)N3CCCC3	0.596	0.652	0.514	34

* Indicates the representative fragments from $frags_{clustered,t=0.8}$

Table B-7. $Frags_{all}$ for BACE1. Each fragment was used individually to rescore the DUD-E dataset for BACE1 using SuCOS.

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
1OIQ	HDU	Cc1c(n2ccccc2n1)c3ccnc(n3)NC(=O)C	0.811	0.812	0.737	1
1PXM	CK5	Cc1c(sc(n1)C)c2ccnc(n2)Nc3cccc(c3)O	0.805	0.808	0.738	2
3QTW	X3A	c1ccc(cc1)Nc2nc(c(s2)C(=O)c3cccnc3)N	0.801	0.774	0.757	3
3QTR	X36	c1ccc(cc1)C(=O)c2c(nc(s2)Nc3ccccc3)N	0.797	0.760	0.754	4
2VTR	LZB	CC(C)Nc1cc(nc2n1ncc2C#N)Cl	0.796	0.793	0.752	5
1PXK	CK3	Cc1c(sc(n1)C)c2ccnc(n2)N\C=N/O	0.795	0.766	0.737	6
2VTL	LZ5	c1ccc(cc1)NC(=O)c2cc[nH]n2	0.788	0.794	0.753	7
3S0O	50Z	C=CCNc1nc(c(s1)C(=O)c2ccccn2)N	0.788	0.765	0.739	8
1W8C	N69	CC(C)c1[nH]c2c(n1)c(nc(n2)N)OCC3CCCCC3	0.788	0.766	0.725	9
2VTN	LZ7	CC(=O)Nc1c[nH]nc1C(=O)Nc2ccc(cc2)F	0.787	0.787	0.747	10
3RZB	02Z	c1ccc(cc1)Nc2nc(c(s2)C(=O)N)N	0.786	0.756	0.749	11
5AND	5JE	c1ccc2c(c1)[nH]c(n2)n3ccnc3	0.784	0.769	0.745	12
3RK9	09Z	CC(C)Nc1nc(c(s1)C(=O)c2ccccnc2)N	0.783	0.754	0.759	13
3S00	Z60	C=CCNc1nc(c(s1)C(=O)c2ccc(s2)Cl)N	0.779	0.743	0.729	14
3R8Z	Z63	C=CCNc1nc(c(s1)C(=O)c2ccncc2)N	0.777	0.743	0.753	15
3R7E	X88	c1cc(c(cc1[N+](=O)[O-])C(=O)N)NCc2cncn2	0.772	0.715	0.739	16
2R3H	SCE	Cc1cnc2n1ccnc2NCc3ccncc3	0.771	0.809	0.718	17
3R7V	Z02	c1ccnc(c1)CNc2ccc(cc2C(=O)N)[N+](=O)[O-]	0.771	0.721	0.732	18
3QTQ	X35	C=CCNc1nc(c(s1)C(=O)c2ccccnc2)N	0.770	0.732	0.746	19
3QQF	X07	c1cc(cnc1)CNc2ccc(cc2C(=O)N)[N+](=O)[O-]	0.770	0.710	0.744	20
1DI8	DTQ	COc1cc2c(cc1OC)ncnc2Nc3cccc(c3)O	0.767	0.703	0.745	21
3QQK	X02	C=CCNc1nc(c(s1)C(=O)c2ccccc2)N	0.766	0.706	0.749	22
1VYZ	N5B	c1ccc(cc1)C(=O)Nc2cc(n[nH]2)C3CC3	0.763	0.772	0.731	23
3RM7	19Z	c1cc(ccc1CNc2ccc(cc2C(=O)N)[N+](=O)[O-])O	0.762	0.717	0.738	24
2C4G	514	CC(=O)[N+]1=CC\2=C(C1)N=N/C2=N/C(=O)c3ccccc3	0.760	0.732	0.746	25
3R1Y	X76	c1ccc(c(c1)C(=O)N)NCc2cccnc2	0.760	0.710	0.732	26
1W0X	OLO	Cn1cnc2c1nc(nc2NCc3ccccc3)NCCO	0.759	0.759	0.720	27
3TIZ	3TI	c1ccc2c(c1)ccc(c2/C=N/c3ccc(cc3)O)O	0.755	0.690	0.733	28
6Q4E	HH5	c1[nH]c2c(n1)nc(nc2I)N	0.755	0.802	0.664	29
1JSV	U55	c1cc(ccc1Nc2cc(ncn2)N)S(=O)(=O)N	0.754	0.762	0.721	30
3QZF	X66	c1cc(c(cc1O)c2nc(nc(n2)N)N)O	0.748	0.712	0.720	31
3QQJ	X11	c1ccc(c(c1)c2nc(nc(n2)N)N)O	0.748	0.724	0.718	32
3QL8	X01	c1cc(c(cc1C#N)c2ncnc(n2)N)O	0.748	0.701	0.739	33
4EK4	1CK	Cc1cc(n[nH]1)NC(=O)c2ccc(cc2)Br	0.746	0.776	0.711	34
3R8M*	Z19	c1ccc2c(c1)c(n[nH]2)C(=O)NN	0.746	0.714	0.730	35
3QZG	X67	c1cc(c(cc1F)c2nc(nc(n2)N)N)O	0.744	0.722	0.712	36
2BTS	U32	CC(C)c1cnc(s1)Nc2ccc(cc2)S(=O)(=O)N	0.743	0.744	0.709	37
3PXZ	JWS	COc1ccc(c(c1)c2nc(nc(n2)N)N)O	0.743	0.693	0.737	38
4D1Z	WG8	c1cc2ccc(nc2c(c1)O)C3=N[C@H](CS3)C(=O)O	0.741	0.677	0.726	39
6Q4F	26D	c1cc(nc(c1)N)N	0.741	0.767	0.677	40
3QRU	X19	CCC(C)(C)[C@H]1CCc2c(c([nH]n2)C(=O)NC)C1	0.740	0.677	0.723	41
4EZ3	0S0	c1cc(ccc1/N=N/C2=CC=C(NC2=O)O)S(=O)(=O)N	0.739	0.744	0.701	42
2BTR	U73	CC(C)C1=CN/C(=N/C(=O)Cc2cccnc2)/S1	0.739	0.683	0.717	43

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
5ANI	ES4	c1[nH]c2c(n1)c(ncn2)Cl	0.737	0.779	0.678	44
3QQG	X06	c1cc(c(cc1Cl)c2nc(nc(n2)N)N)O	0.737	0.699	0.717	45
4EK5	03K	c1cc(ccc1C(=O)N)C(=O)Nc2cc([nH]n2)C3CC3	0.734	0.754	0.704	46
4FKG	4CK	c1cc(ccc1C(=O)Nc2cc([nH]n2)C3CC3)C(=O)O	0.734	0.750	0.704	47
3R8L	Z30	CC(C)(C)[C@@H]1CCc2c(c(n[nH]2)C(=O)NN)C1	0.733	0.665	0.729	48
1E1X	NW1	C1CCC(CC1)COc2c(c(nc(n2)N)N)N=O	0.728	0.806	0.643	49
2CLX	F18	c1cc(ccc1/N=N\c2c([nH]nc2N)N)O	0.728	0.688	0.708	50
5OO1*	9Z2	CCC(=O)Nc1ccc(cc1)c2ncccn2	0.728	0.667	0.705	51
3R8P	Z46	CCC[C@@H]1CCc2c(c(n[nH]2)C(=O)NN)C1	0.726	0.663	0.719	52
5ANE	SZL	COc1c2c([nH]cn2)ncn1	0.725	0.765	0.658	53
2VTJ	LZ4	c1cc(ccc1Nc2cncc(n2)Cl)S(=O)(=O)N	0.724	0.723	0.693	54
1JVP	LIG	c1ccc-2c(c1)Cc3c2n[nH]c3c4ccncc4	0.723	0.627	0.748	55
5ANK	RJI	c1ccc(cc1)N2C(=O)[C@H](C(=O)NC2=O)C(=O)N	0.720	0.738	0.667	56
1WCC	CIG	c1c(nc(cn1)Cl)N	0.719	0.741	0.669	57
1H0V	UN4	c1[nH]c2c(n1)c(nc(n2)N)OC[C@H]3CCC(=O)N3	0.716	0.773	0.638	58
4FKL	CK2	Cc1c(sc(n1)C)c2ccnc(n2)N	0.712	0.765	0.562	59
1H0W	207	c1c2c(c(nc1N)OCC3CCCCC3)nc[nH]2	0.709	0.768	0.638	60
1E1V	CMG	c1[nH]c2c(n1)c(nc(n2)N)OCC3CCCCC3	0.708	0.776	0.624	61
3QQL	X03	CCC(C)(C)[C@@H]1CCc2c(c(n[nH]2)C(=O)NN)C1	0.707	0.640	0.721	62
6Q3C	BYZ	c1c(cn[nH]1)Br	0.697	0.696	0.661	63
6Q3B	PYZ	c1c(cn[nH]1)I	0.696	0.733	0.640	64
6Q4B	HHN	c1c(cncn1)Br	0.692	0.729	0.650	65
3BHT	MFR	COc1ccnc2c1c(c[nH]2)c3ccnc(n3)N	0.691	0.657	0.687	66
6Q4A	HGW	c1c(cncn1)I	0.691	0.722	0.654	67
1GZ8	MBP	CC(C)C(=O)COc1c2c(nc[nH]2)nc(n1)N	0.690	0.756	0.599	68
2VTA	LZ1	c1ccc2c(c1)cn[nH]2	0.679	0.668	0.661	69
2V0D	C53	N=C\1/NC(=O)/C(=C\c2ccccn2)/S1	0.679	0.716	0.572	70
6Q48*	HHQ	C1C(=CC=NC1=O)I	0.678	0.654	0.726	71
5ANJ	ZXC	c1cc(sc1)C(=O)Nc2c3c([nH]cn3)ncn2	0.677	0.742	0.585	72
2EXM	ZIP	CC(=CCNc1c2c([nH]cn2)ncn1)C	0.674	0.740	0.600	73
3TIY*	TIY	c1c2c(c(c(c1O)O)O)C(=O)C(=CC=C2)O	0.674	0.646	0.653	74
5OO3	9ZB	CCC(=O)N1CCN(c2c1cccc2)CC	0.668	0.680	0.646	75
1PXI	CK1	c1cnc(nc1c2cc(sc2Cl)Cl)N	0.662	0.673	0.544	76
5OSM	AEQ	CCC(=O)N1CCCc2c1ccc(c2)C(=O)OC	0.656	0.630	0.675	77
6Q4C	HH8	c1cc2c(ccnc2nc1)Br	0.645	0.639	0.625	78
3BHV	VAR	c1cnc(n2c1c(c3c2nccc3O)c4ccnc(n4)N)N	0.641	0.641	0.629	79
3FZ1	B98	COc1ccc2c(c1)c3c(s2)C(=O)N[C@@H](CN3)CN	0.625	0.611	0.604	80
2VTH	LZ2	c1cc2c(cccc2S(=O)(=O)N)c(c1)O	0.624	0.606	0.620	81
6Q4G	HJK	c1cc(cc(c1)c2c3c([nH]cn3)nc(n2)N)CC(=O)O	0.615	0.646	0.536	82
6Q4J	HHB	c1cc(cc(c1)Nc2ccncn2)CC(=O)O	0.608	0.665	0.520	83
5FP5	1Y6	c1cc(ccc1C(=O)O)F	0.606	0.590	0.614	84
6Q4I*	HGK	c1cc(ccc1CC(=O)O)OC2=CC=NC(=O)C2	0.603	0.616	0.572	85
6Q4H	HGH	c1cc(cc(c1)Oc2c3c([nH]cn3)nc(n2)N)CC(=O)O	0.599	0.652	0.521	86

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
5FP6	MFZ	c1cc(c2c(c1Cl)c(c[nH]2)C#CCO)Cl	0.578	0.552	0.625	87
5ANG	WY3	c1cc2c(cc1O)OC(=O)C=C2CN3CCOCC3	0.578	0.597	0.545	88
4D1X	ESJ	c1cc2c(cc1O)sc(n2)C3=N[C@H](CS3)C(=O)O	0.577	0.652	0.520	89
6Q4D	HHT	COc1cc(ccc1CC(=O)O)Br	0.498	0.575	0.395	90

* Indicates the representative fragments from $frags_{clustered,t=0.8}$

Table B-8. $Frags_{all}$ for CDK2. Each fragment was used individually to rescore the DUD-E dataset for CDK2 using SuCOS.

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
5EH5	XCZ	Cc1ccnc(n1)S(=O)(=O)N	0.920	0.930	0.866	1
2WEG	FBV	c1ccc(c(c1)F)S(=O)(=O)N	0.915	0.915	0.869	2
3S75	EVG	c1cc(oc1)S(=O)(=O)N	0.910	0.923	0.872	3
2HNC	1SA	c1(nnc(s1)S(=O)(=O)N)N	0.909	0.914	0.888	4
5O07	1VQ	Cc1ncc(n1CCOS(=O)(=O)N)[N+](=O)[O-]	0.907	0.925	0.617	5
5JGT	EVI	c1csc(n1)S(=O)(=O)N	0.906	0.926	0.842	6
2WEO	FBW	c1cc(cc(c1)S(=O)(=O)N)F	0.906	0.901	0.869	7
4YX4	FB2	c1ccc(cc1)S(=O)(=O)N	0.906	0.899	0.880	8
3KIG	DA4	C#Cc1cccc(c1)S(=O)(=O)N	0.905	0.894	0.855	9
4YVY*	4J3	NOS(=O)(=O)N	0.905	0.923	0.837	10
1ZFQ	ZEC	c1cc2c(cc1O)sc(n2)S(=O)(=O)N	0.905	0.917	0.836	11
3DD0	EZL	CCOc1ccc2c(c1)sc(n2)S(=O)(=O)N	0.904	0.923	0.766	12
3S71	EVD	c1ccc2c(c1)cc(o2)S(=O)(=O)N	0.904	0.916	0.848	13
1IF4	FBS	c1cc(ccc1F)S(=O)(=O)N	0.903	0.897	0.877	14
1IF5	FBT	c1cc(c(c(c1)F)S(=O)(=O)N)F	0.903	0.903	0.828	15
4YXI*	4J8	Cc1ccc(cc1)S(=O)(=O)N	0.902	0.890	0.882	16
2QOA	MAJ	c1cc2c(cc1S(=O)(=O)N)CCC2	0.901	0.886	0.868	17
3S76	EVH	c1cnc([nH]1)S(=O)(=O)N	0.901	0.907	0.879	18
1KWQ	SG1	c1cc(c(cc1S(=O)(=O)N)[N+](=O)[O-])N2CCCC2=O	0.901	0.906	0.696	19
5JGS	EVF	c1ccc2c(c1)nc(s2)S(=O)(=O)N	0.900	0.926	0.807	20
4RUZ	3W8	CCOc1ccc(cc1)S(=O)(=O)N	0.899	0.898	0.838	21
4KAP	1QV	c12c(c(c(c(c1sc(n2)S(=O)(=O)N)F)F)F)F	0.899	0.910	0.830	22
5NEA	8V8	Cc1ncc(o1)c2ccc(cc2)S(=O)(=O)N	0.898	0.894	0.806	23
3S72	EVE	c1ccc2c(c1)[nH]c(n2)S(=O)(=O)N	0.897	0.906	0.855	24
1ZGE	SDA	c1c(cc(c(c1Cl)N)Cl)S(=O)(=O)N	0.896	0.893	0.801	25
4YXO	4JC	CCc1ccc(cc1)S(=O)(=O)N	0.896	0.882	0.868	26
5T71	75W	N=N/c1ccc(cc1)S(=O)(=O)N	0.895	0.887	0.876	27
1KWR	SG2	CN1c2ccc(cc2C(=O)S1)S(=O)(=O)N	0.895	0.892	0.779	28
2EU3	FF3	c1(nnc(s1)N)C(F)(F)S(=O)(=O)N	0.895	0.921	0.567	29
4RUY	3W6	CCCOc1ccc(cc1)S(=O)(=O)N	0.895	0.902	0.782	30
4RUX	3W3	C=CCOc1ccc(cc1)S(=O)(=O)N	0.894	0.902	0.780	31
4FU5	0VX	C1CO/C(=N\S(=O)(=O)N)/N1	0.894	0.883	0.884	32
4N0X	EVJ	c1cc(sc1)S(=O)(=O)N	0.892	0.876	0.880	33
4FRC	0VY	C1CCN(C1)/C(=N/S(=O)(=O)N)/N	0.892	0.882	0.851	34
1IF6	FBU	c1c(cc(cc1F)S(=O)(=O)N)F	0.892	0.894	0.787	35
4YXU	4JE	CCCc1ccc(cc1)S(=O)(=O)N	0.891	0.880	0.822	36
4XE1	IL5	c1cc2c(cc1S(=O)(=O)N)S(=O)(=O)NC2=O	0.890	0.883	0.833	37
3M14	BEV	C1CS/C(=N\S(=O)(=O)N)/N1	0.890	0.880	0.872	38
2NNG	ZYX	c1cc(ccc1CCN)S(=O)(=O)N	0.890	0.878	0.846	39
3V5G	0F3	c1cc(ccc1NS(=O)(=O)N)S(=O)(=O)N	0.890	0.892	0.823	40
4YYT	S2O	c1cc(ccc1CCO)S(=O)(=O)N	0.890	0.881	0.845	41
2GEH	NHY	C(=O)(N)NO	0.889	0.898	0.828	42
5WLR	86B	CC(C)NC[C@@H](COc1ccc(cc1)S(=O)(=O)N)O	0.889	0.894	0.728	43

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
3BL0	BL0	CN(C)c1nnc(s1)CS(=O)(=O)N	0.889	0.878	0.705	44
3S74	03T	c1ccc2c(c1)cc(s2)S(=O)(=O)N	0.888	0.875	0.843	45
4Q6D	WW3	c1cc(ccc1/N=N/N2CCCCCC2)S(=O)(=O)N	0.888	0.893	0.697	46
5SZ5	72E	Cc1ccccc1c2ccc(cc2)S(=O)(=O)N	0.886	0.883	0.809	47
3RYV	RYV	CCNC(=O)c1ccc(cc1)S(=O)(=O)N	0.883	0.880	0.810	48
3T5U	A09	c1ccc(cc1)S(=O)(=O)NO	0.881	0.889	0.726	49
5SZ4	72D	c1ccc(cc1)c2ccc(cc2)S(=O)(=O)N	0.881	0.879	0.806	50
5N25	8HK	c1cc(cnc1)c2ccc(cc2)S(=O)(=O)N	0.880	0.886	0.754	51
2NNO	M28	c1cc(ccc1CCC(=O)O)S(=O)(=O)N	0.880	0.869	0.798	52
3MZC	S6I	c1cc(ccc1NC(=O)NC2CCCC2)S(=O)(=O)N	0.879	0.898	0.670	53
3IGP	DT7	COc1cc2c(cc1OC)C[N@@](CC2)S(=O)(=O)N	0.878	0.879	0.673	54
5N24	8HE	c1cc(ccc1C#N)c2ccc(cc2)S(=O)(=O)N	0.877	0.892	0.706	55
5LL8	6YP	CCCCc1ccc(cc1)S(=O)(=O)N	0.877	0.879	0.739	56
6CEH	EZ1	C=CC[Se]c1ccc(cc1)S(=O)(=O)N	0.876	0.880	0.725	57
5LLG	VD9	CCCSc1ccc(cc1)S(=O)(=O)N	0.876	0.879	0.775	58
3RYY	RYY	CCCNC(=O)c1ccc(cc1)S(=O)(=O)N	0.875	0.876	0.776	59
1CIM	PTS	C[C@H]1C[C@@H](c2cc(sc2S1(=O)=O)S(=O)(=O)N)N	0.874	0.867	0.756	60
4FVN	0VW	C1CNC(=NS(=O)(=O)N)NC1	0.874	0.858	0.858	61
6H29	FK8	c1ccc(cc1)COC(=O)N	0.874	0.891	0.689	62
5E28	BC5	c1cc(ccc1c2ccc(cc2)S(=O)(=O)N)N	0.874	0.889	0.717	63
5SZ6	72G	c1cc(cc(c1)c2ccc(cc2)S(=O)(=O)N)C=O	0.874	0.877	0.776	64
3RYJ	RYJ	c1cc(ccc1C(=O)NCC(F)(F)F)S(=O)(=O)N	0.873	0.880	0.746	65
4FPT	0VZ	CCOC(=O)[C@@H]1CS/C(=N\S(=O)(=O)N)/N1	0.873	0.880	0.719	66
2WEH	FB1	c1ccc(c(c1)S(=O)(=O)N)Cl	0.873	0.872	0.703	67
2NNS	M25	CC(=O)NCCc1ccc(cc1)S(=O)(=O)N	0.870	0.879	0.702	68
4RFD	3O5	c1cc(ccc1OCCCCN)S(=O)(=O)N	0.870	0.897	0.694	69
5TXY	7Q1	c1ccc(cc1)[C@@H]2C(=O)NC(=O)O2	0.869	0.832	0.729	70
2O4Z	HSW	NS(=O)(=O)NO	0.869	0.869	0.832	71
5E2K	BX4	c1cc(cc(c1)N)c2ccc(cc2)S(=O)(=O)N	0.869	0.888	0.730	72
4CQ0	SXS	c1cc2c(cc1N)S(=O)(=O)NC2=O	0.868	0.869	0.688	73
2NNV	M29	CCOC(=O)CCc1ccc(cc1)S(=O)(=O)N	0.867	0.877	0.670	74
5BYI	4WA	c1cc(ccc1N)N=Nc2ccc(cc2)S(=O)(=O)N	0.867	0.893	0.685	75
6D1L	FQV	CC#CC[Se]c1ccc(cc1)S(=O)(=O)N	0.865	0.879	0.679	76
1BCD	FMS	C(F)(F)(F)S(=O)(=O)N	0.865	0.863	0.660	77
3OYQ	OYQ	CC[C@H](C)CCC(=O)Nc1ccc(cc1)S(=O)(=O)N	0.864	0.896	0.658	78
1AM6	HAE	CC(=O)NO	0.863	0.858	0.783	79
5TY9	7QV	COc1ccc(c(c1)OC)[C@@H]2C(=O)NC(=O)O2	0.862	0.856	0.647	80
2QP6	MB1	c1cc(c(cc1[N+](=O)[O-])S(=O)(=O)N)Cl	0.862	0.872	0.582	81
4RIV	LSA	c1ccc2c(c1)C(=O)NS2(=O)=O	0.861	0.860	0.687	82
5EKH	5RD	CCCCNCc1ccc(cc1)S(=O)(=O)N	0.861	0.876	0.672	83
3RZ5	RZ5	CCCCCNC(=O)c1ccc(cc1)S(=O)(=O)N	0.861	0.871	0.717	84
1OKL	MNS	CN(C)c1cccc2c1cccc2S(=O)(=O)N	0.860	0.861	0.637	85
3RZ0	RZ0	CCCCNC(=O)c1ccc(cc1)S(=O)(=O)N	0.860	0.872	0.725	86

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
3RZ8	RZ8	CCCCCCNC(=O)c1ccc(cc1)S(=O)(=O)N	0.859	0.869	0.716	87
4FVO	0VV	c1ccc2c(c1)CN/C(=N\S(=O)(=O)N)/N2	0.859	0.846	0.807	88
5N1S	8GE	c1ccc2c(c1)cc([nH]2)c3ccc(cc3)S(=O)(=O)N	0.858	0.878	0.706	89
5E2S	5CX	CC(C)c1ccccc1c2ccc(cc2)S(=O)(=O)N	0.857	0.865	0.630	90
3OYS	OYS	c1ccc(cc1)CC(=O)Nc2ccc(cc2)S(=O)(=O)N	0.857	0.891	0.631	91
5NXO	9HK	c1cc(ccc1Cc2ccc(cc2)S(=O)(=O)N)[N+](=O)[O-]	0.854	0.869	0.668	92
5EKM	TG5	CCCCNCC(=O)Nc1ccc(cc1)S(=O)(=O)N	0.853	0.883	0.688	93
3MNU	BON	B1(OC(C(O1)(C)C)(C)C)c2ccc(cc2)NS(=O)(=O)N	0.853	0.866	0.668	94
4QEF	0NM	C(#N)O	0.853	0.835	0.870	95
1OKM	SAB	c1cc(ccc1C(=O)NCCCCN)S(=O)(=O)N	0.852	0.871	0.672	96
6GOT	F6W	c1ccc(cc1)CCSc2ccc(cc2)S(=O)(=O)N	0.849	0.872	0.659	97
5SZ3	72H	c1ccc2c(c1)cc(cn2)c3ccc(cc3)S(=O)(=O)N	0.849	0.877	0.658	98
5FDI	5WM	c1ccc2c(c1)C(=CS2(=O)=O)CNS(=O)(=O)N	0.849	0.861	0.665	99
4PQ7	IL3	CC#CCOc1ccc(cc1)CNS(=O)(=O)N	0.840	0.878	0.648	100
5FDC	5WN	c1ccc2c(c1)c(cs2)CNS(=O)(=O)N	0.838	0.814	0.710	101
5THI	0TR	C1=CC=C(C(=O)C=C1)O	0.834	0.824	0.730	102
3L14	I7B	c1c(c(cc(c1Cl)S(=O)(=O)N)S(=O)(=O)N)N	0.832	0.849	0.577	103
3IBI	BOW	CCCCCCCCOS(=O)(=O)N	0.831	0.863	0.656	104
3PO6	RDT	C[C@@H]1c2cc(c(cc2CC[N@]1S(=O)(=O)N)OC)OC	0.826	0.895	0.530	105
3HKQ	1SD	C([C@@H]1[C@@H]([C@@H]([C@H]([C@@H](O1)S(=O)(=O)N)O)O)O)O	0.823	0.866	0.587	106
4MO8	2VQ	Cc1ncc(n1CCNS(=O)(=O)N)[N+](=O)[O-]	0.815	0.875	0.536	107
4FL7	BHO	c1ccc(cc1)C(=O)NO	0.812	0.850	0.607	108
5TYA	7QS	c1ccc(cc1)[C@@H]2C(=O)NC(=O)S2	0.809	0.755	0.688	109
5FNH	YIP	c1cc(cc(c1)Cl)OCc2n[n-]nn2	0.807	0.802	0.544	110
4Q8Z	4HO	CC1=CC(=O)N(C=C1)O	0.806	0.803	0.695	111
5CLU	S8A	c1ccc2c(c1)C(=O)N(S2(=O)=O)CC(=O)O	0.801	0.788	0.678	112
5WG7	AUD	CC1=CC(=O)NS(=O)(=O)O1	0.797	0.776	0.656	113
5FLQ	IO2	c1ccc(cc1)COc2ccc(cc2)CC(=O)O	0.794	0.825	0.670	114
5FNG	YIE	c1cc(ccc1Cc2[n-]nnn2)Cl	0.793	0.811	0.608	115
1CRA	TRI	c1[nH]cnn1	0.791	0.664	0.770	116
2EU2	5DS	CN(C)c1nnc(s1)[C@H](N)S(=O)(=O)N	0.791	0.857	0.529	117
5EH7	5O5	c1cc(c(cc1OCc2[nH]nnn2)Cl)Cl	0.788	0.834	0.574	118
3T5Z	B09	CONS(=O)(=O)c1ccccc1	0.786	0.800	0.554	119
3IBU	O48	CCCCCCCCCCOS(=O)(=O)N	0.785	0.845	0.632	120
4BCW	TU0	c1cc(c(cc1Br)/C=C/S(=O)(=O)O)O	0.782	0.767	0.656	121
5W8B*	A9J	FP(F)(F)(F)(F)F	0.781	0.500	0.781	122
5FLO	J4K	c1cc(ccc1OCc2[nH]nnn2)Cl	0.774	0.819	0.580	123
5U0D	7R7	COc1ccc(c(c1)OC)[C@@H]2C(=O)NC(=S)O2	0.768	0.749	0.579	124
3M04	BE9	c1ccc(cc1)O/C(=N\S(=O)(=O)N)/N2CCOCC2	0.767	0.788	0.574	125
3P5A	IT2	C1COCCN1C(=S)S	0.749	0.737	0.624	126
5FNI	YIH	c1cc(c(cc1OCc2n[n-]nn2)Cl)Cl	0.743	0.781	0.571	127
4WL4	FC5	c1cc2c(cc1O)C=CC(=S)O2	0.741	0.729	0.661	128
4FIK	TH7	c1cc(c(cc1O)O)SO	0.732	0.747	0.594	129

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
5EHW	5OO	c1cc(c(cc1Cl)Cl)/C=C/C(=O)O	0.726	0.746	0.616	130
5FLP	6J5	c1ccc(c(c1)OCc2[nH]nnn2)Cl	0.719	0.672	0.687	131
2OSM	HTS	c1ccc(c(c1)O)S	0.718	0.734	0.574	132
4MLX	TM7	CC1=CC(=S)C(=CO1)O	0.718	0.706	0.640	133
4Q8X	7FH	C1=CC(=S)N(C=C1C(F)(F)F)O	0.717	0.673	0.677	134
2OSF	S24	c1cc(c(cc1O)O)SC(=O)O	0.717	0.723	0.600	135
4E49	RCO	c1cc(cc(c1)O)O	0.703	0.721	0.660	136
4Q99	HT4	Cc1ccc(c(c1)S)O	0.700	0.698	0.580	137
5FNJ	YI6	CCOc1ccc(cc1)CC(=O)O	0.689	0.701	0.571	138
5THN	7CZ	C1=CC=C(C(=S)C=C1)O	0.683	0.695	0.564	139
4E3H*	HQE	c1cc(ccc1O)O	0.681	0.744	0.601	140
3M1K	BEW	c1cc[n+](c(c1)S)O	0.671	0.706	0.590	141
4Q9Y	M3T	Cc1cccc(c1)S	0.669	0.662	0.569	142
4Q83	3FH	C1=CN(C(=S)C(=C1)C(F)(F)F)O	0.667	0.696	0.510	143
5U0G	7QY	COc1ccc(c(c1)OC)C[C@@H]2C(=O)NC(=O)S2	0.658	0.708	0.503	144
4Q7P	3MH	CC1=CC=CN(C1=S)O	0.657	0.675	0.519	145
5VGY	9AA	COc1ccc(c(c1)OC)C[C@@]2(C(=O)NC(=S)N2)O	0.641	0.677	0.537	146
5U0E	7R4	c1ccc(cc1)C[C@@H]2C(=O)NC(=S)S2	0.640	0.649	0.509	147
4Q7S	2YU	CC1=CC(=S)N(C=C1)O	0.632	0.611	0.584	148
4MLT	TM4	CC1=C(C(=S)C=CO1)O	0.631	0.644	0.526	149
4Q8Y	HQT	c1ccc2c(c1)C=CN(C2=S)O	0.628	0.648	0.498	150
4Q87	4FH	C1=CN(C(=S)C=C1C(F)(F)F)O	0.627	0.616	0.574	151
5U0F	7R1	COc1ccc(c(c1)OC)C[C@@H]2C(=O)NC(=S)S2	0.620	0.669	0.512	152
4Q7V	5MH	CC1=CN(C(=S)C=C1)O	0.619	0.594	0.599	153
6HX5	GXE	c1ccc(cc1)[SeH]	0.606	0.592	0.552	154
4Q81	7MH	CC1=CC(=S)N(C(=C1)C)O	0.598	0.602	0.519	155
5EH8	5O6	C/C(=C\C(=O)O)/c1ccc(cc1)OC	0.591	0.596	0.549	156
3P58	P58	CN(Cc1ccccc1)C(=S)S	0.584	0.700	0.504	157
6MBY	FER	COc1cc(ccc1O)\C=C\C(=O)O	0.579	0.610	0.529	158
5FLS	6ZX	C/C(=C\C(=O)O)/c1ccc(cc1)Cl	0.578	0.582	0.540	159
4Q90	4H2	CC1=CC(=S)NC=C1	0.576	0.542	0.557	160
4HEW	2MZ	Cc1[nH]ccn1	0.575	0.584	0.545	161
4E3G	PHB	c1cc(ccc1C(=O)O)O	0.560	0.558	0.543	162
4E3D	GTQ	c1cc(c(cc1O)C(=O)O)O	0.553	0.571	0.551	163
6MBV	NIO	c1cc(cnc1)C(=O)O	0.552	0.590	0.522	164
5M78*	SAL	c1ccc(c(c1)C(=O)O)O	0.547	0.576	0.565	165
3HFP	MIZ	Cc1cc([n+](c(c1)C)CCc2c[nH]cn2)C	0.540	0.575	0.517	166
5FLT	VJJ	c1ccc(cc1)Oc2cccc(c2)C(=O)O	0.534	0.560	0.535	167
6B4D	53X	CCn1cc(c2c1cccc2)c3cc(n(n3)C)C(=O)O	0.518	0.550	0.525	168
3EFI*	TRP	c1ccc2c(c1)c(c[nH]2)C[C@@H](C(=O)O)N	0.516	0.571	0.512	169
1MOO*	4MZ	Cc1c[nH]cn1	0.512	0.530	0.510	170
3IEO*	AMJ	CC(=O)N[C@@H](COC)C(=O)NCc1ccccc1	0.510	0.533	0.494	171
2FMG	PHE	c1ccc(cc1)C[C@@H](C(=O)O)N	0.509	0.538	0.488	172

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
4E4A	JKE	c1ccc(c(c1)C(=O)O)S	0.507	0.566	0.416	173
4E3F	GRE	c1cc(c(c(c1)O)C(=O)O)O	0.506	0.496	0.560	174
2EZ7	DHI	c1c([nH+]c[nH]1)C[C@H](C(=O)O)N	0.506	0.545	0.476	175
4HEZ	1MZ	Cn1cc[nH+]c1	0.501	0.384	0.652	176
3P5L	IT5	c1ccc(cc1)C2(CCN(CC2)C(=S)S)C#N	0.498	0.504	0.487	177
2FMZ	DPN	c1ccc(cc1)C[C@H](C(=O)O)N	0.496	0.549	0.469	178
1AVN	HSM	c1c(nc[nH]1)CCN	0.490	0.462	0.497	179
3M5T*	BFG	c1ccc(cc1)C(CC(=O)NCCS)c2ccccc2	0.476	0.490	0.470	180
2ABE*	HIS	c1c([nH+]c[nH]1)C[C@@H](C(=O))N	0.460	0.433	0.478	181
4Q7W	6MH	CC1=CC=CC(=S)N1O	0.460	0.443	0.561	182
5BNL*	2HC	c1ccc(c(c1)\C=C\C(=O)O)O	0.435	0.490	0.427	183
3KWA	SPM	C(CCNCCCN)CNCCCN	0.430	0.421	0.450	184
3F8E	TE1	CC(C)C[C@@H](c1cc(c(cc1OC)O)/C=C\C(=O)O)O	0.430	0.494	0.425	185
2HKK	ALE	CNC[C@@H](c1ccc(c(c1)O)O)O	0.405	0.507	0.391	186

* Indicates the representative fragments from $frags_{clustered,t=0.8}$

Table B-9. $Frags_{all}$ for CAH2. Each fragment was used individually to rescore the DUD-E dataset for CAH2 using SuCOS.

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
3RXA	SZ1	C1CCCC(CC1)N	0.843	0.860	0.728	1
3VPK	GHS	N=C(\N)/NCCCCCC(=O)O	0.840	0.865	0.741	2
3A7Z	O14	N=C(/c1ccc(cc1)/C=N/OC2CCN(CC2)C)\N	0.835	0.815	0.755	3
3A7W	O04	N=C(c1ccc(cc1)/C=N/O[C@H]2CCCCO2)\N	0.830	0.815	0.751	4
3A88	H06	N=C(c1ccc(cc1)/C=N/NC(=O)c2cccnc2)\N	0.828	0.807	0.752	5
2FX4	C1R	C1CNCCC1CCCC=O	0.823	0.843	0.706	6
3A7Y	O13	N=C(/c1ccc(cc1)/C=N/O[C@@H]2CCCN(C2)C)\N	0.822	0.812	0.741	7
3RXI	TSS	c1ccc2c(c1)c(c[nH]2)CCN	0.820	0.830	0.707	8
4ABB	K9S	c1ccc2c(c1)c(cs2)CN	0.819	0.826	0.732	9
1UTO	PEA	c1ccc(cc1)CC[NH3+]	0.819	0.840	0.690	10
3RXB	ALG	N=C(\N)/NCCCC(=O)O	0.819	0.799	0.762	11
4ABF	513	c1cc2c(cc1Br)c(cs2)CN	0.817	0.821	0.734	12
3V0X*	ANH	COC(=O)[C@H](Cc1cccc(c1)C(N)=N)NC(=O)CNS(=O)(=O)c1ccc(C)cc1	0.816	0.793	0.759	13
5MNG	BAM	c1ccc(cc1)C(=[NH2+])N	0.816	0.790	0.804	14
3A80	O15	N=C(/c1ccc(cc1)/C=N/OC(C)(C)C(=O)O)\N	0.814	0.815	0.738	15
3A7V	3FZ	N=C(c1cccc(c1)C=O)/N	0.814	0.801	0.788	16
1TX7	4CM	C[P@](=O)(c1ccc(cc1)C(=N)N)O	0.813	0.813	0.745	17
1TX8	AM4	CS(=O)(=O)c1ccc(cc1)C(=N)N	0.811	0.807	0.765	18
3A8A	4FZ	N=C(c1ccc(cc1)C=O)/N	0.809	0.803	0.770	19
1C5S	ESX	c1ccc2c(c1)cc(s2)C(=[NH2+])N	0.809	0.808	0.770	20
2G5V	22M	N=C(c1ccc2cc([nH]c2c1)c3ccccc3C)/N	0.808	0.794	0.763	21
5MN1	2AP	c1cc[nH+]c(c1)N	0.808	0.815	0.757	22
2AH4	GBS	c1cc(ccc1C(=O))NC(=N)N	0.807	0.816	0.750	23
1UTP	PBN	c1ccc(cc1)CCCCN	0.803	0.826	0.693	24
3NKK	JLZ	N=C(c1ccc(c(c1)F)C)\N	0.803	0.802	0.758	25
4I8H	BEN	N=C(\c1ccccc1)/N	0.800	0.778	0.801	26
1G3C	109	C[C@@H](C(=O)O)NCc1ccc(cc1O)C(=N)N	0.799	0.795	0.730	27
1TPP	APA	N=C(c1ccc(cc1)C[C@@H](C(=O)O)O)\N	0.798	0.828	0.697	28
3GY4*	PBZ	c1cc(ccc1C(=[NH2+])N)N	0.798	0.767	0.796	29
3RXK	SZ8	CN1C[C@@H](C[C@H]1C(=O)OC)N	0.797	0.790	0.723	30
1C5U	ESP	c1cc2cc(sc2nc1)C(=[NH2+])N	0.793	0.786	0.783	31
1GI6	124	c1ccc(c(c1)c2cc3cc(ccc3[nH]2)C(=[NH2+])N)O	0.793	0.797	0.728	32
1C2K	ABI	c1cc2c(cc1C(=[NH2+])N)[nH]cn2	0.793	0.772	0.789	33
4ABA	SW1	c1cc(sc1)c2nc(cs2)CN	0.791	0.781	0.766	34
1TNG	AMC	C1CCC(CC1)C[NH3+]	0.789	0.770	0.764	35
1GHZ	120	c1cc2c(cc1C(=[NH2+])N)nc([nH]2)C3=CC=CNC3=O	0.786	0.770	0.740	36
1O35	802	c1cc2c(cc1C(=[NH2+])N)nc([nH]2)c3cc(ccc3[O-])F	0.784	0.798	0.724	37
1G3B	108	C[C@@H](C(=O)O)NCc1cc(ccc1O)C(=N)N	0.783	0.758	0.753	38
1O33	801	c1cc(c(nc1)c2[nH]c3ccc(cc3n2)C(=[NH2+])N)[O-]	0.781	0.767	0.733	39
3RXL	SZ9	Cc1cc(c(o1)C)CN	0.780	0.768	0.759	40
3RXG	SZ3	C1CC(CCC1N)O	0.779	0.748	0.725	41
3GY2	BRN	c1cc(ccc1C(=N)N)N/N=N/c2ccc(cc2)C(=N)N	0.779	0.767	0.725	42
1BJU	GP6	N=C(c1ccc(cc1)NC(=O)Nc2ccc(cc2)Cl)/N	0.778	0.758	0.724	43

PDB ID	Ligand ID	Ligand SMILES	SuCOS AUC	Feature AUC	Shape AUC	Rank
2G5N	23M	N=C(c1ccc2c(c1)cc([nH]2)c3cccc(c3)C)/N	0.777	0.753	0.729	44
1C1P	BAI	c1ccc2c(c1)[nH]c(n2)Cc3[nH]c4ccc(cc4n3)C(=N)N	0.776	0.736	0.737	45
3RXP	SW3	Cc1cc(nn1C)CN	0.776	0.742	0.770	46
2G8T	MI2	N=C(c1ccc2c(c1)cc([nH]2)c3ccccc3C)/N	0.776	0.752	0.723	47
1O2S	CR4	c1ccc(c(c1)c2[nH]c3ccc(cc3n2)C(=[NH2+])N)[O-]	0.773	0.765	0.727	48
4AB9	VXQ	c1cc2c(cc1CN)CCO2	0.773	0.765	0.747	49
1GI1	BMZ	c1ccc(c(c1)c2[nH]c3ccc(cc3n2)C(=[NH2+])N)O	0.772	0.780	0.731	50
4AB8	VXU	c1cc(c2c(c1)OCCCO2)CN	0.772	0.759	0.727	51
1GI4	122	c1ccc(c(c1)c2[nH]c3cc(ccc3n2)C(=[NH2+])N)O	0.771	0.751	0.730	52
3AAS	GUS	N=C(N)/Nc1ccc(c(c1)/C=N/[C@@H](C)C(=O)O)O	0.771	0.825	0.709	53
3RXF	4AP	c1c[nH+]ccc1N	0.767	0.758	0.721	54
3A7X	O09	N=C(c1ccc(cc1)/C=N/OCC(=O)O)\N	0.767	0.795	0.678	55
1TNK	PRA	c1ccc(cc1)CCC[NH3+]	0.766	0.737	0.753	56
2OTV	NCA	c1cc(cnc1)C(=O)N	0.766	0.726	0.801	57
3RXD	SZ4	COc1cccc(c1)CN	0.765	0.753	0.752	58
3NK8	JKZ	C1CC2=C(C1)NC(=O)C=C2C(F)(F)F	0.765	0.737	0.717	59
4ABG	91B	CN1CCN(CC1)c2cccc(c2)CN	0.765	0.750	0.727	60
2FX6	270	C1=CC2=NC(N=C2C=C1)N	0.765	0.742	0.775	61
3RXH	HSM	c1c(nc[nH]1)CCN	0.764	0.746	0.707	62
4ABD	SW2	c1ccn(c1)c2cccc(c2)CN	0.762	0.752	0.759	63
5MNK	ABN	c1ccc(cc1)CN	0.762	0.759	0.787	64
1GI5	123	COc1ccc(c(c1)c2[nH]c3ccc(cc3n2)C(=[NH2+])N)O	0.761	0.739	0.730	65
4ABE	913	c1cc(cc(c1)n2cccn2)CN	0.759	0.745	0.766	66
1TNL	TPA	c1ccc(cc1)[C@H]2C[C@@H]2[NH3+]	0.759	0.740	0.784	67
4ABH	7Z3	c1cc(cc(c1)N2CCCC2)CN	0.758	0.751	0.739	68
5MNC	ANL	c1ccc(cc1)N	0.757	0.759	0.739	69
1TNH	FBA	c1cc(ccc1C[NH3+])F	0.750	0.745	0.763	70
5JYI*	SIN	C(CC(=O)O)C(=O)O	0.552	0.532	0.552	71
2TGD*	DFP	CC(C)OP(=O)OC(C)C	0.550	0.539	0.548	72

*** Indicates the representative fragments from $frags_{clustered,t=0.8}$**

Table B-10. $Frags_{all}$ for TRY1. Each fragment was used individually to rescore the DUD-E dataset for TRY1 using SuCOS.

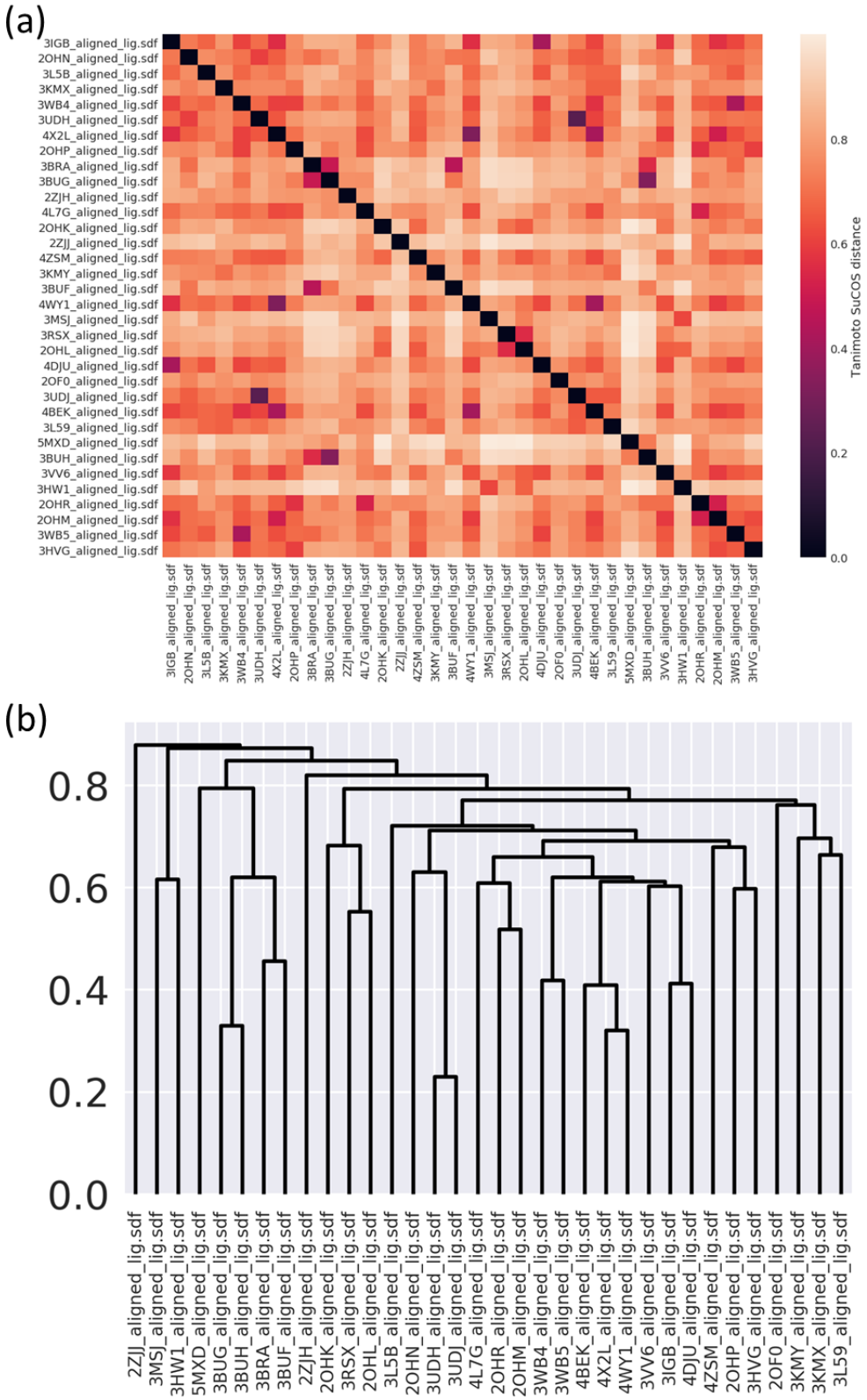


Figure B.2. (a) Heat map showing the distance matrix between all of the 34 BACE1 fragments. The distances were calculated using Tanimoto SuCOS (Section 4.2.3). Fragments with exact overlap of shape and chemical features have a distance of zero, whereas fragments which do not have any overlap of shape or chemical features have a distance of one. (b) The dendrogram obtained from hierarchical clustering, using the *average* method.

Appendix C : Chapter 5

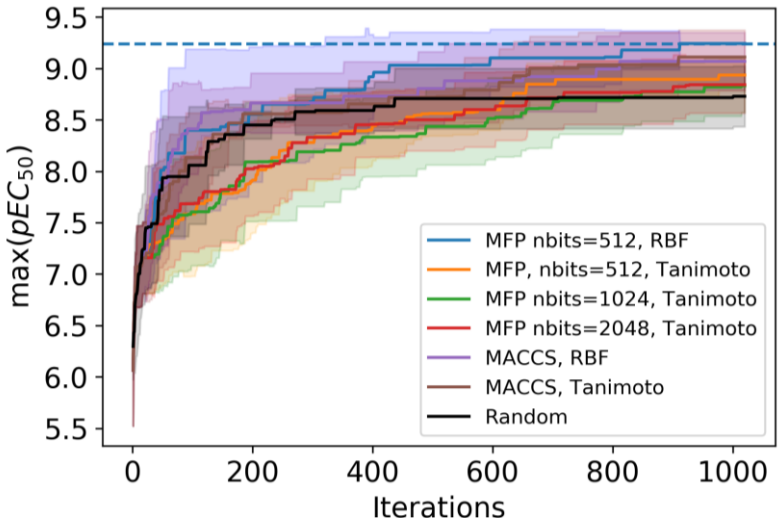
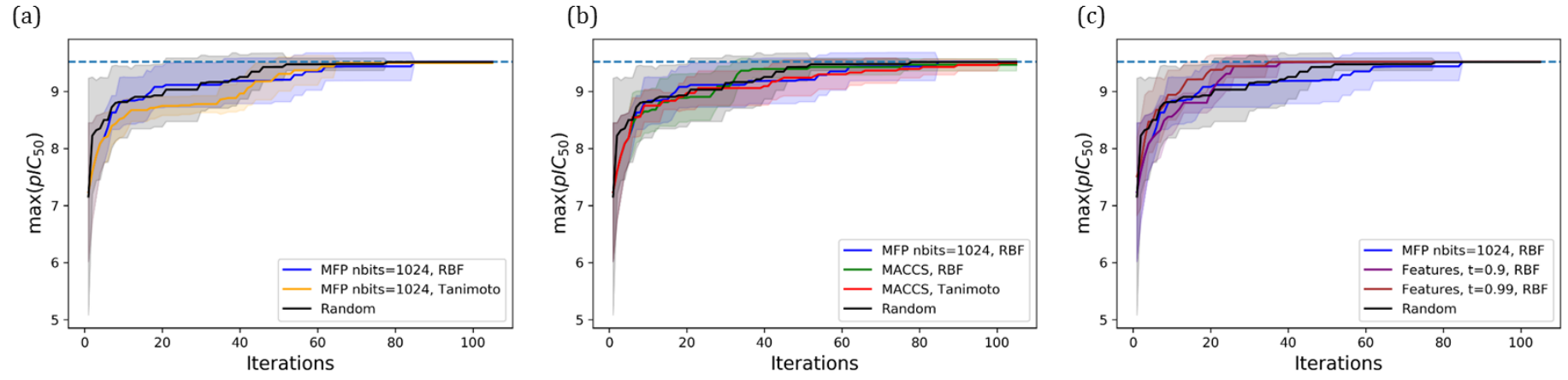
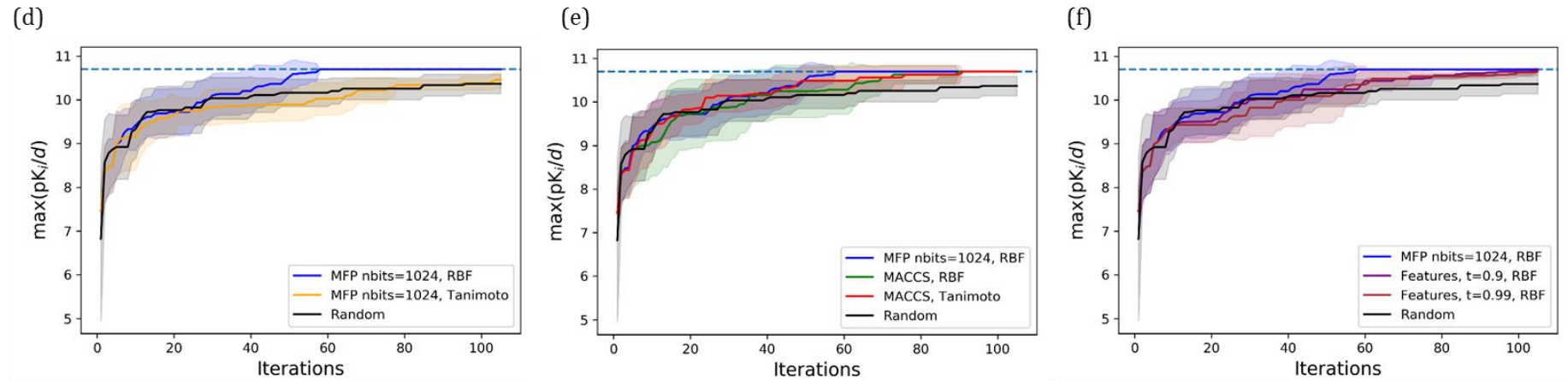


Figure C.1. Evolution of the maximum pEC_{50} for the different Bayesian optimisation methods for the malaria dataset. This figure is the same as Figure 5.12 but the ± 1 standard deviation errors from each method's average are shown by the shaded regions.

BACE1


(a)
max(pIC50)
Iterations
MFP nbits=1024, RBF
MFP nbits=1024, Tanimoto
Random
(b)
max(pIC50)
Iterations
MFP nbits=1024, RBF
MACCS, RBF
MACCS, Tanimoto
Random
(c)
max(pIC50)
Iterations
MFP nbits=1024, RBF
Features, t=0.9, RBF
Features, t=0.99, RBF
Random

CAH2


(d)
max(pKi/d)
Iterations
MFP nbits=1024, RBF
MFP nbits=1024, Tanimoto
Random
(e)
max(pKi/d)
Iterations
MFP nbits=1024, RBF
MACCS, RBF
MACCS, Tanimoto
Random
(f)
max(pKi/d)
Iterations
MFP nbits=1024, RBF
Features, t=0.9, RBF
Features, t=0.99, RBF
Random

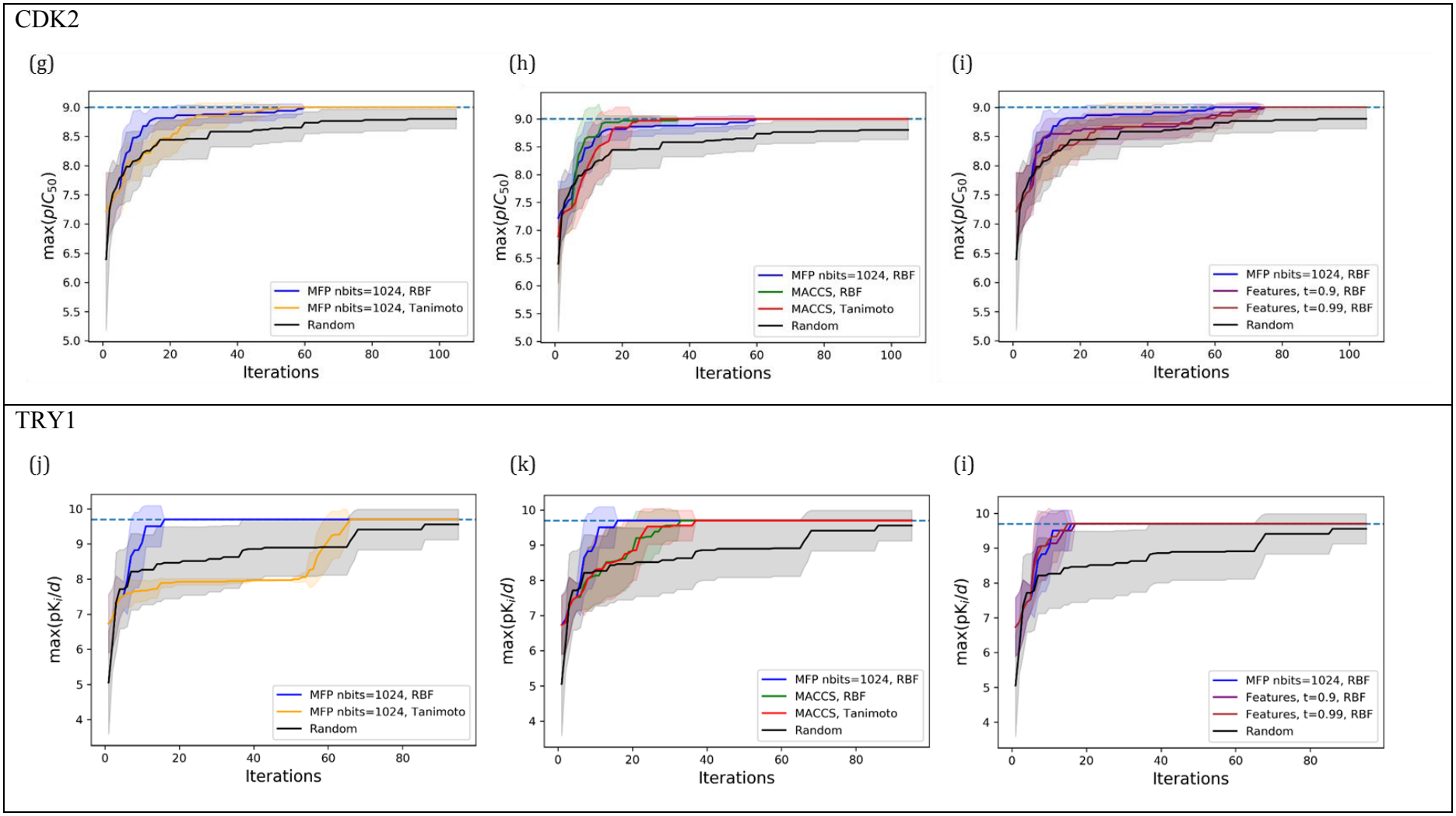


Figure C.2. Evolution of the maximum pIC_{50} or $pK_{i/d}$ found with iteration number for the four targets. This figure is the same as Figure 5.18 but for each method, the ± 1 standard deviation errors from the average are shown by the shaded regions, and for each target the plot has been split into three separate plots to show the results of the different methods compared to random sampling and Bayesian optimisation with MFP (*nbits*=1024) and the RBF kernel.

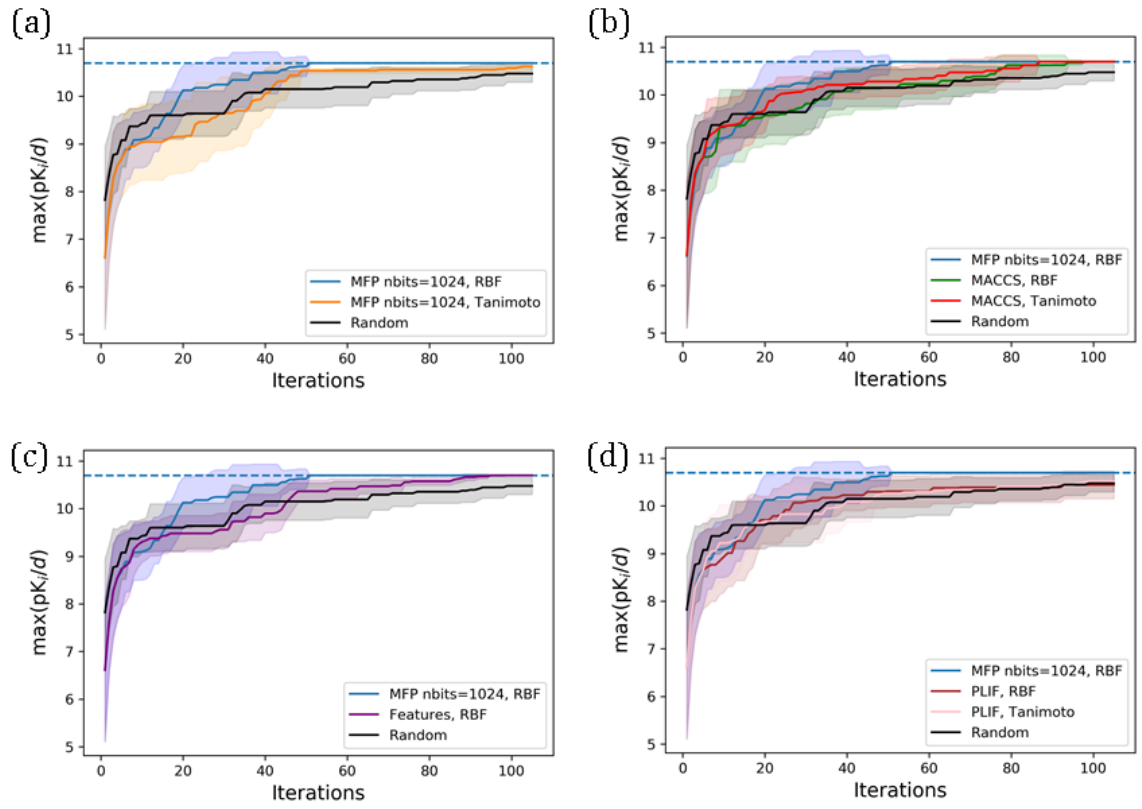


Figure C.3. Bayesian optimisation was run with the various methods on the CAH2 dataset from PDBbind, curated for the investigation of PLIFs as the Bayesian optimisation search space. This figure is the same as Figure 5.23 but for each method, the ±1 standard deviation errors from the average are shown by the shaded regions. For clarity, the plot has been split into four separate plots (a)-(d) to show the results of the separate methods compared to random sampling and Bayesian optimisation with MFP (*nbits*=1024)and the RBF kernel.