



The
University
Of
Sheffield.

Substructural Analysis Using Evolutionary Computing Techniques

By:

Nor Samsiah Binti Sani

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

The University of Sheffield
Faculty of Social Sciences
School of Information

February 2016

Acknowledgments

First and foremost, I would like to thank my supervisors Professor Peter Willett and Dr John Holliday for their constant supervision and guidance during this PhD study, and also during the writing up of this thesis. I can't thank you enough for all the fruitful discussions and advices given in the throughout the research. I would like to thank all the staff, colleagues and members of the University of Sheffield's Chemical Information System Research Group (CISRG) for their kindness and support during the course of this work. Not forgetting Dr Sudholt from the University of Sheffield's School of Computer Science for his valuable insights and discussions on evolutionary algorithms.

I would like to convey my deepest thanks to my whole family members in Malaysia, especially my parents (Hj Sani Ithnin and Hj Artini Dayat) and my parents-in-law (Hj Azmi Anuar and Hj Ungku Mariam) for their love, care and constant prayer for my success. To both my siblings, Nor Suriani and Mohd Aliff Afira, for their never-ending words of encouragements. Special appreciation to my husband Ahmad Kamal Azmi, the one whom have persistently given me strength, motivation, encouragements when needed, and sometimes when all else fails, just practically a shoulder to lean on. To my children, Arissa Humaira and Adam Ashraf, you both cheer me up constantly no matter how busy I get, and I truly appreciate it. I would never have made this far without their support and love. I would like to dedicate this achievement to all of you.

I would also like to thank my main sponsors: The Ministry of Higher Education (MOHE), Malaysia and also my employer, Universiti Kebangsaan Malaysia (UKM) for the opportunity given in pursuant of this PhD study. Thank you.

Abstract

Substructural analysis (SSA) was one of the very first machine learning techniques to be applied to chemoinformatics in the area of virtual screening. For this method, given a set of compounds typically defined by their fragment occurrence data (such as 2D fingerprints). The SSA computes weights for each of the fragments which outlines its contribution to the activity (or inactivity) of compounds containing that fragment. The overall probability of activity for a compound is then computed by summing up or combining the weights for the fragments present in the compound. A variety of weighting schemes based on specific relationship-bound equations are available for this purpose. This thesis identifies uplift to the effectiveness of SSA, using two evolutionary computation methods based on genetic traits, particularly the genetic algorithm (GA) and genetic programming (GP). Building on previous studies, it was possible to analyse and compare ten published SSA weighting schemes based on a simulated virtual screening experiment. The analysis showed the most effective weighting scheme to be the R4 equation which was a part of document-based weighting schemes. A second experiment was carried out to investigate the application of GA-based weighting scheme for the SSA in comparison to an experiment using the R4 weighting scheme. The GA algorithm is simple in concept focusing purely on suitable weight generation and effective in operation. The findings show that the GA-based SSA is superior to the R4-based SSA, both in terms of active compound retrieval rate and predictive performance. A third experiment investigated the genetic application via a GP-based SSA. Rigorous experiment results showed that the GP was found to be superior to the existing SSA weighting schemes. In general, however, the GP-based SSA was found to be less effective than the GA-based SSA. A final experiment is described in this thesis which sought to explore the feasibility of data fusion on both the GA and GP. It is a method producing a final ranking list from multiple sets of ranking lists, based on several fusion rules. The results indicate that data fusion is a good method to boost GA-and GP-based SSA searching. The RKP rule was considered the most effective fusion rule.

Table of Contents

Acknowledgments	ii
Abstract	iii
Table of Contents	iv
List of Figures	x
List of Tables	xvii

CHAPTER 1: Introduction

1.1 The drug discovery process	1
1.2 Chemoinformatics and the use of machine learning methods	3
1.3 Research aim and objectives.....	5
1.4 Thesis outline.....	6

CHAPTER 2: Virtual Screening and Substructural Analysis

2.1 Introduction	7
2.2 Representation of chemical structures	7
2.2.1 Connection tables	7
2.2.2 Morgan algorithm.....	8
2.2.3 Line notation	10
2.2.4 Wiswesser Line Notation (WLN).....	11
2.2.5 Simplified Molecular Input Line Entry System (SMILES)	11
2.2.6 International Chemical Identifier (InChI)	12
2.3 Molecular descriptors	14
2.4 2D descriptors.....	16
2.4.1 Topological indices	16
2.4.2 Fragment-based descriptors.....	16
2.5 Searching databases of molecules	20
2.5.1 Structure searching	20
2.5.2 Substructure searching	21
2.5.3 Similarity searching.....	22
2.6 Virtual and High-Throughput Screening	23

2.7	Structure-Activity Relationship (SAR) and Quantitative Structure-Activity Relationship (QSAR)	27
2.8	Machine learning	30
2.9	Substructural analysis	35
2.9.1	History and definition.....	35
2.9.2	Fundamental components of the SSA	35
2.9.3	The SSA weighting schemes	37
2.10	Application of SSA for drug discovery.....	44
2.10.1	Hodes study in National Cancer Institute (NCI) for tumour screening program.....	45
2.10.2	CASE and MULTICASE	46
2.10.3	SLASH	47
2.10.4	Other applications of SSA.....	47
2.11	SSA and Naive Bayesian Classifier (NBC).....	48
2.12	Performance evaluation and validation for SSA.....	51
2.13	Conclusion	52

CHAPTER 3: Research Methodology

3.1	Introduction	55
3.2	Datasets.....	55
3.2.1	MDDR.....	55
3.2.2	WOMBAT	56
3.2.3	ChEMBL	56
3.3	Fingerprints.....	58
3.4	Test set and training set	59
3.5	SSA weighting schemes evaluation methods	62
3.5.1	Enrichment Factor (EF).....	63
3.5.2	Analysis of diversity.....	64
3.5.3	Statistical tests	65
3.5.3.1	Kendall's W analysis	65
3.5.3.2	Wilcoxon signed rank test.....	66
3.6	Hardware	68
3.7	Conclusion	69

CHAPTER 4: The Comparison of Different Weighting Schemes in Substructural Analysis Using Large Datasets

4.1	Introduction	70
4.2	Experimental details	70
4.3	Experimental procedure.....	70
4.3.1	Weighting schemes.....	71
4.3.2	Benchmarking SSA performance against NBC Pipeline Pilot.....	71
4.4	Analysis of SSA weighting schemes	71
4.4.1	Enrichment curve analysis	72
4.4.2	Kendall’s W analysis	73
4.4.3	Analysis of the SSA R4 fragment weights	76
4.5	Discussion.....	78
4.6	Conclusion.....	80

CHAPTER 5: Genetic Algorithm Approach to Substructural Analysis

5.1	Introduction	96
5.2	Fundamental components of GA for SSA	96
5.2.1	Encoding of chromosomes	97
5.2.2	Fitness criterion	98
5.2.3	Chromosome selection methods for genetic operations.....	98
5.2.4	Evolutionary operators	99
5.2.5	Chromosome’s principle of elitism	100
<i>Simple-state preservation model</i>	101	
<i>Steady-state preservation model</i>	101	
5.3	Previous works in GA.....	102
5.4	Experiment details	104
5.4.1	Dataset.....	104
5.4.2	Hardware	104
5.4.3	Algorithm implementation	105
5.4.3.1	Suitable fitness function for SSA–based GA.....	106
5.4.3.2	Weight polarity to overcome overfitting.....	108
5.4.3.3	Selection of chromosomes and genetic operations	111
5.5	Experimental procedure.....	114

5.6	Experiment setup: Parameterisation of GA-based SSA	115
5.6.1	Fitness function	116
5.6.2	GA weight range of chromosomes	117
5.6.3	Population size and generations of evolution.....	117
5.6.4	Elitism model	119
5.6.5	Evolution control.....	119
5.6.6	Final parameterisation selections	120
5.7	Experiment result: Analysis of performance of GA-based SSA	121
5.7.1	GA robustness	121
5.7.2	GA weights correlation and consistency of compounds retrieval	122
5.7.3	Analysis of GA runs on all activity classes	123
5.7.3.1	Enrichment curve analysis	123
5.7.3.2	Analysis of diversity	124
5.7.4	Wilcoxon signed rank test	125
5.7.5	Model validation with Y-randomisation.....	125
5.7.6	Run-time benchmarks of GA-based SSA.....	127
5.8	Discussion.....	128
5.9	Conclusion.....	129

CHAPTER 6: Genetic Programming Approach to Substructural Analysis

6.1	Introduction	159
6.2	Fundamental components of GP for SSA.....	159
6.2.1	Encoding of Chromosomes	161
6.2.2	Population growth	162
6.2.3	Evolutionary operators	163
6.3	Previous works in GP	163
6.4	Experimental details	164
6.4.1	Dataset.....	164
6.4.2	Hardware	164
6.4.3	Algorithm Implementation.....	165
6.4.3.1	Chromosomes population and generation.....	165
6.4.3.2	Suitable fitness function design for GP-based SSA	167
6.5	Experimental procedure.....	167

6.6	Experiment setup: Parameterisation of the GP-based SSA.....	168
6.6.1	Fitness function	169
6.6.2	Terminal and function sets for chromosome generation	170
6.6.3	Chromosome structure	171
6.6.4	Population size, generation and tree properties.....	172
6.6.5	Elitism model	173
6.6.6	Evolution control.....	174
6.6.7	Final parameterisation selections	174
6.7	Experiment result: Analysis of the performance of GP-based SSA	175
6.7.1	GP robustness	175
6.7.2	GP weights correlation and consistency of compounds.....	176
6.7.3	Analysis of GP runs on all activity classes.....	177
6.7.3.1	Enrichment curve analysis	178
6.7.3.2	Analysis of diversity	179
6.7.4	Kendall's W analysis	179
6.7.5	Wilcoxon signed rank test	180
6.7.6	GP-generated equations.....	181
6.7.7	Model validation with Y-randomisation.....	181
6.7.8	Run-time benchmarks of GP-based SSA	183
6.8	Discussion.....	184
6.9	Conclusion.....	185

CHAPRER 7: Investigations Into The Application of Data Fusion

7.1	Introduction	228
7.2	Data fusion.....	228
7.3	Experimental details	230
7.3.1	Datasets	230
7.3.2	Fusion rules	230
7.4	Results and discussion	232
7.4.1	GA and GP-based fusion performance analysis.....	232
7.4.2	Kendall's W analysis	233
7.4.3	Wilcoxon signed rank test	235
7.5	Conclusion.....	237

CHAPTER 8: Conclusion and Future Work

- 8.1 Introduction 247
- 8.2 Contributions 248
 - 8.2.1 The comparison of existing SSA weighting schemes 248
 - 8.2.2 The use of GA to the SSA method 249
 - 8.2.3 Investigation of the use of GP in the SSA method 250
 - 8.2.4 Investigation into the use of data fusion to the GA-and GP-based SSA..... 251
- 8.3 Suggestion for future work 251
- 8.4 Conclusion 252
- Appendix A: The GA-based SSA pseudocode 254
- Appendix B: The GP-based SSA pseudocode..... 256
- REFERENCES 258

List of Figures

Figure 1.1: Trends in pharmaceutical research for drug discovery and design (after Terstappen and Reggiani, 2001)	2
Figure 2.1: Connection table representation (after Leach & Gillet, 2007). Information such as atom coordinates relating to its formation and bonding pairs are stored for accurate rebuilding of the molecules	8
Figure 2.2: Morgan algorithm procedure (after Leach & Gillet, 2007)	9
Figure 2.3: Canonical state of structure for computer representation (adapted from UTM open courseware chemistry website)	11
Figure 2.4: 2D chemical structure (left) and structural key fingerprint (right)	17
Figure 2.5: Graph A and B are isomorphic and topologically identical (after Yirka, 2015)	21
Figure 2.6: Schema of chemical substructure searches (after Horai et al., 2010)	22
Figure 2.7: Overview of typical screening process (after Lengauer, Lemmen, Rarey, & Zimmermann, 2004)	25
Figure 2.8: Different types of QSAR strategies (after Todeschini & Consonni, 2008): (a) Regression model focusing on the best fit classifications; (b) Classification model to characterise the similarity between certain properties, and (c) Partial order ranking models, based on Hasse diagram technique. The figure shows the ranks of chemicals according to their toxicity levels, which relies on statistical significance	28
Figure 2.9: Simple schematics of Substructural Analysis	36
Figure 2.10: Plot of weight versus number of standard deviation with conversion to log 1/P (Hodes et al., 1977)	41
Figure 3.1: (a) A molecule from the ChEMBL database and (b) Its corresponding Murcko scaffold	58
Figure 3.2: Pipeline Pilot workflow for fingerprint	59

Figure 4.1: Methodology of experimental procedures conducted in predictive analysis.....	82
Figure 4.2: Pipeline Pilot workflow to generate NBC models using training sets.....	82
Figure 4.3: Pipeline Pilot workflow for new candidate screening using generated NBC models	82
Figure 4.4: Cumulative recall plots of the various SSA weighting schemes for the 5HT3 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset.....	83
Figure 4.5: Cumulative recall plots of the various SSA weighting schemes for the COX activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset.....	84
Figure 4.6: Cumulative recall plots of the various SSA weighting schemes for the D2 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset.....	85
Figure 4.7: Cumulative recall plots of the various SSA weighting schemes for the RNN activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset.....	86
Figure 4.8: Cumulative recall plots of the various SSA weighting schemes for the PKC activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset.....	87
Figure 4.9: Comparison of the eleven MDDR activity classes based on the enrichment factor of actives in the top 1% of the rankings	88
Figure 4.10: Comparison of the fourteen WOMBAT activity classes based on the enrichment factor of actives in the top 1% of the rankings.....	88
Figure 4.11: Comparison of the fifteen ChEMBL activity classes based on the enrichment factor of actives in the top 1% of the ranking	89
Figure 4.12: Comparison of fragment weights of 166 fragments, at m equals 0.0000001, 0.01, 0.05, 0.1 and 0.5. The SSA R4 weights are computed using the training sets of predictive analysis of COX activity class in the MDDR database.....	89
Figure 5.1: The basic genetic algorithm flowchart.....	131
Figure 5.2: Roulette wheel selection after Goldberg (1987). (a) Outlines a set of evaluated chromosomes with different fitness scores, and their relative percentage of the total fitness.	

(b) The chromosomes are sorted and fitted into a roulette wheel model where larger chromosomes take a bigger portion of the wheel. A random number generated ranging from 0 to 100% will iterate through the wheel until the value is achieved, thus selecting the parent chromosome 132

Figure 5.3: Tournament selection. (a) Four chromosomes are selected at random and assigned as paired opponents. Fitness scores are observed between opponents and the winner progresses to the next round. (b) The winners of round 1 pitted against one another in round 2 by observing their fitness score. (c) The winner of round 2 is selected as the parent chromosome for the next genetic operation. This process is repeated to select the other parent chromosome as genetic operation requires two parents to proceed 132

Figure 5.4: Genetic operations in the genetic algorithm. (a) A population list consisting of chromosomes represented as bit-strings. (b) Two parent chromosomes selected from the population list to perform a crossover operation, which takes a portion of each chromosome’s genes and recombines them into a single, new chromosome. (c) Mutation operation flips a random bit of the chromosome. (d) Child chromosome inserted back into the population list 133

Figure 5.5: Crossover methods in the GA. (a) One point crossover method; (b) Two points crossover, and (c) Uniform crossover 133

Figure 5.6: Simple-state elitism model with six chromosomes created at initialisation. After going through reproduction process, each generation concludes with modification to the population. In this case, an elitism of two parents ensures that all the chromosomes are replaced through genetic operations except for the two best parent chromosomes 134

Figure 5.7: Steady-state elitism model with six chromosomes created at initialisation. During the replacement process only one chromosome, being the worst performing one, is replaced with a reproduced, offspring chromosome. The remaining chromosomes are maintained, which is also known as the overlapping population method..... 134

Figure 5.8: The genetic algorithm flowchart with inclusion of weight polarity constraining operations. The GA assumes normal operation except that the weight polarity needs to be identified first, and both the population initialisation and subsequent genetic operations include conditional weight assignment based on the polarity criterion 135

Figure 5.9: Cumulative recall plots of the GA-based SSA against SSA R4 for the RNN activity class from the MDDR dataset based on the different fitness function (a) In the top 1%; and (b) In top 10% of ranked compounds.....	136
Figure 5.10: Cumulative recall plots of the GA-based SSA against SSA R4 for the COX activity class from the MDDR dataset based on the different fitness function (a) In the top 1%; and (b) In top 10% of ranked compounds.....	137
Figure 5.11: Error plot of training set versus predicted test set of the GA-based SSA following GA iterations for MDDR (a) RNN and (b) COX activity classes. Both GA instances were executed based on a chromosome population of 200 and maximum iteration of 500 to signify (i) Overfitting case, and (ii) Presence of improved recall rates in large iterations.....	138
Figure 5.12: The cumulative recall of active compounds plotted against the entire compound over 10 runs of the GA program: (a) GA instances for MDDR-based RNN activity class; (b) GA instances for MDDR-based COX activity class	139
Figure 5.13: Cumulative recall plots of the GA-based SSA against SSA R4 for the 5HT3 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	140
Figure 5.14: Cumulative recall plots of the GA-based SSA against SSA R4 for the COX activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	141
Figure 5.15: Cumulative recall plots of the GA-based SSA against SSA R4 for the D2 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method.....	142
Figure 5.16: Cumulative recall plots of the GA-based SSA against SSA R4 for the RNN activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	143
Figure 5.17: Cumulative recall plots of the GA-based SSA against SSA R4 for the PKC activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	144

Figure 5.18: Permutation plots (Y-randomisation) of the MDDR-based (a) RNN and (b) COX classes, with weights calculated and applied to non-permuted test sets	145
Figure 6.1: Genetic programming basic flow, (after Poli, Langdon, McPhee & Koza, 2008)	187
Figure 6.2: Simple structure of a tree model in GP, (a) a valid tree model compared to (b) invalid tree with incomplete equation portion in its child branch, highlighted in red	187
Figure 6.3: Chromosome tree creation using the grow method. Tree defined with a maximum depth of 3 levels	188
Figure 6.4: Chromosome tree creation using the full method. Tree defined with a maximum of 3 levels of tree depth	188
Figure 6.5: GP's crossover operation diagram	189
Figure 6.6: GP's mutation example showing the (a) Single terminal mutation, and (b) Sub-tree mutation.....	189
Figure 6.7: A GP equation producing chaotic fragment weights of inappropriate large values. The multiplication of a large variable <i>TOT</i> with itself while enhanced by the accompanying exponential term causes a number of weights to be significantly larger in value.....	190
Figure 6.8: An equation from Figure 6.7 now wrapped by a mandatory log function generates much smaller weight values	190
Figure 6.9: GP representation of chromosomes towards fitness determination. (a) A training set made up of compounds via 2D fingerprints description. (b) A GP population representing chromosome equations made up of parameters to explain training set. (c) Chromosome chr1 is translated from the equation form to weight values, applied to training set to determine compound score. (d) Compounds are ranked in descending order. (e) Fitness of chromosome is calculated as the rate of the active retrieval in the top percentile of the ranked compounds set	191
Figure 6.10: Cumulative recall plots of the GP-based SSA against SSA R4 for the RNN activity class from the MDDR dataset based on the different fitness function in (a) The top 1%; and in (b) The top 10% of ranked compounds.....	192

Figure 6.11: Cumulative recall plots of the GP-based SSA against SSA R4 for the COX activity class from the MDDR dataset based on the different fitness function in (a) The top 1%; and in (b) The top 10% of ranked compounds.....	193
Figure 6.12: Error plot of training set versus predicted test set of the GP-based SSA for MDDR RNN activity class, based on (a) VARIABLES_A set only, and (b) VARIABLES_A and VARIABLES_B combination.....	194
Figure 6.13: Error plot of training set versus predicted test set of the GP-based SSA for MDDR COX activity class, based on (a) VARIABLES_A set only, and (b) VARIABLES_A and VARIABLES_B combination.....	194
Figure 6.14: The Cumulative recall of active compounds plotted against the entire compound over 10 runs of the GP program: (a) GP instances for MDDR-based RNN activity class; (b) GP instances for MDDR-based COX activity class	195
Figure 6.15: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for 5HT3 activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	196
Figure 6.16: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for COX activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	197
Figure 6.17: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for D2 activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	198
Figure 6.18: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for RNN activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	199
Figure 6.19: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for PKC activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method	200

Figure 6.20: Permutation plots (Y-randomisation) of the MDDR-based (a) RNN and (b) COX classes, with weights calculated and applied to non-permuted test sets 201

Figure 6.21: Example of a GP-based SSA (a) Original equation and (b) The simplified equation using Wolfram Alpha expression simplifier online tool 202

Figure 7.1: Fusion rules..... 231

List of Tables

Table 2.1: Linear chemical notation	14
Table 2.2: Approaches in ligand-based and structure-based virtual screening.....	26
Table 2.3: Comparison of machine learning classification techniques in LBVS (after Lavecchia, 2015).....	32
Table 2.4: Summary of past performance evaluation programs on SSA	53
Table 3.1: (a) MDDR, (b) WOMBAT and (c) ChEMBL activity classes considered in this study	60
Table 3.2: Computer's hardware specification used to run the GA-based SSA. (a) Server setup; (b) Multimedia-intensive workstation; (c) Office-level workstation; and (d) Laptop setup	69
Table 4.1: Enrichment factor of actives retrieved in the top 1% of the ranked compounds of (a) The eleven activity class in MDDR (b) The fourteen activity class in WOMBAT dataset, and (c) The fifteen activity classes in ChEMBL dataset	90
Table 4.2: Kendall's W analysis for the top 1% actives retrieved of the ranking for (a) Eleven activity classes in MDDR, (b) Fourteen activity classes in WOMBAT, and (c) Fifteen activity classes of the ChEMBL database	92
Table 4.3: Kendall's W analysis for the top 1% based on the average of enrichment factor actives in the top 1% from the MDDR, WOMBAT and ChEMBL databases	93
Table 4.4: Fragment weights computed using R4 function for fourteen fragments (1, 2, 5, 19, 13, 17, 74, 67, 38, 53, 166, 164, 127 and 154) at m equals to 0.0000001, 0.01, 0.05, 0.1 and 0.5. The weights were derived using training sets in the predictive analysis.....	94
Table 4.5: Statistics for 166 fragment weights for m equals 0.0000001, 0.01, 0.05, 0.1 and 0.5. The SSA R4 fragment weights are derived using the training set of predictive analysis of COX activity classes in the MDDR database	95

Table 5.1: Example of a GA operation based on a population containing five molecules, with six chromosomes created at initialisation. (a) Fingerprints for five molecules M1-5 encoding five different substructural fragments F1-5; (b) The molecule activity state, 1 referring to an active molecule, while 0 represents inactive ones. (c) Six chromosomes C_{1-6} encoding the weights W_{1-5} for F_{1-5} ; and (d) Sums-of-weights using each chromosome C_{1-6} for each molecule M_{1-5} 146

Table 5.2: Weight polarity determination using the SSA R4 weighting scheme, following the example molecule and activity dataset in Table 5.1. (a) Summary of a five-fragment dictionary based on the common properties in SSA weighting schemes. (b) Weight polarity for fragments is determined on the basis of greater value between the rate-of-actives (*ROA*) against the rate-of-inactives (*ROI*). (c) The equivalent weight values and its polarity calculated using the SSA R4 weighting scheme 147

Table 5.3: Fitness score calculation using chromosome C_3 with its weights restricted by the weight polarity. (a) Chromosome C_1 weights combination and the corresponding weight polarity. (b) Assignment of chromosome weight to each fragment in the molecule set. (c) Sum of fragments' score of each molecule. (d) Ranking of chromosome based on molecule score in descending order from largest to smallest. All the active molecules are seen to benefit from SSA R4 weight polarity assignment based on their rankings at the top..... 148

Table 5.4: Top 1% active retrieval rates for the fitness function parameter test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters..... 149

Table 5.5: Top 1% active retrieval rates for the GA weight range parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters..... 149

Table 5.6: Top 1% active retrieval rates for the GA Population and Generation parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters 150

Table 5.7: Top 1% active retrieval rates for the elitism model parameter. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters..... 151

Table 5.8: Top 1% active retrieval rates for the evolution control parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters..... 152

Table 5.9: The top-ranked molecules in ten GA runs based on test set applied data, showing the occurrences of ranked compounds based on GA run-1 that fall outside the top 1% in the other nine remaining GA runs using the (a) RNN and (b) COX activity classes in the MDDR dataset. The numbers in brackets show the number of actives actually retrieved in the top 1% for that particular GA-run 153

Table 5.10: Enrichment curve of actives count in the top 1% for ten GA runs of (a) Eleven activity classes in MDDR dataset; (b) Fourteen activity classes in WOMBAT dataset and; (b) Fifteen activity classes in ChEMBL dataset. Included are the mean and standard deviation for the Pearson correlation coefficients between the sets of 166 weights computed for each distinct pair of runs..... 154

Table 5.11: Screening results using the GA-based SSA and its comparison to the SSA R4 weighting scheme for the (a) MDDR; (b) WOMBAT and (c) ChEMBL datasets. The number of actives retrieved at the top 1% based on the worst performing GA runs is recorded for the

calculation of Tanimoto coefficient and the <i>BemisMurckoAssemblies</i> based diversity analysis	156
Table 5.12: GA run-time benchmark at different iterations using the 10% training set of the RNN activity class, based on the (a) MDDR, (b) WOMBAT and (c) ChEMBL databases. Parameterisation of the GA is based on the final chosen ones as in Section 5.7.5, among them the population of 200 chromosomes, and a maximum iteration 200 evolutions.....	158
Table 6.1: Summary of differences between GA and GP.....	203
Table 6.2: Top 1% active retrieval rates for the GP fitness function definition test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded	204
Table 6.3: Terminal and function variable combinations used for chromosome initialisation. (a) List of tested variable combinations defined for the GP chromosome terminal set; (b) List of tested operator functions for the GP chromosome function set	205
Table 6.4: Top 1% active retrieval rates for the GP terminal and function set combination test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded.....	206
Table 6.5: Top 1% active retrieval rates for the GP's chromosome structure tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. Each parameter's tree depth and node size values obtained are listed as well. The GPs was performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded.....	207

Table 6.6: Top 1% active retrieval rates for the GP's population and generation based parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded 208

Table 6.7: Top 1% active retrieval rates for the GP's elitism model parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded 209

Table 6.8: Top 1% active retrieval rates for the GP's evolution control parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded..... 210

Table 6.9: The top-ranked molecules in ten GP runs based on test set applied data, showing the occurrences of ranked active compounds based on GP run-1 that fall outside the top 1% in the other nine remaining GP runs using the (a) RNN and (b) COX activity classes in the MDDR dataset. The numbers in brackets show the number of actives actually retrieved in the top 1% for that particular GP-run..... 211

Table 6.10: Enrichment curve of actives count in the top 1% for ten GP runs of (a) Eleven activity classes in MDDR dataset; (b) Fourteen activity classes in WOMBAT dataset and; (b) Fifteen activity classes in ChEMBL dataset. Included are the mean and standard deviation for the Pearson correlation coefficients between the sets of 166 weights computed for each distinct pair of runs..... 212

Table 6.11: Screening results using the GP-based SSA and its comparison to the SSA R4 and GA-based SSA weighting scheme for the (a) MDDR; (b) WOMBAT and (c) ChEMBL datasets. The number of actives retrieved at the top 1% based on the worst performing GA

runs is recorded for the calculation of Tanimoto coefficient and the <i>BemisMurckoAssemblies</i> based diversity analysis	214
Table 6.12: Kendall's W analysis for the top 1% actives retrieved of the ranking for (a) Eleven activity classes in MDDR, (b) Fourteen activity classes in WOMBAT, and (c) Fifteen activity classes of the ChEMBL database	217
Table 6.13: The worst and best performing GP equations selected from the 10 runs of each activity class, based on the (a) MDDR, (b) WOMBAT, and (c) ChEMBL18 datasets. All equations are simplified from its original form.....	218
Table 6.14: GP run-time benchmark at different iterations using the 10% training set of the RNN activity class, based on the (a) MDDR, (b) WOMBAT and (c) ChEMBL databases. Parameterisation of the GP is based on the final chosen ones as in Section 6.6.1.6, among them the population of 200 chromosomes, and a maximum iteration 200 evolutions.....	227
Table 7.1: Enrichment factor of actives when using combination of different GA runs for top 1% in (a) MDDR dataset for eleven activity classes (b) WOMBAT dataset for fourteen activity classes and (c) ChEMBL dataset for fifteen activity classes.....	238
Table 7.2: Enrichment factor of actives when using a combination of different GP runs for the top 1 % in (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset for fifteen activity classes.....	240
Table 7.3: Kendall's W analysis for the number of actives retrieved in top 1% of the GA searches and after application of data fusion on (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset of fifteen	242
Table 7.4: Kendall's W analysis for the number of actives retrieved in top 1% of the GP searches and after application of data fusion on (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset of fifteen	244
Table 7.5: Kendall's W analysis for the top 1% based on the average of enrichment factor actives in the top 1% of (a) The GA-based SSA and (b) GP-based SSA from the MDDR, WOMBAT and ChEMBL databases	246

Chapter 1

Introduction

1.1 The drug discovery process

The drug discovery process can be divided into five distinct stages. These are target identification, lead selection, lead optimisation, preclinical testing and clinical development. The target is a protein involved with a particular identified disease. It is commonly known that drug discovery is both time consuming and expensive, and that the risks of drug design failure increase after each stage within a typical pharmaceutical research timeline. Rishton (2005) reported a high failure rate suffered by the pharmaceutical industry during the drug discovery and design process, despite the advancements in related technology. One major challenge in drug discovery is that pharmaceutical research often requires the processing of huge amounts of structurally complex and unrelated molecular data. The individual assessment of compounds is therefore virtually impossible for the purpose of scrutinising possible lead candidates among a pool of millions of molecules. With the constant new discoveries of compounds and the availability of both commercial and free databases of compounds, the selection of potential molecule candidates continues to pose a challenge (Walters, Stahl, & Murcko, 1998). Important drug discoveries were, and still are, key components in pharmaceutical research as such processes may impact on the long lifecycle usually imposed in pharmaceutical research studies. This process is highlighted in Figure 1.1 below.

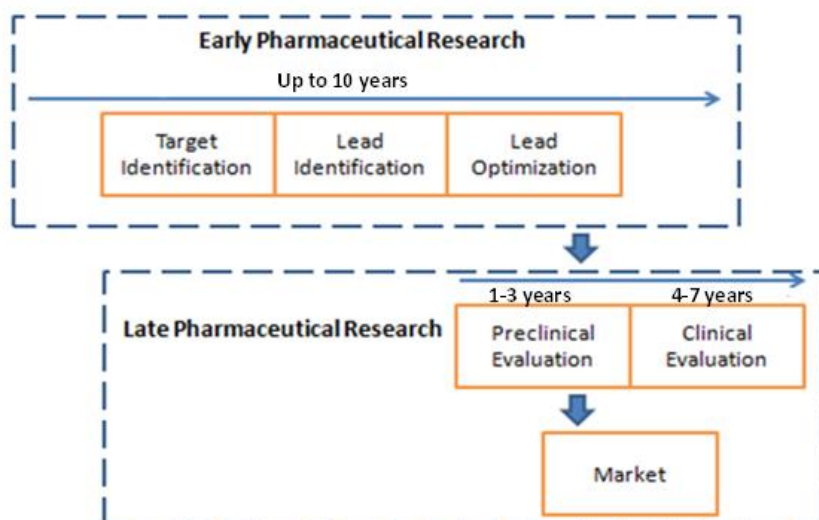


Figure 1.1: Trends in pharmaceutical research for drug discovery and design (after Terstappen and Reggiani, 2001)

The first step in the drug discovery cycle is typically target identification. This step essentially involves identifying a particular protein target from a particular disease from genetic information. The next step is the identification of suitable compounds that react with the particular protein target. Such compounds are otherwise known as lead compounds. Out of these one or several potential compounds are chosen based on a number of measurable properties and analyses. These include structural activity relationship and a favourable absorption, distribution, metabolism and excretion (ADME) profile. The lead compounds require further optimisation to increase the biological activity and to improve the effectiveness of the compounds against the target. This process is also known as lead optimisation and typically lasts around 12 to 18 months. The later stages of pharmaceutical research involve many more clinical evaluations and trials before the compound can be marketed. There is always a high risk that the results will be unsuitable for further development, from the initial stages of drug discovery. In this case, chemical compounds may seem to be promising drug candidates during the initial level of screening. Subsequently, failing during tests in the expensive pre-clinical and clinical stages where they may often prove to be unsuitable for further development (Bleicher et al., 2003).

Major hurdles in drug discovery and design include the crucial pre-clinical stages. At this stage, it is necessary to identify suitable or less risky lead compounds. Accurate assessment of

the safety and toxicity of compounds is stressed during this stage through rigorous experiments. Risks associated with the clinical stages include the impact of clinical trial transitions from animal tests to live human testing. There is also a significant impact from a business perspective: investments of pharmaceutical companies often range from hundreds to billions of US dollars. This high level of investment bringing increased pressure for experimental success. The transitional stages from a target discovery to a lead discovery are important as only a minimal number of potential compounds chosen from a large pool are carried through to clinical testing. There is only a small margin for error, as the whole drug discovery and design process will need to be repeated or even abandoned if the potential compounds do not work. This process can also be extremely time-consuming. The development and marketing of a single drug can take a long time, typically between 10 and 15 years (Lindsley, 2014). All of these risks can potentially be reduced very early in the drug discovery process through the accurate identification of promising lead compounds.

1.2 Chemoinformatics and the use of machine learning methods

All of these risks can potentially be reduced very early in the drug discovery process through the accurate identification of promising lead compounds. Today, chemical databases contain many millions of structures available for synthesis. Computational methods which incorporate informational techniques are therefore frequently used to improve the efficiency of the screening procedure, otherwise referred to as the field of chemoinformatics. One of the first scholars to define chemoinformatics was Brown (1998), who stated:

“Chemoinformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and optimisation.”

In the search of suitable drug candidates, a large numbers of compounds are evaluated in order to find molecules that are effective against the biological target. This screening process is known as high throughput screening (HTS), which is an approach to target validation. It allows the assaying of very large numbers of potential compounds against a chosen set of defined targets using an *in vitro* technique. This involves controlled environment and equipment. The most potent compounds obtained are called hits. HTS can rapidly select those substances that affect the target; however, it is expensive, requires extensive skills and HTS data are typically noisy or false positives. Much effort has been invested in the use of

computational methods to increase the speed and reduce the cost of lead discovery. The range of techniques developed for this purpose is generally referred as virtual screening (VS). VS involves the computational filtering of a large body of molecules (e.g., comprising a company's corporate database) to identify those that have a high probability of activity in a biological test system of interest. Thus, the VS method takes as its input all of those molecules that might be acquired (or synthesised) and tested. It then gives as its output the few molecules that should actually be tested. VS methods are increasingly being used to increase the cost-effectiveness of drug discovery programmes (Klebe, 2000).

Virtual screening is an *in silico* analogue of HTS which aims to identify which compound to synthesise or to purchase and to select compounds for an *in vitro* experiment. VS can help to analyse the results of an HTS run by identifying false positives hits. Over the past years, VS has become an essential companion of HTS as they are complementary to each other to support the lead discovery process. One popular approach to virtual screening involves the use of algorithms from the area of computer science which are referred to machine learning. A machine-learning method takes as its input a training set of compounds that have previously been classified as active or inactive. These are then analysed to develop a model that can be used to classify new molecules into active or inactive classes.

The earliest example of a machine learning method in chemoinformatics is considered to be the Substructure Analysis (SSA), while popular ones used today is Support Vector Machines (SVMs), Random Forest (RF) and Artificial Neural Networks (ANN). SSA, in particular, was proposed by Cramer in the 1970s. It was based purely on the identification of suitable lead compounds based on the relationship between molecular activity and fragment structure. SSA uses this relationship to extract a fragment weighting scheme, which is applied to the compounds for scoring, ranking and finally for assessment. The main criterion of the assessment is based on the number of active molecules that occur in the top of the ranking. The SSA method has been further developed through the introduction of various weighting schemes. It has not, however, progressed as much as other machine learning methods in recent times. SSA is very closely related to the Naïve Bayesian Classifier (NBC), a machine learning method that has become very popular in the last few years with its availability in the Pipeline Pilot system (Hert et al., 2006). NBC is a simple classification algorithm that is based on the use of Bayes' theorem and on strong assumptions as to the statistical independence of the descriptors characterising the objects that are to be classified. Here, it is

argued that the SSA is still a unique method to identify potential compounds based on its simplicity. It is a simple yet powerful method of quantifying a compound's influence purely on the basis of the fragment properties (via weighting schemes). As it has never been attempted before, the question remains as to whether there can be any possibility of improving the SSA method and its weighting scheme definition through the use of more robust and evolutionary approaches, specifically the GA (genetic algorithm) and GP (genetic programming). Both of these have reported success in various fields of application. The other question is whether both approaches, which are stochastic in nature, can be enhanced through the use of the use of a data fusion method. Data fusion is a deterministic method to produce a single, unified outcome.

1.3 Research aim and objectives

The aim of this study is to develop new weighting schemes in SSA that might have a better level of prediction performance than the existing procedures, based on evolutionary approaches. This application may enhance the cost-effectiveness of research programmes seeking to identify novel bioactive molecules. In order to achieve the above aim, several research objectives have been identified which need to be explored as summarised below:

- i) The first research objective is to quantify the level of performance of all of the existing SSA weighting schemes, which have been introduced by various researchers, and establish the best overall scheme.
- ii) The second objective is to assess the use of GA to determine the fragment weighting scheme to be used in the SSA and whether it can provide an upper-bound to the performance of the SSA when compared to the existing weighting schemes. The level of improvement will be quantified, if any uplift exists.
- iii) The third objective is to investigate the use of GP to determine a fragment weighting scheme to be used in the SSA. It is thus necessary to determine whether it can provide an upper-bound to the performance of the SSA when compared to both the existing weighting schemes and the GA-based scheme. The level of improvement will be quantified, if any uplift exists.
- iv) The final objective is to investigate and assess the use of data fusion, a technique that combines multiple ranking to provide further enhancement to the GA and GP-based SSA.

1.4 Thesis outline

This chapter describes a general introduction to the concept of chemoinformatics. This includes its application in the real world and the discussion on the virtual screening research background. This research focuses on ligand-based virtual screening, specifically on SSA. Based on the research aim and objectives defined in Section 1.3, the study is organised into eight chapters.

Chapter 2 of this thesis begins with an overview of the main elements of extracting information from a chemical structure. Commonly used methods for molecular descriptors and molecular representation are also included. Chapter 2 also reviews the basic principles and the development of the SSA method used in this thesis. Chapter 3 discusses the experimental design used in this study. It includes the details of the databases, the evaluation methods and the statistical analyses adopted to evaluate the results of the experiments conducted in this study. Chapter 4 reports on the comparison and the evaluation of the existing SSA weighting schemes. In total, ten published SSA weighting schemes are analysed. Chapter 5 investigates the use of the GA approach to weighting schemes to identify whether GA-based weighting determination yields similar or improved results when compared to the existing SSA weighting schemes. The GA approach is based on its direct determination of fragment weights as opposed to the pre-existing SSA weighting schemes. Thus, the feasibility of such a method is compared against existing weighting schemes in terms of an improvement in predictive performance. Chapter 6 examines the use of GP to develop new weighting schemes, taking as a starting point the pool of possible variables and the pool of simple arithmetic operators. The new weights resulting from the GP are then evaluated. Chapter 7 investigates the use of the data fusion method to combine the retrieval results from the multiple GA and GP searches, in which eight fusion methods are applied and assessed. Chapter 8 draws this work to a conclusion, with suggestions for more investigation into and further improvement of the SSA.

Chapter 2

Virtual Screening and Substructural Analysis

2.1 Introduction

The evolution of chemical computing has given rise to several established chemical information storing methods, and subsequently machine-led screening processes. Central to such methods are the extraction, analysis and manipulation of fragments from compounds. These methods are performed *in silico* to guide the initial discovery stage of the drug design processes. It is therefore necessary to consider potential chemical-based attributes, such as the activity and relevance of the molecules in question. Theoretically, this should increase the likelihood of discovering novel compounds for further lead verification and optimisation. This chapter describes the methods used to store and retrieve chemical structures. In addition, there are two principal components that will affect the performance of substructure analysis (SSA). These are the representation used to characterise the compounds, and the SSA weighting functions used to compare them.

2.2 Representation of chemical structures

The search for desirable compounds in existing databases is largely influenced by the clarity and accessibility of molecular information to chemists in the chemical search space. This has led to the introduction of various formats of chemical representations. Such forms of compound representations may range from simple to complex, depending on the requirements imposed on such representations. Two common forms of presentation are connection tables and line notations.

2.2.1 Connection tables

Connection tables are effective in documenting atoms and the bonds between them, which are not readily available in other types of 1D- or 2D-based structural representations. Figure 2.1 below shows an example of a connection table for the compound Aspirin. The table is primarily formed by capturing the spatial coordinates of each atom by definition of its x , y , and (perhaps) z coordinates of the atom, and their associated bonds. A connection table also provides bond information for each atom forming the compound.

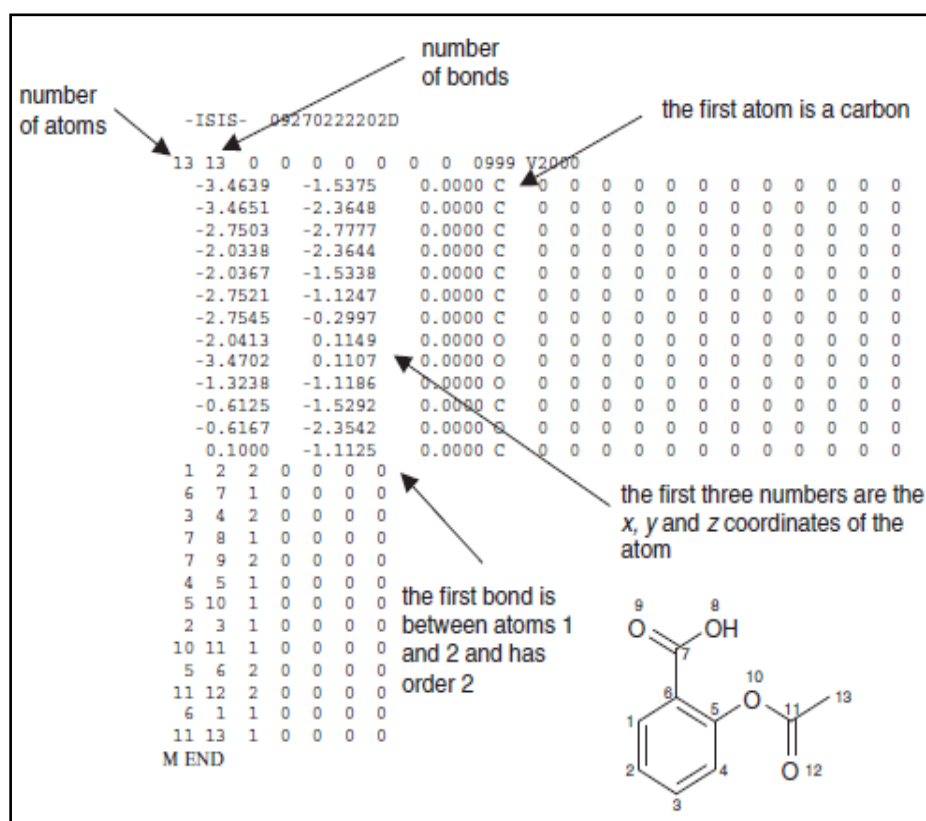


Figure 2.1: Connection table representation (after Leach & Gillet, 2007). Information such as atom coordinates relating to its formation and bonding pairs are stored for accurate rebuilding of the molecules

The connection table shown above is the simplest form of representation. More detailed forms may include further additional information, such as the hybridisation states of the atoms and the bond orders (Leach & Gillet, 2007). Hydrogen atoms are usually not included in the majority of standard connection tables, but this is also the case for many other structural notation systems (such as in SMILES, as will be discussed below). Information regarding the (xy) or (xyz) coordinates of the atoms enables standard chemical drawings to be produced for use in a molecular graphics program or any 3D molecular manipulation / analysis methods.

2.2.2 Morgan algorithm

The atoms of chemical compounds originally are not denoted with descriptive labels and the numbering system may be arbitrary. Issues arise when many different labels for the atoms can be represented for the same chemical compound. The Morgan algorithm can be used to solve this problem by providing a canonical label or a unique number for an atom. Canonicalisation is an important concept in chemical representation, as it allows the representation of a

particular compound to be unique and unambiguous. The Morgan algorithm uses iterative calculations of the connectivity values to differentiate atoms, as illustrated in Figure 2.3.

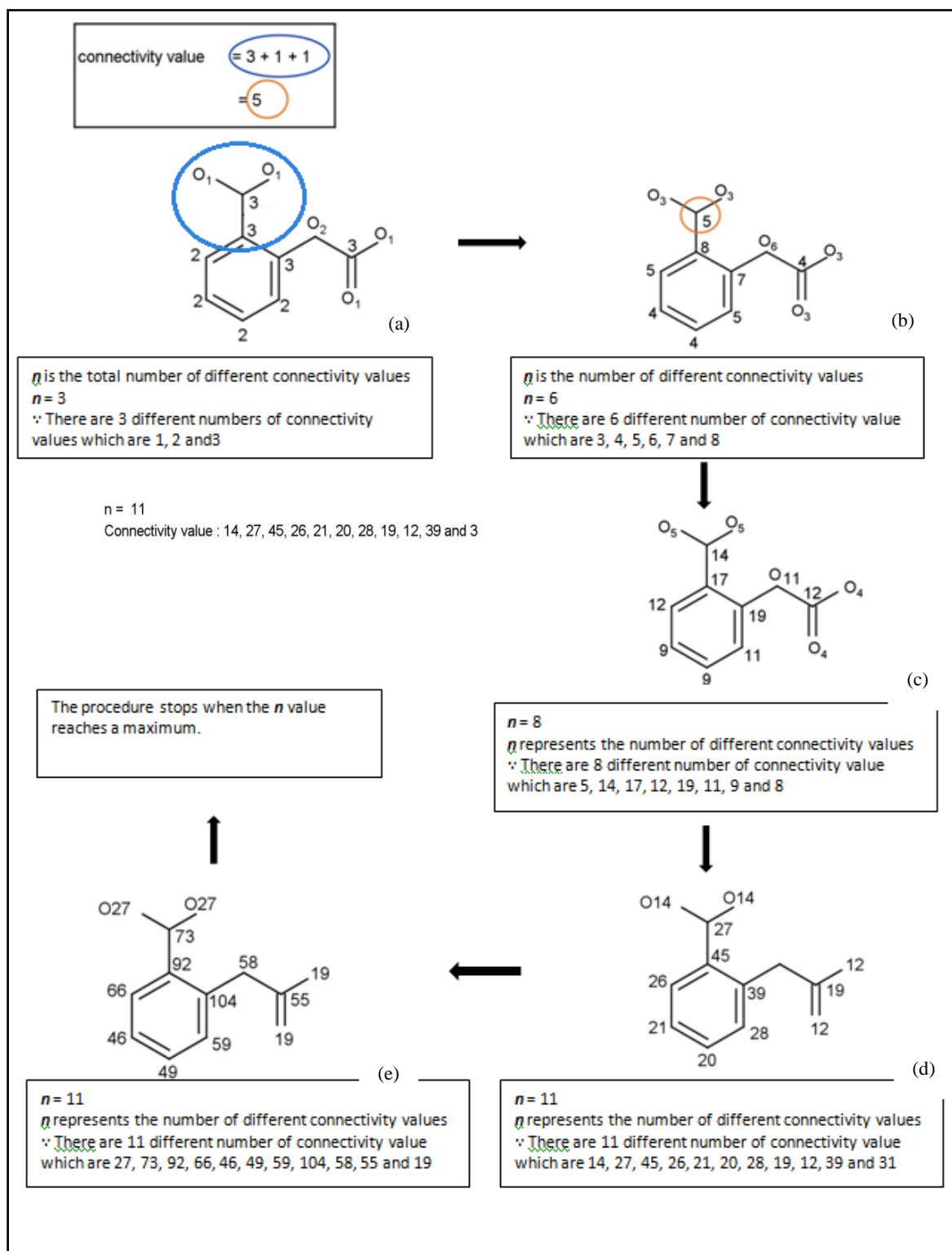


Figure 2.2: Morgan algorithm procedure (after Leach & Gillet, 2007)

From Figure 2.3, a compound is first defined as a series of connected atoms, where the connectivity value of each atom is calculated from the number of connections to other atoms. The next iteration seeks to update the connectivity value of the atom based on the initial values of the neighbouring atoms (Figures 2.3(b) and (c)). For example, the orange highlighted atom in Figure 2.3(b) has its connectivity value updated to 5, as a summation of the connectivity values of its neighbours, which are 1, 1 and 3 respectively. The iteration continues for each atom until a maximum connectivity value is reached, as shown in Figure 2.3(e). Based on the finalised connectivity values, the connection table uses atoms with the highest connectivity value as the first atom for definition, and subsequently other atoms in decreasing order of their connectivity values (Leach & Gillet, 2007).

2.2.3 Line notation

Line notation is a compact, alphanumeric representation of molecules. It is more compressed than connection tables. It is thus able to encode a large number of molecular structures while requiring only a small memory space for storage. It is widely used for storing, representing, communicating and checking the identity of chemical structures. It can encode structures in compact form, and this may be human-readable and writable. It can also be easily used with respective software, and it also provides a canonical representation. Line notation is of particular importance in chemoinformatics which include Wiswesser Line Notation (WLN), Simplified Molecular Input Line Entry System (SMILES) and InChI (Willett, 2009). Linear notation represents the complete constitution and connectivity of chemical compounds as a linear sequence of character.

A given chemical structure can have many valid and unambiguous representations. A molecule may be presented in the form of a different numbering system for the atoms in connection table. It would be useful to use a standard numbering system to derive a single unique representation. The process of converting an input representation to a canonical form is called canonicalisation. Many methods have been developed for a unique and unambiguous numbering of the atoms of a molecule. The canonicalisation process involves deriving a canonical code for numbering or labelling the atoms in a unique and reproducible way (Gasteiger & Engel, 2006). Canonical-based representations introduced with WLN, SMILES and InChI will be discussed in greater later in this chapter.

2.2.4 Wiswesser Line Notation (WLN)

Wiswesser Line Notation (WLN) was developed by William Wiswesser in the 1950s. It was considered one of the most popular and widely used notations to enforce canonical representation of molecules, whereby each molecule has one and only one formula. Figure 2.2 below stresses the exclusive relationship between the computer representation and the molecular structure. ‘*Unique*’ refers to only one instance of computer representation derived from a structure; ‘*Unambiguous*’ refers to only one structure produced from one computer representation.

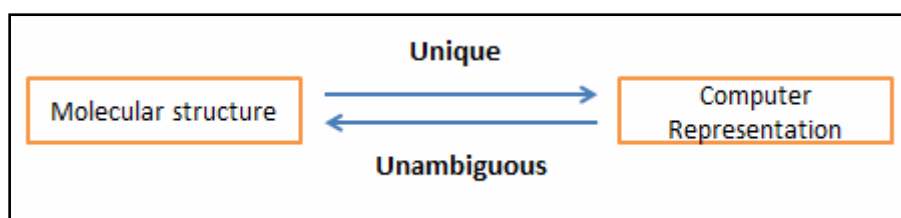


Figure 2.3: Canonical state of structure for computer representation (adapted from UTM open courseware chemistry website)

WLN represents structural formulae with a short combination of numerals, capital letters and punctuation marks. It also makes extensive use of special symbols to denote common structural fragments. These symbols are formed into a specific code by encoding them in the same order in which the fragments are connected in the structural formula (Vollmer, 1983). A set of rules is devised to ensure that such notations enforce a canonical form, as shown in Figure 2.2. Although WLN has been widely used to represent structures in the form of line notation, it was difficult to adopt and maintain. This is mainly because many rules must be followed to generate the correct notation of a complex structure (Weininger, 1988). It thus proved particularly difficult to implement notation in computer terms; consequently, this led to the introduction of alternative notation systems instead.

2.2.5 Simplified Molecular Input Line Entry System (SMILES)

SMILES is one of the most popular line notations in current use. It was created in response to the need for a simpler, more computer accessible and human-friendly notation than WLN. SMILES retains the concept of canonical representation, but it is easier to encode with a computer than was the case with WLN. The original SMILES specification was developed by Arthur Weininger and David Weininger in the late 1980s. It has since been modified and extended by numerous other organisations (Weininger, 1988).




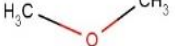
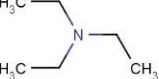
In 2007, an open standard called OpenSMILES was developed by the Blue Obelisk open source chemistry community (<http://www.opensmiles.org>). In SMILES, atoms are represented using their respective atomic symbol. Upper case letters refer to aliphatic atoms; lower case letters refer to aromatic atoms. If the atomic symbol has more than one letter, the second letter must be in lower case. The structures entered using SMILES are hydrogen-suppressed, which means that the molecules are represented without hydrogen atoms. SMILES is able to handle branches denoted with the symbol () and also supports nesting. In terms of bonding, the following is used: '-' for a single bond (which are usually not shown at all); '=' for double bonds; '#' for triple bonds and ':' for aromatic bonds. Single and aromatic bonds may always be omitted. Many weak points of other line notation systems can be attributed to the overuse of symbols and hierarchical rules based on the length of the final notation. SMILES notation was designed to introduce a standard, unified description of chemical structures so that it can be used and understood by both users and computers. For example, a chemist is able to retrieve a list of compounds from a database defined in terms of coded strings as notated by the SMILES. The selected compounds in SMILES notation can be further utilised with a computer to identify structural descriptors as necessary. Likewise, chemical information can be exchanged universally with ease between researchers by utilising the standard unique SMILES string to name a molecule. Chemical programmers also use SMILES in the process of entering chemical data into a computerised database, and thus, maintaining unique structural descriptions is necessary. The description used by computers requires a far more complex set of rules and hierarchies, and this translates to an extensive dependency on an efficient computer algorithm (Weininger, 1988).

2.2.6 International Chemical Identifier (InChI)

InChI is a recent notation system introduced to standardise chemical structure information. InChI was a joint project by many organisations, such as standards agencies, chemical field experts, and educational and commercial based participants. The goal was to achieve a more consistent and standardised definition of chemical structures. InChI uses a layered format to represent all the available structural information relevant to compound identity, where by each layer designates a specific type of structural information, with the layers ordered to provide successive structural refinement. There are six major InChI layer types, with each giving a different class providing structural information. The InChI layers consist of the main layer, a charge layer, a stereochemical layer, an isotopic layer, a fixed-H layer and a reconnected layer. The main layer, which specifies chemical formula, atoms, and bonds between them, is

required for all InChIs. However, the other layers appear only when corresponding input information is provided (Heller, McNaught, Stein, Tchekhovskoi & Pletnev, 2013). Every InChI starts with the string "InChI=" followed by the version number, currently 1. This is followed by the letter S for standard InChIs. Layers and sub-layers start with "/" (forward slash) followed by a letter denoting the identity of the layer (except for the chemical formula layer). InChI notations are meant to be processed and decoded by computers only: they are not designed to be interpreted by users. Comparisons between SMILES and InChI have been drawn in various studies (Boda, 2010), where the former has more flexibility and the latter is more consistent in terms of representations. SMILES is proprietary and has more software support. As O'Boyle (2012) states: "other commercial and open-source software developed their own algorithms for generating canonical SMILES all of which differed from each other and none of which are published" (O'Boyle). This has led to the use of different generation algorithms, and thus, different SMILES versions of the same compound have been found. The lack of a single commonly-adopted standard has resulted in inconsistent representation terms. To address the lack of a non-proprietary, strictly-unique standard chemical identifier, the InChI project was initiated. InChI is non-proprietary, open-source, and freely available to the scientific community. As the software for generating InChI strings is freely available, it also avoids the interoperability issue. SMILES, however, is generally considered to be more human-readable than InChI. Table 2.1 shows the comparison between the SMILES and InChI notations.

Table 2.1: Linear chemical notation

Bonds	IUPAC Name	Chemical Compound	Chemical Formula	SMILES	IUPAC Standard InChi
Single Bond	Ethane		C ₂ H ₆	CC	InChI=1S/C2H6/c1-2/h1-2H3
Double Bond	Ethene		H ₂ C=CH ₂	C=C	InChI=1/C2H4/c1-2/h1-2H2
Triple Bond	Hydrogen Cyanide		HCN	C#N	InChI=1S/CHN/c1-2/h1H
Aromatic Bond	Dimethyl Ether		CH ₃ OCH ₃	COC	InChI=1S/C2H6O/c1-3-2/h1-2H3
Branches	Triethylamine		N(CH ₂ CH ₃) ₃	CCN(CC)CC	InChI=1S/C6H15N/c1-4-7(5-2)6-3/h4-6H2,1-3H3

2.3 Molecular descriptors

The manipulation and analysis of chemical structural information often requires the use of molecular descriptors generated from the molecule representations described above. The molecular descriptor can characterise and classify structural patterns by means of encoding, using numerical values to characterise the properties, e.g. the physicochemical properties of molecules (Todeschini & Consonni, 2008). A descriptor classification essentially requires two criteria: the molecular representation of the compound, and the algorithm used to calculate the descriptor information. These descriptors may contain detailed information regarding various properties useful for analysis within Virtual Screening, such as the Structure-Activity-Relationship (SAR), Quantitative-SAR (QSAR) studies and molecular diversity analysis. It is reported that there are more than 3,000 different types of molecular descriptors available (Leszczynski & Shukla, 2009). For any typical drug discovery process, however, only descriptors that successfully correlate structural features with the biological activity of interest are explored (Khanna & Ranganathan, 2011).

There are several ways to classify descriptors, categorised as physicochemical (hydrophobic, steric or electronic), structural (frequency of occurrence of a substructure), topological,

electronic (molecular orbital calculation) or geometric (molecular surface area calculation). Molecular descriptors (specifically, physicochemical descriptors) can also be categorised into two groups. The first is the descriptors representing properties of molecules, for instance, the logP and the molecular weight. Physicochemical parameters govern the physical and chemical properties of chemical entities. The behaviour of bioactive chemical entities may be affected by changes in physicochemical properties. Physicochemical properties are also known as an example of 1D descriptors. Examples of properties include molecular weight; logP (the logarithm of the partition coefficient between octanol and water) which quantifies a molecular hydrophobicity, solubility and simple count of features (e.g. number of atoms, H-bond donors, H-bond acceptors and ring systems). All these properties can be used as part of QSAR studies, which correlates quantitative chemical structure attributes (e.g. physicochemical, biological and toxicological) of molecular descriptors to a biological activity. For instance, Lipinski et al. (2012) in their studies of solubility and permeability prediction in drug discovery used four physicochemical properties. These were molecular weight, sum of nitrogen, oxygen and hydrogen-bond acceptors.

The second type is descriptors categorised according to dimensionality, which are 2D, and 3D. 2D descriptors are derived from molecular connectivity table and apply a simple count of features to characterise molecules. Examples of 2D descriptors calculated from 2D graphs are topological indexes and 2D fingerprints.

3D descriptors are generated from 3D connection tables, which can be obtained either experimentally or theoretically. For example, a 3D structure builder such as CONCORD or CORINA (Clark, 1999) can generate 3D structures from the chemical graph. Examples of descriptors requiring 3D representations are the pharmacophore descriptors, affinity fingerprints, distance-based descriptor, 3D atom environment for use in atom mapping similarity searching and 3D molecular fields for use in field-based similarity searching. 3D molecular field descriptors involve generally a 3D grid, each element of which is characterised by the values/properties of the steric, electrostatic and hydrophobic. It should be noted that these 3D properties are used in QSAR methods, such as Comparative Molecular Field Analysis (COMFA), where it is one of the most significant development in QSAR. This research, however, focuses only on 2D descriptors, which are considered more appropriate to the aims of this research, namely, the SSA. There are also several descriptors, which are based

on 3D molecular fields that have been used to predict ADMET properties, e.g. Volsurf descriptors developed by Cruciani, Crivori, Carrupt and Testa (2000).

2.4 2D descriptors

2D descriptors are widely used in chemoinformatics in terms of calculation, storage and interpretation, and extensively in VS methods. This may include similarity searching, where a simple count of shared features, such as common fragment substructures, can be a measure of chemical distance when used with similarity coefficients. In addition, they are commonly used because of their simplicity: they are easy to analyse and calculate (Leach & Gillet, 2007). Two important types of 2D descriptors are discussed here: topological indices and fragment-based descriptors.

2.4.1 Topological indices

Topological indices are single-valued descriptors derived from a 2D graph of molecules. They are presented as real numbers, characterising structures according to properties such as size, overall shape and atom connectivity. In other words, the index encodes the information of the molecule, rather than the presence of particular fragments in a given compound. There are many examples showing the use of topological indices applied as descriptors. One is the Wiener Index, which computes the number of bonds and the distance between pairs of atoms (Wiener, 1947). Another example is the Randić molecular connectivity index, also known as the branching index. This index is the accumulation of the connectivity of bonds over all bonds in the molecule (Randic, 1975). A further example is the family of chi molecular indices. To calculate the index, valence values are introduced to encode the counts of pi, sigma and lone pair electrons for each atom (Leach & Gillet, 2007). Balaban's J index (Balaban, 1982) is also an example of a topological index.

2.4.2 Fragment-based descriptors

Fragment-based descriptors are also called 2D fingerprints or fragment bit-strings. They consist of a vector of bits (each bit position corresponding to one or more specific substructural fragments). The descriptors encode the presence or absence of a particular substructure or fragment in a compound. They encode the presence ('1') or absence ('0') of a fragment or substructure in a molecule as illustrated in Figure 2.4. There are two basic types of fragment-based 2D fingerprints widely used in VS: (1) Fragment Dictionary and (2) Hashed Fragments.

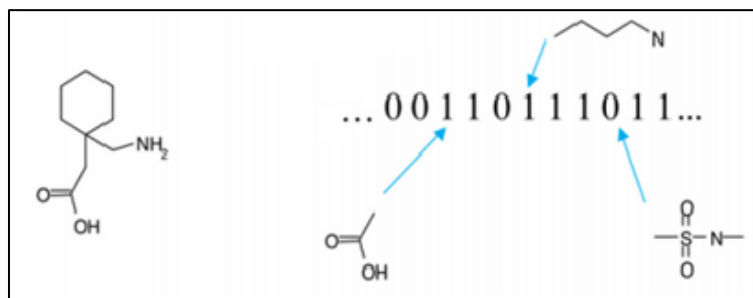


Figure 2.4: 2D chemical structure (left) and structural key fingerprint (right)

Fragment dictionaries

Fragment dictionaries are also known as structural keys which were, historically, the first type of screen employed in chemical database searching (Willett, 1987). They are usually represented as a bit-string, each entry of which represents the presence or absence of a specific 2D fragment. Structural keys rely on the use of list of fragments selected to be important for the medicinal chemists, i.e., the predefined fragment dictionary. Thus, it imposes several limitations. The main weakness of this approach is that the fingerprints only contain a limited number of keys, i.e., those present in the dictionary. The predefined dictionary will later introduce a ‘library bias’ problem, where only fragments contained in the library will be considered for the contribution in the similarity search. Another limitation of a fragment dictionary is that the relevance of the fragments depends on the database considered. If the choice is ineffective, it will cause many defined fragments in the library that might not be useful in a given problem.

Several studies have been conducted to approach these limitations. For example, Adamson et al. (1973) introduced a method to select the best structural keys for screening in a huge chemical database. The selection was made according to the disparate frequencies of many types of fragments in the database. These types are classified based on the structural characteristics of compounds of the Chemical Abstract Service Registry System. Hodes later implemented a modification of Adamson’s approach by focusing on the automated fragment generation for screening discrimination (Hodes, 1976). Barnard Chemical Information (BCI) later created the BCI fragment dictionary, used much later, as the basis to generate fragment-based fingerprints (Barnard & Downs, 1997). Predefined dictionaries of structural fragments such as MDL MACCS keys are used to identify features contained in molecules. These are based on the pattern matching of chemical compound structures (Durant et al., 2002). The

descriptors consist of 166 structural fragments and each fragment has a key, represented in a fixed position in the bit-string. Chemists have recognised that these fragments are important for the bioactivity of chemical compounds. The structural fragments of features in a given molecule are flagged as 1 or 0, so as to replicate the nature of binary format. The latter, as a whole, forms a bit-string, as determined by the elements of the dictionary. Limitations were observed, however, in that bits may be set only once, irrespective of the frequency of occurrence of the given key.

Circular substructure fingerprints

Circular substructure fingerprints are a popular subclass of molecular fingerprints and have been successfully applied in the Pipeline Pilot software, also known as the Extended Connectivity Fingerprint (ECFP). Pipeline Pilot implemented the ECFP algorithm to generate these fingerprints. They encode the central atom and the neighbouring atoms within a diameter of 2 (i.e., ECFP(C)_2), 4 (i.e., ECFP(C)_4) or 6 (i.e., ECFP(C)_6) atoms. The ECFP algorithm was derived from the Morgan algorithm. A study reported on the combination of ECFPs and Functional Connectivity Fingerprints (FCFPs) which is found to be more effective than hashing, dictionaries or topological based fingerprints (Hert, 2004).

In generating the fingerprint, there are three stages that will be carried out sequentially. First, a numeric identifier (NID) is assigned to each atom. Second, iteration is performed to update the NID and to reflect the NID of the neighbours of each atom in a given diameter. All of the iterations can encode a list which consists of integer(s) that are calculated by a suitable hashing function. Finally, multiple occurrences of the same feature are removed. Several studies have been carried out in order to evaluate ECFP in analysing the output of High-Throughput Screening (HTS).

Rogers and Hahn (2010) used a modified Bayesian model to determine whether compounds exhibited false positive or false negative hits in HTS data consisting of more than 50,000 compounds. Both Hu et al. (2009) and Glen et al. (2006) applied ECFP in ligand-based virtual screening methods for the classification of active and inactive molecules. From these studies, it was concluded that circular fingerprints are one of the most effective representations to use as a search tool. It is considered fast and efficient method for detecting the presence and absence of fragments, and has obtained very good performances in chemoinformatics (Rogers

& Hahn, 2010). ECFP can be applied in similarity searching, clustering, and virtual screening, in a similar way to other types of fingerprints.

Hashed fingerprints

Hashed fingerprints are an alternative fragment-encoding procedure consisting of indexing all the substructural patterns present in a particular molecule. It is interesting to note that some consider the circular fingerprints to be arguably a form of hashed fingerprint. In hashed fingerprint, all the patterns are hashed using a hashing function to fit into the length of the bit-string. This approach allows for more generalisations because it does not depend on a predefined list of structural fragments. Instead of using a fragment dictionary, the hashed method defines a set of patterns to index. For every molecule, each distinct fragment will be mapped to a set of numbers using a pseudo-random number generator. All the representative numbers produced for one molecule will be combined using a hashing function to produce a bit-string. The corresponding bit positions in the bit-string are subsequently set to 'on'. In this approach, a given pattern always results in the same set of numbers.

This approach is used in the Daylight Chemical Information Systems (James & Weininger, 2006) and Tripos systems (Tripos Inc., 2010). Daylight fingerprints report all path lengths of a molecule up to a certain value (by default 7 bonds). The Unity System only considers patterns up to 6 bonds long and uses a combination of structure keys and hashed fingerprints to generate 988 bit fingerprints (Khanna & Ranganathan, 2011). Once the numbers for all patterns have been generated, the fingerprint is generally folded to obtain a fixed-length bit-string.

The folding method was introduced to optimise the number of bits in the hashed fingerprints and to increase the searching speed. This method is applied by dividing the obtained fingerprint into two equal parts of the bit string and combines the two parts by using the logical OR. This produces a high density of information with fewer bits (Todeschini & Consonni, 2008). Folding has several limitations as the common bits in two fingerprints can be set by unrelated fragments. These bits then lose their specificity and can potentially become irrelevant, which is otherwise known as bit collision.

Another widely used example of hashed fingerprints, which includes frequency information, is the molecular hologram invented by Tripos (Tripos Inc, 2003). Molecular Holograms

encode the number of times each fragment appears in a molecule. Instead of using a binary bit ('1' or '0') to represent the existence of each fragment, the hologram uses integers to represent the total number of occurrences of each fragment.

2.5 Searching databases of molecules

Chemical structure databases containing computerised representations of traditional chemical structure diagrams have been used to support various tasks in chemical research and development. In chemoinformatics, searching for a molecule in a molecular database using a query structure is common. It involves the use of graph matching algorithms by comparing the connection tables. This requires a large number of tests of all possible mappings of the query structure to each molecule in a database. Contemporary chemical databases can be very large. For example, the ChEMBL Open Data database contains over 1 million compounds (Gaulton et al., 2012) while Chemical Abstracts Service (CAS) reported over 100 million chemical substances to date (CAS, 2016). Thus, an effective approach is important as it enhances search efficiency in a appropriate time frame. Early, there were two types of searching database mechanisms, namely: structure searching and substructure searching. These mechanisms were later complemented by another mechanism which is similarity searching.

2.5.1 Structure searching

The chemical compounds stored in a machine-readable form allow analysts to perform direct searches for a particular molecule or molecules. The earliest and simplest type of search mechanism is structure searching, which checks for the presence or absence of a specific molecule in a database (Willett, 2009). This is very simple to implement if a canonical character-string representation of a molecule is available, such as the Wiswesser Line Notation. Structure searches of connection-table files were, however, more problematic until subsequent development by Morgan at the Chemical Abstract Service (henceforth 'CAS') of a simple canonicalisation procedure. This produced a canonical numbering of a set of atoms. Another effective way to implement structure searching is by generating a hash code from the connection tables. The hash code of the query is compared to the hash code of the compounds in the database (Maurer & Lewis, 1975, Sheridan, 2002). Consequently, the structures that match the query code can be evaluated using isomorphism algorithms, as shown in Figure 2.5. The hash code of the query is first compared to the hash code of the compounds in the database and only the structures that match the code of the query are evaluated using an isomorphism algorithm (Bawden et al., 1981).

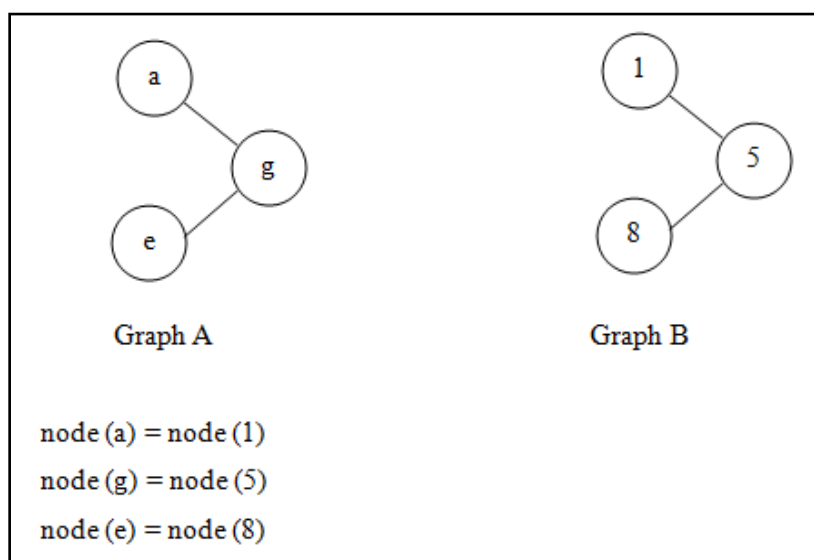


Figure 2.5: Graph A and B are isomorphic and topologically identical (after Yirka, 2015)

2.5.2 Substructure searching

Unlike structure searching, which only targets a whole molecular structure, substructure searching seeks to determine the presence of a partial structure in a complete molecule. The graph represents the molecule. The atoms of one molecule are the nodes of the graph and the covalent bonds between the atoms are the edges. There are two main procedures employed in substructure searching. Screening is firstly carried out to discard non-matching molecules with the query structure. An atom-by-atom search then follows, which further analyses all the available molecules using subgraph algorithms. Subgraph algorithm for substructure searching was introduced almost half a century ago (Ray & Kirch, 1957). The algorithm performed subgraph isomorphism procedures that exhaustively searched a connection table for the presence of the query pattern. It was largely inefficient, however, and often led to subgraph isomorphism problems. A problem arose in determining whether the input target (T) graph contained a subgraph that was isomorphic to the input query (Q) graph.

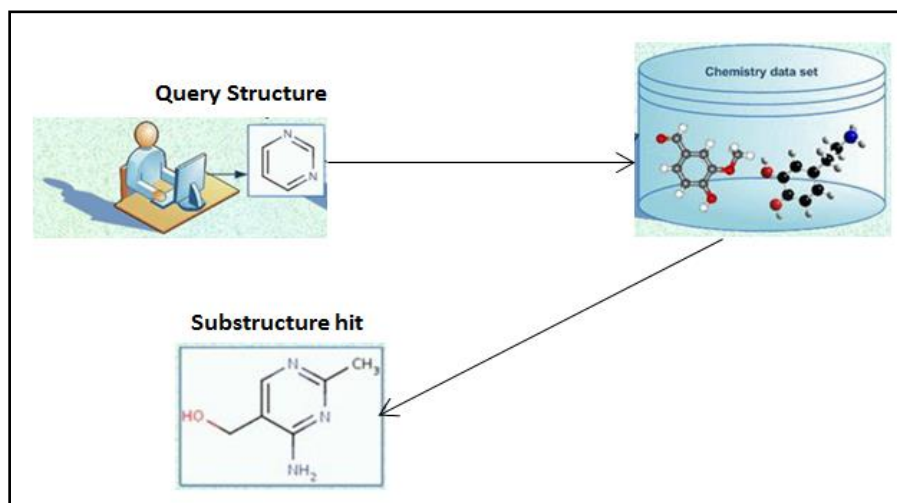


Figure 2.6: Schema of chemical substructure searches (after Horai et al., 2010)

Figure 2.6 shows the fundamental schematics of substructure searching. A query structure is relayed to a database of molecules. The substructure search yields all the molecules that contain the specified query structure and will result in a substructure hit, if found. Large scale substructure searching is relatively quicker with the use of 2D fingerprints. This method can be applied in a substructure search. Accordingly, if an *X* structure is a substructure of *Y*, then each bit set for structure *X* must also be a set for structure *Y*. The fingerprint screening can then be used to compare and eliminate molecules in the data set before invoking a subgraph isomorphism algorithm. The function of the filter is to rapidly discard those molecules which do not contain the query's substructural fragments to pass on to the time-consuming subgraph isomorphism search (Willett, 2009). A substructure search, in fact, only divides the database into two subsets: those containing the query and those that do not contain it.

2.5.3 Similarity searching

Similarity searching provides a complementary alternative technique to substructure searching. It involves comparing the query with every compound in the database. A measure of similarity is then calculated between the target structure and every database structure. Similarity measures quantify the relatedness of two molecules with a large number (or one) if their molecular descriptions are closely related and with a small number (large negative or zero) when their molecular descriptions are unrelated. There are many measures available to quantify the degree of similarity between a pair of molecules. The computational requirements

of these measures vary depending on the level of detail used to represent the molecules that are being compared.

The results of the similarity measure will be used to sort the database structures in the order of decreasing similarity with the target. The resulting ranked list of structures will then be returned to the user. Molecular similarity can be defined in many ways depending on the information used to represent the molecules and also the measures employed to quantify the degree of similarity between two molecules (Johnson and Maggiora, 1990; Dean, 1995). The first reports on similarity searches appeared in the mid-1980s, based on the work carried out at Lederle Laboratories (Carhart et al., 1985) and Pfizer (Willett & Winterman, 1986). The use of similarity calculations between molecules have since not only been used in similarity searching, but also in application like compounds selection (Bawden, 1993; Lajiness, 1997) and molecular diversity (Martin et al., 1995; Holliday, Ranade & Willett, 1995; Holliday and Willett, 1996; Gillet, Willett & Bradshaw, 1997). There are three principal tools used for the similarity measure (Willett, 2000): a structural representation used to characterise the compounds that are being compared; a similarity coefficient used to quantify the degree of resemblance between a target structure and each of the structures present in the database; and a weighting scheme used to differentiate more important from less important features in a compound.

2.6 Virtual and High-Throughput Screening

High-Throughput Screening (HTS) was first developed to assay tens to hundreds of compounds conducted using *in vitro* technique. There can now be as many as hundreds of thousands of active compounds discovered from HTS processes, which require further screening for optimal selection within 6 months to a year time frame (Oprea & Matter, 2004). Specifically, the use of combinatorial libraries of chemical compounds through HTS yields a large number of active compounds for further analysis, otherwise known as hits. In this way, the hits can be refined and analysed to determine a suitable series of structures, taking into account preferred biological and drug-like activity, known as leads.

HTS and VS are important methods in drug discovery process used to find chemical compounds that bind to the identified target. These screening methods are different in approach but have similar objectives. Chemists are given a targeted series of biological activities, constituting a set of objectives for compounds to achieve, also known as target

identification and validation, derived from the disease in focus. Figure 2.7 illustrates a typical screening process in drug discovery.

Virtual screening is increasingly influential and involves the lead identification process from large chemical compounds databases in pharmaceutical research (Reddy et al., 2007; Willett, 2005). By definition, virtual screening is the use of computer methods to perform such screening, and is capable of automatically evaluating a large library of compounds (Walters et al., 1998; Willett, 2009). The ultimate aim of virtual screening is to identify novel molecular structures which respond to, or achieve, a target set of objectives or profiles.

The HTS method did not entirely fulfil its high expectations, however, since one of the most common problems with HTS is the resulting large number of false positives. There could be a pool of highly diverse hits, and thus additional screening methods were usually required to determine an acceptable number of leads (Diller & Hobbs, 2004). Overall, this method has a number of limitations, which are that it is time-consuming, expensive and requires intensive skills. As a consequence of a large number of commercially available and synthetically accessible molecule structures, efficient algorithms for searching large datasets are becoming more vital. Bajorath points out that the effectiveness of the HTS can be enhanced by combining the HTS with VS. The VS method can be applied in parallel or prior to HTS to maximise effectiveness of the drug discovery screening process (Bajorath, 2002).

High Throughput Screening (HTS) is a drug-discovery process widely used in the pharmaceutical industry. HTS identifies lead molecules by performing individual biochemical assays with large number of compounds. The huge cost and time consumed with this technology has led to the integration of cheaper and effective computational methodology namely virtual High Throughput Screening (vHTS). vHTS is a computational screening method which is widely applied to screen *in silico* collections of compound libraries to check the binding affinity of the target receptor with the library compounds. The compounds that are predicted to bind strongly to the target are then extracted from the database for further testing (Dahlin & Walters, 2014). This is achieved by using a scoring function which computes the complementarity of the target receptor with the compounds. HTS and vHTS are complementary methods and vHTS has been shown to reduce false positives in HTS and to increase hit rates or enrich hit lists from HTS. vHTS can effectively enrich the output of HTS by removing predicted compounds that are least likely to engage the target. (Jenkins, Kao &

Shapiro, 2003). Lengauer et al. (2004) summarised “*Better drug candidates originate from better leads, and better leads will come from better hits*”.

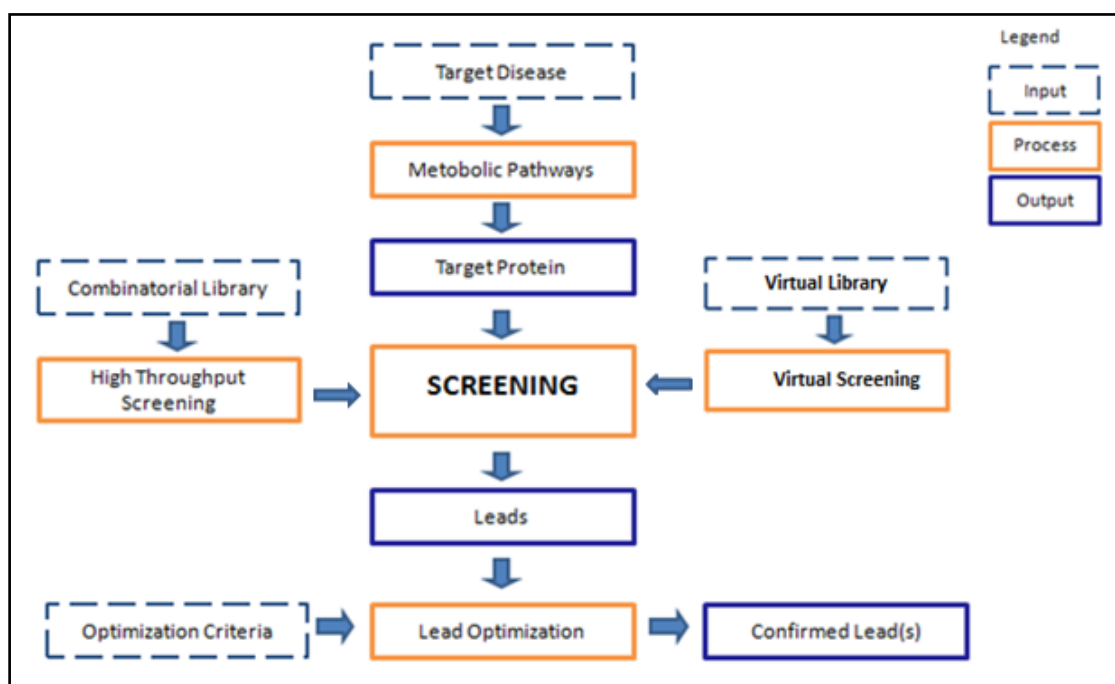


Figure 2.7: Overview of typical screening process (after Lengauer, Lemmen, Rarey, & Zimmermann, 2004)

VS methods can be classified into two main groups: ligand-based and structure-based methods. The ligand-based (LBVS) method can be employed when the 3D structure of the target is unknown, whereas the structure-based (SBVS) is applied when the 3D structure is known (Bajorath, 2001; Leach & Gillet, 2007; Willett, 2006). LBVS is essentially a method where chemical compounds (ligands) and their associated biological information serves as the primary components' starting point (Langer, Hoffmann, Bryant, & Lesur, 2009). In order to perform screening, large searchable collections of compounds are needed. These are accessed from compound databases, such as from virtual and real compounds, and ligand knowledge-bases, which include WDI (available via <http://thomsonreuters.com/en/products-services/pharma-life-sciences/life-science-research/world-drug-index.html>) and the MDL Drug Data Report which is available via <http://lifesciences.thomsonreuters.com/prouis>. Langer et al. (2009) stated that the main purpose of biological screening is to extract from a large pool of compounds, a number of high quality hits which can then be translated into candidate lead

compounds for further testing. A large screening pool is therefore advantageous to the effort of collecting a good number of hits for lead compounds identification.

When only a single active molecule is known, similarity searching can be performed, where the database will be ranked in decreasing order, according to the degree of similarity between the query and the known active structure. However, if several actives are known then pharmacophore mapping can be applied to determine the patterns of features responsible for the biological activity (Willett, 2009). The identification of a common 3D pharmacophore, followed by a 3D database search, is possible when such data are available. On the condition that a reasonable number of active and inactive structures are known, machine learning methods such as Substructural Analysis and Binary Kernel Discrimination (BKD) may be used for virtual screening. Finally, protein ligand-docking or de novo design may be applied when the 3D structure of the protein or biological target is available to enable detection of its ligand binding. A summary of the methods described above is presented in Table 2.2.

Table 2.2: Approaches in ligand-based and structure-based virtual screening

Virtual Screening Method		
Ligand-based virtual screening	Amount of structural and bioactivity data	Approach
Similarity searching	Single active known	<p>Determine the value of similarity between reference structure and each structure in a chemical database.</p> <p>Database then ranked in decreasing order. The most similar structure to the active reference structure will be listed at the top of the database.</p> <p>Based on Similar Property Principle (SPP) (Johnson & Maggiora, 1990) which suggests similar structures likely to have similar activities and properties.</p>

Pharmacophore mapping	Several actives known	To identify common features and structures by examining the interaction between receptor and ligand. The 3D database search usually employed to identify lead molecules and new classes of compounds based on desired biological activity (Güner, 2000).
Training data for machine learning system	Reasonable number of active and inactive known	Ranking of database in decreasing order, based on a set of calculated weights based on the active and inactive molecules information in the training set. The top-ranked molecules possess the highest probability of activity. Examples of such methods include Substructural Analysis, Binary Kernel Discrimination, Bayesian Inference Network, etc.
Structure-Based Virtual Screening	Amount of structural and bioactivity data	Approach
Docking study De novo design	3D structure of the protein target known	To find the molecules forming the best fit between receptor and ligand that correspond to the binding site (Halperin, Ma, Wolfson, & Nussinov, 2002).

2.7 Structure-Activity Relationship (SAR) and Quantitative Structure-Activity Relationship (QSAR)

SAR is an approach to find relationships between chemical structures and biological activity. SAR theory is based on the principle that molecules with similar structures will have similar properties (Hansch, 1969). SAR theory states that a small change in the structure of a compound is expected to have a small favourable or undesirable effect on its activity. In actual practice, however, though it holds as a general rule, this is often not the case, as even small changes in the structure may have a much greater effect than expected (Tong, Welsh, Shi,

Fang, & Perkins, 2003). The identification of a relationship between structure and activity requires that the significant structural characteristics of molecules with known activities should be encoded in a manner that can be understood, and operated on, by computational algorithms.

Several descriptors, in principle, such as structural features and characteristics which include the size, shape and geometry of chemical structures, can be used in a structure-activity relationship (Carhart, Smith, & Venkataraghavan, 1985). SAR theory considers activity as either a quantitative measure or as discrete states, such as a binary classification of active and inactive, or as a level of activity, such as weakly active or strongly active. Where quantitative measures of activity are used, a SAR is often referred to as a QSAR. Measurements include half maximal effective concentrations, for example, and the concentration of a compound that causes some specific level of effects, such as a drug or toxicant (Sherhod, 2011).

QSAR modelling techniques mainly rely on the use of molecular descriptors to perform “quantitative” analysis on such descriptors. They can yield useful physiochemical information and subsequently, a correlation between the structure and biological activity.

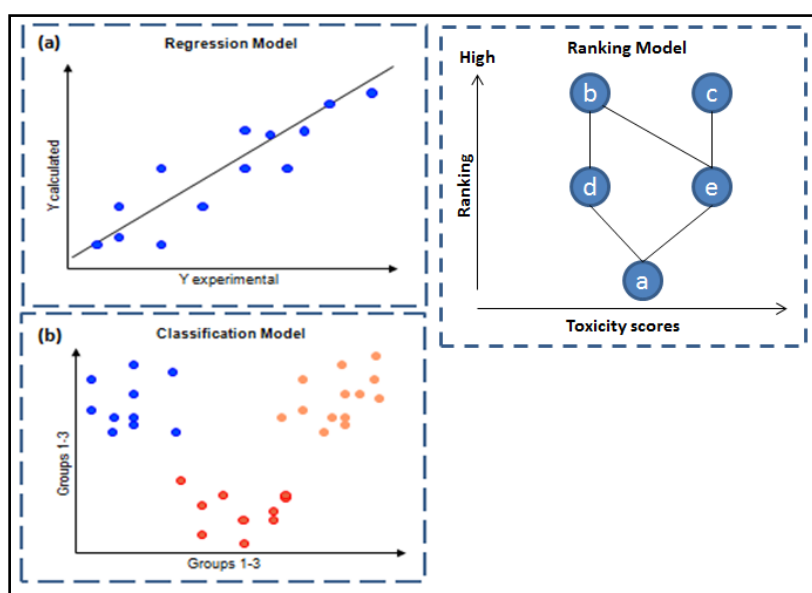


Figure 2.8: Different types of QSAR strategies (after Todeschini & Consonni, 2008): (a) Regression model focusing on the best fit classifications; (b) Classification model to characterise the similarity between certain properties, and (c) Partial order ranking models, based on Hasse diagram technique. The figure shows the ranks of chemicals according to their toxicity levels, which relies on statistical significance

With the availability of different types of properties and descriptors, QSAR is able to deploy different strategies, such as regression models, classification models and ranking models, to enhance relationship analysis, as illustrated in Figure 2.8. One of the earliest techniques to address the relationship between compound's physicochemical properties or structure and biological activity, was described by Hansch in 1969. The Hansch Analysis is primarily based on the regression concept, which aims to obtain the best-fit for the observed values. The Hansch function as shown in Equation 2.1 in its simplest form is given below:

$$\log \left(\frac{1}{C} \right) = k_1 \log P - k_2 (\log P)^2 + k_3 E_s + k_4(\partial) + k_5 \quad (\text{Equation 2.1})$$

Where $\log 1/C$ is the logarithm inverse of the molar concentration C ;

C denotes the minimum effective dose of concentration;

$\log P$ constitutes the logarithm of the water coefficient P ;

P denotes the coefficient of water partition;

E_s and ∂ are descriptors derived from experimental observation;

∂ denotes Hammett substituent constant;

and, K_x is constant values to influence the fitting of the relationship.

This equation derives a relationship between the relative biological activity ($\log 1/C$) and the drug activity hydrophobicity defined by $\log P$ (Hansch, 1969). Another well-known technique is the Free-Wilson analysis, which is essentially a mathematical model examining structural features and their presence or absence. Free-Wilson analysis is expressed by the Equation 2.2 below:

$$\log \left(\frac{1}{C} \right) = \sum a_i x_i + \mu \quad (\text{Equation 2.2})$$

Where x_i denotes the presence or absence of a particular substituent at the relevant position in the compound;

a_i denotes the contribution of the corresponding substituent/position i combination to the activity;

and μ is the activity value of un-substituted compound.

The two methods cited above, however, have several limitations. These include being dependent on only similarly structured compounds in their analyses. This makes the process more difficult as a chemical database would normally contain a structurally diverse set of compounds with different types of biological activities (Cramer, Redl, & Berkoff, 1974). It is worth noting that Hansch Analysis and the Free-Wilson method are closely related, even

though they exhibit marked differences in regard to their approaches to analysis. In addition, the two techniques are aimed more towards lead optimisation rather than lead discovery. This eventually led to the concept of Substructural Analysis as posited by Cramer et al. (1974). They attempted to systematically correlate sets of biological and substructural data for lead discovery, and consequently developed Virtual Screening.

2.8 Machine learning

In 1959, computer scientist Arthur Lee Samuel popularised the term machine learning as the *“Field of study that gives computers the ability to learn without being explicitly programmed.”* Mitchell (1997) later provided a broader definition of machine learning where *“A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E ”*. A recent definition states that machine learning is a field in computer science concerned with the computational properties of algorithms. This allows a machine, under human supervision, to learn meaningful and complex patterns and predict new outcomes of results or conclusions (Wale, 2011). Generally, machine learning explores the study and construction of algorithms that are able to learn from, and make predictions about, data. Such algorithms operate by building a model from examples of inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions. Machine learning positions itself to address the limitations of human cognition and information processing, especially on the handling of enormous data, their relationships and the analysis that follows. To employ such a powerful tool, however, the rationale, objectives and expectations have to be clearly defined and understood. It is necessary to identify the tasks humans can and cannot do, and what tasks the machine can do better than humans. In terms of cooperation, it is important to establish what tasks the two can achieve together. From this, machine learning techniques can essentially be grouped into supervised and unsupervised learning, depending on human-level instructions and machine-level freedom (Wang and Summers, 2012).

Supervised learning is a type of learning that attempts to gauge data and models based on user-defined labels and objectives. In LBVS, supervised learning has been used to solve classification problems, such as the popular classification between active or inactive compounds (Duch, Swaminathan & Meller, 2007). The learning process is applied on a training set, while a corresponding test set validates the accuracy of the prediction algorithm. In the real world, the concept of supervised learning is likened to a child learning about fruit and vegetables based on labelled objects from the two groups, such as fruit like apples and

oranges, and vegetables like broccoli and carrots. The children are exposed to (learn about) both fruit and vegetables for several days. After a week, they are tested by the teacher to validate how well they have learned to differentiate the two, given that this time there is a wider selection of fruit and vegetables to identify.

On the other hand, unsupervised learning is the automated learning process of understanding a data set, given little or no knowledge of its specific property, label or objective. Such data are also known as unlabelled data. The method is used to look for structures or unusual patterns but without having a clear goal of the learning purpose, thus having no clear reward system that might indicate a solution. An example of unsupervised learning is common to the function of the human brain. For instance, almost all activities of a human involve sensory and visionary elements (touch, smell and sight), which provide information that gets passed to the human brain. Having no clear objective of what this information constitutes, somehow the brain is able to analyse the information from the elements, form a suitable response, and subsequently react to the activities in its own unique way.

It is interesting to note that almost every technique in Virtual Screening involves the use of machine learning methods. The list of VS methods in Table 2.2 includes similarity searching (SS), itself primarily a data mining and machine learning tool. The similarity searching method uses a single reference compound to predict new active compounds, based on the similarity value calculated using a similarity coefficient. This approach has led to advanced applications of machine learning methods which require a subset of compound data to form a training set. This is used to extract knowledge or a predictive model which is then applied to a test set for identification of new active compounds. Although various machine learning techniques have been available for some time, this study focuses on techniques that have been used in the ligand based virtual screening context (LBVS). In addition to SS, other popular machine learning techniques in LBVS include Substructural Analysis (SSA), Binary Kernel Discrimination (BKD), Support Vector Machines (SVMs), Decision Trees (DT), Artificial Neural Networks (ANN) and K-Nearest Neighbours (KNN). All of the techniques mentioned above make use of both active and inactive compound information while progressing through rigorous stages of learning. The goal of learning in this case is to be able to achieve the accurate prediction potency of a new chemical compound, given its chemical structure, against the target receptor (Wale, 2011). Table 2.3 compares and summarises the machine learning techniques described above, related to LBVS.

Table 2.3: Comparison of machine learning classification techniques in LBVS (after Lavecchia, 2015)

Machine learning techniques	Researchers	Definition	Applications	Advantages	Disadvantages	Type of learning
SVM	Vapnik and Chervonenkis (1964) were the first scholars to introduce the SVMs concept.	<p>Support vector machines (SVMs) are a set of kernel-based supervised learning methods used for classification and regressions.</p> <p>SVMs try to minimise the empirical classification error and maximise the geometric margin simultaneously on the training set, which leads to a high generalisation ability for the new samples.</p> <p>Some features being extracted may not be relevant and become noisy features which can result in poor performance. The best features can be identified by minimising bounds on the leave-one-out error (Wang, 2005).</p>	<p>In LBVS, this technique is applied to predict compounds' activity, based on the classification and ranking of their probability, in decreasing order.</p> <p>Classification is performed for the purpose of separating drugs and non-drugs and also to differentiate between active and inactive compounds (Byvatov, Fechner, Sadowski and Schneider, 2003; Zernov, Balakin, Ivaschenko, Savchuk and Pletnev, 2003; Warmuth, Liao, Rättsch, Mathieson, Putta and Lemmen, 2003 and Jorissen and Gilson, 2005).</p> <p>The active compounds are separated from the inactive compounds by the maximum margin hyperline. The margin is the distance from the decision boundary to the nearest training point of both classified sets.</p> <p>Hyperline is defined by identifying the support vector (i.e. the point on the margin).</p>	<p>No presumption of association between target property and descriptors used to represent compounds.</p> <p>Overfitting rarely occurs.</p> <p>Accurate classification method.</p>	SVM technique suitable for binary classification tasks.	Supervised learning.

DT	Breiman, Friedman, Olshen and Stone (1984) were apparently the first to develop simultaneous Classification And Regression Trees (CART) in statistic.	<p>The decision tree is usually represented as a treelike structure. The root is placed at the top of the tree and the leaves at the base.</p> <p>The tree is divided into two or more branches from a single trunk. Each branch denotes the outcome of the tests, and the leaves denote the label of a class: i.e. decisions taken after a series of queries, (questions are asked, starting from the root node). The answers are used to choose branches until a leaf is reached, based on classification rules.</p> <p>A path from root to leaf denotes the classification rules (Kohavi and Quinlan, 2002)</p>	<p>Commonly, this technique is used in the prediction of drug-likeness and the identification of biological activity, which identifies substructures of separate active compounds from inactive ones in a dataset.</p> <p>DT is also used to classify chemical compounds as drug or nondrug.</p> <p>A number of studies have applied DTs in predicting ADME properties (Lamanna, Bellini, Padova, Westerberg and Maccari, 2008; Sakiyama, Yuki, Moriya, Hattori, Suzuki, Shimada and Honma, 2008; Hou, Wang and Li, 2007; Deconinck, Zhang, Coomans, and Vander Heyden, 2006; Gleeson, Waters, Paine and Davis, 2006, Mente and Lombardo, 2005).</p>	<p>No presumption of association between target property and forms used to represent compounds.</p> <p>A fast classification method.</p> <p>Able to perform multiclass classification.</p>	Affected by overfitting possibility when the training set is small and filled with a huge variety of descriptors.	Supervised learning.
NBC	NBC is based on the work of Thomas Bayes between 1702 to 1761 (Panda and Patra, 2007)	NBC is based on Bayes' theorem with independence assumptions between predictors. The equation of Bayes' theorem is as follows:	<p>In chemoinformatics Bayes' theorem helps chemists to predict new active compounds by using known actives.</p> <p>Specifically, the NBC is used to identify the probability that a</p>	<p>A very fast classification algorithm to use.</p> <p>Can be applied to solve real data.</p>	Assumes features are not related.	Supervised learning.

		$P(A/B) = \frac{P(A/B) \cdot P(A)}{P(B)}$ <p>The equation describes the probability P of the condition A that might occur in relation to another condition (Kotsiantis, Zaharakis and Pintelas, 2007).</p>	<p>compound is active, based on descriptor representation (Angelopoulos, Hadjiprocopis, Walkinshaw, 2009).</p> <p>NBC is also applied to rank compounds of the database structure based on the activity probability (Lavecchia, 2015).</p>	Not affected by insignificant features.		
KNN	KNN was originally introduced by Fix and Hodges in 1951.	<p>KNN is a simple algorithm, which is used to classify a new data point, x, by identifying the most similar x-based training points.</p> <p>In other words, the KNN has been employed to identify k nearest neighbours of x, regardless of labels in a given training set (Cunningham and Delany, 2007).</p>	<p>In LBVS, KNN is used to classify a molecule based on its properties in relation to its neighbouring molecules.</p> <p>Its neighbouring molecules are relatively defined by the variable k, where a larger k value constitutes a larger group of neighbouring molecules considered when classifying a specific molecule.</p> <p>Usually Euclidean and Manhattan distance is applied to calculate the distance between objects, which is represented by a position vector in the multidimensional feature space (Lavecchia, 2015).</p>	<p>No association between target property and descriptors used to represent compounds.</p> <p>Able to perform multiclass classification.</p> <p>Fast application on training sets.</p>	Classification performance largely depends on the distance measures used.	Unsupervised learning.

A major concern in machine learning applies to the extent to which the machine can actually learn, and the extent to which learning to accomplish a task successfully is considered proof of intelligence. Such limitations may be an apparent barrier to the possibility of highly efficient and inexpensive in-silico screening methods. These constraints may be significant factors when critically evaluating the various machine-learning based systems used in Virtual Screening. Specifically in LBVS, the first application of machine learning in computer-aided molecular design (CAMD) was substructural analysis. It was introduced by Cramer et al. (1974) in the early Seventies as a tool for the automated analysis of biological screening data, and which continues to be of much current interest.

2.9 Substructural analysis

2.9.1 History and definition

Substructural analysis (SSA) is a method under ligand-based virtual screening and was pioneered by Cramer et al. (1974). The technique is one of the earliest forms of machine learning method used in chemoinformatics. In substructural analysis, it is assumed that each molecule in a dataset is characterised by a series of binary descriptors, most commonly in the form of a 2D fingerprint in which each bit denotes the presence or absence of a substructural feature (often referred to as a fragment). Associated with each such bit is a weight that is a function of the number of active and inactive molecules that have that bit switched on, i.e., that contains the corresponding fragment. This weight reflects the probability that a molecule containing that substructural feature will be active (or inactive); for example, the weight might be the fraction of the active molecules containing that particular fragment. A molecule is then scored by summing (or otherwise combining) the weights of those bits that are set in its fingerprint, the resulting score representing the overall probability that the molecule will be active. A major assumption of SSA is that a given substructure can influence the determination of the activity level of a molecule, regardless of the compound in which it occurs (Cramer, Redl, & Berkoff, 1974).

2.9.2 Fundamental components of the SSA

SSA stresses the relationship between a substructure and its activity state, either active or inactive, for each given compound. The analysis involves a determination of a series of weights for every fragment in a compound. These individual compounds are then ranked in a particular order of scores in order to highlight activity frequency. This, in essence, acts as the basis for a large machine-learning-based screening program which enhances the efficiency of

lead identification. This is because it is able to predict the differential probabilities of activity in a set of untested, structurally diverse compounds. SSA mainly uses fragment representations, such as 2D fingerprints in its analyses, in order to characterise fragment occurrences more successfully throughout each compound. SSA is considered to be most useful in the context of large screening programs as it is able to focus on the importance of molecular features independent of the diversity of the compounds' structural representation in nature (Cramer, Redl, & Berkoff, 1974). Substructural Analysis does not discard the contributions of information from low-probability structures, based on their activity state. It uses such information to quantify the degree of probability of each compound being a potential hit. Specifically, fragment occurrence levels in active and inactive compounds are gauged from a known set of compounds. At all times, however, it is assumed that most known active compounds are similar to other active compounds.

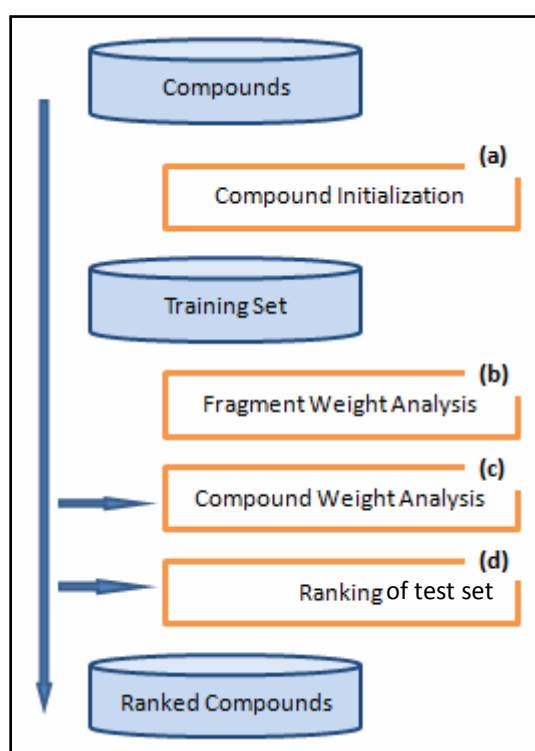


Figure 2.9: Simple schematics of Substructural Analysis

Figure 2.9 illustrates the steps involved in Substructural Analysis. Selected compounds are initialised at first, otherwise known as a training set (Fig. 2.9a). All compounds in the training set are represented by substructural fragments. Each of the fragments is assigned a weight value based on the weighting function chosen and on the occurrences of the fragment in the active and inactive molecules in the training set (Fig. 2.9b). This provides a measure of the

likelihood that a compound containing that fragment will be active. The overall score for a compound in the test set is computed by summing (or otherwise combining) the weights for the fragments present in that compound (Fig. 2.9c). All the compounds in the test set are then ranked (Fig. 2.9d) in the order of highest to lowest compound weights, reflecting the most likely active compounds at the top. Fundamentally, SSA assumes that all fragments in a given structure have some degree of influence on activity levels in screened compounds.

Computation of the fragment weights is usually in the form of a statistical evaluation of relevancy. It uses a form of equation that manipulates the active and inactive state of the compounds. Ranking of these compounds theoretically highlights the best set of compounds in a dataset. Compounds at the top of the ranking, i.e., compounds with high scores, will contain fragments with high weights and thus these compounds will have high probabilities of being active (Cramer et al., 1974). The predictive study method was actually introduced in the pioneering work on SSA. The basic step involves choosing at random a certain percentage of active and inactive molecules called a training set, then deriving the weights based on that set; and finally calculating the scores across all data. This model of study has been implemented by various authors, especially in their comparative performance review of Substructural Analysis (Ormerod, Willett & Bawden, 1989; Cosgrove & Willett, 1998; Wilton, Willett, Lawson & Muller, 2003).

2.9.3 The SSA weighting schemes

Weights are calculated for each fragment in the molecule database to mark their likelihood of usefulness and relevancy as a lead compound. Various weighting schemes have been introduced by researchers with the aim of improving the prediction model since the original weighting scheme was proposed by Cramer et al. (1974). The weighting calculation ranges from simple to complex. While the use of active molecule information is compulsory in any given scheme, inactive molecules are included in some of the more advanced and recent schemes.

Cramer's weighting schemes

Cramer et al. (1974) introduced the original weighting scheme in the pioneer study on Substructural Analysis. The method involves the following: (1) Calculation of the Structure-Activity Frequency (SAF) for each fragment *I* in every compound, governed by the Equation 2.3:

$$SAF(I) = \frac{ACT(I)}{TOT(I)} \quad (\text{Equation 2.3})$$

Where $ACT(I)$ is the number of active compounds containing the fragment I ;
and $TOT(I)$ is the total number of compounds containing the fragment I .

Then (2) computation of the mean SAF value for every compound, also known as the Mean Structure-Activity Frequency (MSAF). The MSAF values for each compound are then (3) used to rank the whole compound collection in descending order. This is based on the theory that the molecules with greatest probability of activity are located at the top. Both the summation mean of each compound's weighting values and the subsequent ranking methods have remained unchanged in recent alternative weighting schemes. SAF is perhaps the most traditional scheme for substructural analysis, relying on a rather simple, yet effective statistical measure. It is also considered to be the most basic of all weighting schemes (Cosgrove & Willett, 1998). While it is appealing in terms of its simplicity, Cramer's SAF does not directly adopt the inactive molecule information in his equation. In essence, this causes a certain sense of bias to the active compound group in the calculation of weights. The clear limitation of this weighting scheme has prompted many subsequent studies to achieve a more accurate analysis.

Cramer (ibid) identified the possibility of bias in cases where there is only a unique presence of individual fragments. The latter presented a 'workaround' by introducing a factor of randomness in SAF averaging. A set number of groups were formed randomly and the computation of the MSAF was calculated or recalculated for each group. This decreases the probability of the compound's MSAF values being affected by certain members of its group. Such a workaround was, however, abandoned or not carried through in subsequent studies, as the manipulation of both the active and inactive compound information was later considered more insightful. An alternative scheme was suggested by Redl, Cramer, and Berkoff (1974) which is closely related to the above, and is known as the Substructure Activity Score (SAS), is given by the Equation 2.4:

$$SAS(I) = ACT(I) - (TOT(I) \times NACT/N) \quad (\text{Equation 2.4})$$

Where $NACT$ is the total number of active compounds in the database;
and N denotes the total number of compounds in the database.

The SAS scheme above takes into account the direct contribution of inactive compounds. It achieves this by providing a difference between the active numbers of molecules containing the fragment in question versus the number to be expected if the fragment had no influence on the activity of the molecule (Cosgrove & Willett, 1998)

Carhart's method

Carhart et al. (1985) published work on a method called '*trend vector*'. This performs correlations between atom-pairs and structure-activity properties to estimate the biological activity of new compounds. Trend Vector was proposed using the atom-pair concept, but Ormerod et al. (1989) and Cosgrove et al. (1998) both reported that it can be extended with the use of other 2D fingerprints. It is worth noting that fragment weights generated by the trend vector are a somewhat scaled version of Cramer's SAS. Each unique atom pair is assigned a score and this score for the atom pair *I* as shown in Equation 2.5:

$$\frac{ACT(I) - \left(TOT(I) \times \frac{NACT}{N} \right)}{N} \quad \text{(Equation 2.5)}$$

Hodes' measures of statistical-heuristic methods

Hodes, Hazard, Geran, and Richman (1977) introduced a different weighting scheme to address the limitations of Cramer's weighting functions. It has been claimed that their methods are more statistically derived (Ormerod et al., 1989), as they seek to quantify the probability of a compound being active based on estimation by statistical averaging (the statistical mean in particular); and then subsequently correlated with the actual observation of the activity. The standard deviation principle was mainly used to derive a formulated weighting scheme. In this sense, weights are assigned according to the statistical significance between the active features against the expected number. Hodes' method of weighting scheme is, relatively speaking, not a simple equation, compared to other existing weighting schemes. The number of Standard Deviation (*NSD*) works by calculating the Standard Deviation *I* value - termed *SD(I)* - for each fragment in the compound, and summing it to get a total *SD* - termed *TOTNSD* - which would act as the fragment weight.

$$P = \frac{NACT}{N} \quad (\text{Equation 2.6})$$

$$MAVG(I) = TOT(I) \times P \quad (\text{Equation 2.7})$$

$$DVFACT(I) = \sqrt{TOT(I) \times P(1 - P)} \quad (\text{Equation 2.8})$$

$$SD(I) = \frac{(ACT(I) - MAVG(I))}{DVFACT(I)} \quad (\text{Equation 2.9})$$

Where *MAVG* is the expected mean average of compounds to contain fragment *i*;
P is the probability that a compound will contain fragment *i*;
and *DVFACT(I)* is the Deviation factor as constant for Standard Deviation calculation.

As reported by several authors (Ormerod, 1989; Cosgrove 1996), Hodes method is by no means a simple calculation process and shown in Equations 2.6 to 2.9. Consider this example, which shows the workings of the Hodes *SD* value. Given a set of 100 total compounds (represented as $N=100$) in which 20 are active ($NACT=20$), 10 compounds are determined to contain the fragment of interest *I* ($TOT(I)=10$). The probability *P* is then calculated as 20/100, or 0.20. The mean average of fragment *I*, *MAVG(I)*, is calculated as $(10 * (20/100))$, which equals 2. Calculation of *DVFACT(I)* yields a constant factor of $(10 * 0.20 (1-0.20))^{1/2}$, equalling 1.265. Assuming that there are 5 active compounds which actually have the fragment *I*, *SD(I)* can be computed as $(5 - 2) / 1.265$, which is 2.371 SD away from the mean value of 2. Hodes (1976) also mentioned the probability of getting 2.371 SDs away by chance, hence further approximation by normal distribution for the two-tailed value for *P* is used. This is done by referring to the statistical table for the area under the curve of the SD which yields a *P* value of 0.0178. The weight needs to be the inverse of *P* (i.e. 1/*P*) and also logged: this gives 1.749. This value indicates the measure of probability of predicting that the compound is inactive.

The above equation involves computing a measure of the probability of activity of a compound based on the statistical evidence of its features. The use of summation and multiplication for index *i* are used to combine probabilities under the assumption of the

independence of features. The use of $\log 1/P$ gives a good spread and avoids lower values of weights, since the active and inactive compounds have been specifically identified earlier (Hodes et al., 1977). The conversion from a number of standard deviations to $\log 1/P$ is shown in Figure 2.10. Hodes (1976) and Ormerod (1989) each reported that either the number of standard deviations (NSD) or the probability (P) could be used as a fragment-weighting scheme for SSA. In this way, Hodes presented a wide choice of weighting schemes, which can be derived from this particular workflow: (i) weighting based on the standard deviation values above, termed SD; (ii) weighting based on the probability of being SD away by chance, termed PR; and (ii) the inverse of SD and PR, each termed SDI and PRI respectively. These four weighting schemes were implemented in Ormerod's review.

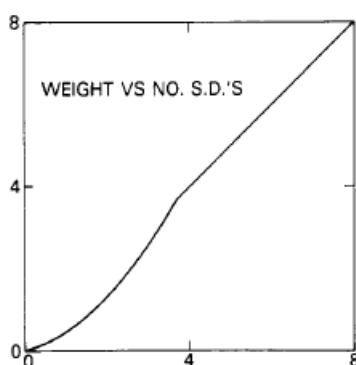


Figure 2.10: Plot of weight versus number of standard deviation with conversion to $\log 1/P$
(Hodes et al., 1977)

Robertson-Sparck Jones statistical measures

Ormerod et al. (1989) investigated the use of weighting schemes meant for general information retrieval, based on the works of Robertson and Sparck Jones. They are derived from a formal probabilistic theory of relevance weighting, and were intended to be used in bibliographical searches rather than as a chemistry-based screening method. Unlike Cramer's scheme, which generally disregards the inactive molecular contribution or the rather indirect use of such information in Hodes' methods, the Robertson-Sparck Jones scheme utilises inactive molecule information by incorporating them in their equations as shown in Equations 2.10 to 2.13.

The Robertson-Sparck Jones weighting is originally a set of techniques used to measure the relevancy of documents by use of statistical weighting. It is meant for general document retrieval systems, but makes heavy use of the binary format of descriptors (in this case binary

index descriptions of documents). It is thus possible to relate the collection of documents represented by index terms in binary format to that of a database of chemical compounds containing binary indices of 2D fragments. This assumption is similar to that described by Willett (2009). He suggests that there is a general similarity between textual and chemical databases, not just by virtue of their impending characteristics but by how they are accessed, analysed, manipulated and put into context in terms of the relevance and relativity between the items in such databases. In the case of a chemoinformatics' perspective, four different relevance weighting schemes, termed R1 to R4, as originally derived by Robertson and Sparck-Jones (1976), were modified by Ormerod et al. (1989) in a detailed study to accommodate their use for chemical substructures:

$$R1 = \log \left(\frac{ACT(I)/NACT}{TOT(I)/N} \right) \quad (\text{Equation 2.10})$$

$$R2 = \log \left(\frac{ACT(I)/NACT}{INACT(I)/NINACT} \right) \quad (\text{Equation 2.11})$$

$$R3 = \log \left(\frac{ACT(I)/(NACT - ACT(I))}{TOT(I)/(N - TOT(I))} \right) \quad (\text{Equation 2.12})$$

$$R4 = \log \left(\frac{ACT(I)/(NACT - ACT(I))}{INACT(I)/(NINACT - INACT(I))} \right) \quad (\text{Equation 2.13})$$

Where $INACT(I)$ is the total number of inactive compounds containing fragment I ;
and $NINACT$ is the total number of inactive compounds in the dataset.

Weighting schemes R1 and R3 use the active molecule information, while R2 and R4 in particular use the active and inactive molecule information, here termed $INACT(I)$ and $NINACT$, to reflect the probabilistic derivation of such relevance. As stated by Robertson and Sparck-Jones (1976), these schemes are driven by 2 assumptions: (1) the assumption of independence; and (2) the principles of ordering. Assumption 1 states that the distribution of terms in the relevant documents is independent, and the distribution of terms in all documents is independent. Assumption 2 states that the distribution of terms in the relevant documents is independent and the distribution of non-relevant documents are also independent. The two

ordering principles are as follows: Ordering 1 is based only on the presence of search terms; Ordering 2 is based on both the presence and absence of search terms in the documents.

Several principles can be drawn when the weighting scheme is applied in molecular information contexts; in the case of the assumption of independence, two assumptions exist. Assumption 1 is that the distributions of fragments in the active compounds and in all compounds is independent; Assumption 2 is that the distribution of fragments in active compounds is independent and the distribution of fragments in in-active compounds is independent. R1 is based on Assumption 1, Ordering 1; R2 is based on Assumption 2, Ordering 1; R3 based on Assumption 1, Ordering 2; and R4 is based on Assumption 2, Ordering 2. TOTR1, TOTR2, TOTR3 and TOTR4 values are then calculated for each compound by summing the individual fragment values for R1, R2, R3 and R4 respectively. The compounds in the datasets are then ranked in descending order based on TOTR1, TOTR2, TOTR3 and TOTR4 values.

In regard to the relevance weighting schemes above, if any of the elements are zero, then it would ultimately cause the weighting to be zero as well, which will definitely be problematic. Robertson and Sparck-Jones (1976) reported in the document retrieval context on how to overcome such a problem; if there is a zero value for any element in the weighting scheme, the zero value will have to be replaced with 0.0000001.

Mayer and Sens weighting schemes

Mayer and Sens (1988) have reported substructural analysis studies using a fragment weight, given by Equation 2.14:

$$\frac{ACT(I)/NACT}{INACT(I)/NINACT} \quad (\text{Equation 2.14})$$

The equation above is identical to the R2 weighting scheme, except there is no logarithms function included, as used by the R2 scheme. The rationale for using logarithms is explained in detail by Robertson and Sparck-Jones (1976).

AVID weighting schemes

Several other variations of weighting schemes, closely related to Cramer's original scheme, were investigated by Ormerod et al. (1989). One is based on the work of Avidon et al. (1982) known as the Avidon scheme (AVID) is defined by the Equation 2.15:

$$AVID(I) = \frac{ACT(I) + 1}{(TOT(I) \times NACT / N) + 1} \quad (\text{Equation 2.15})$$

Similar to Cramer's scheme, a mean AVID (MAVID) is also calculated for each compound by summing the AVID values for all fragments in that compound. It then divides this total by the number of fragments in the compound. The compounds in a dataset are then ranked based on the relevance value of each compound. (Avidon, Pomerantsev, Golender, & Rozenblit, 1982) also reported the use of two further fragment weights:

$$WT1(I) = \frac{ACT(I)}{INACT(I)} \quad (\text{Equation 2.16})$$

$$WT2(I) = \frac{ACT(I) - INACT(I)}{TOT(I)} \quad (\text{Equation 2.17})$$

Referring to the fragment weights above (Equation 2.16 and 2.17), there is an issue in the calculation of WTI whenever $INACT(I)$ is equal to 0. For such cases, a prior value of 0.0000001 is assigned for $INACT$ of instance I . Mean values of MWT1 and MWT2 are calculated for each compound of the dataset by summing the WTI or WT2 values for all fragments in the compound and the sum is divided by the total number of fragments in the compound. The compounds in the datasets are then ranked in descending order, based on MWT1 and MWT2 values.

2.10 Application of SSA for drug discovery

In previous studies a number of SSA implementations have been commercialised for industry, or developed for academic pharmaceutical studies worldwide. The earliest applications known were carried out by the Drug Research and Development Programme in The United States, which was for the selection of compounds for antitumor screening (Hodes et al., 1977). The SSA method was chosen as this method has the ability to handle a large number of structurally

diverse compounds in an automated manner, and for the analysis of their biological activity relationships using virtual screening methods. The National Cancer Institute in the 1970s was heavily involved in the development of new drugs to treat human cancer. Anti-tumour screening programmes involved the testing of compounds in a variety of animal tumour models.

Certain limitations were prevalent at the time: there was a limited capacity for screening (15000 synthetics per year) and close to an infinite possibility of new compounds. The automated selection of drugs differs from standard approaches, such as the quantitative structure-activity approach, whereby the former is applied across a broad range of compounds rather than a single class. There may be unusual and unnoticed combinations of certain chemical structure features that impart a specific biological activity (Hodes et al., 1977).

2.10.1 Hodes study in National Cancer Institute (NCI) for tumour screening program

Hodes' detailed study on cancer disease led to the introduction of a statistical-relevance weighting scheme. The first application of SSA was initiated in 1976 on a specific Mouse Ependymoblastoma study, where a training set comprising 170 compounds was selected, covering a broad range of structure classes. These had been subject to previous structure-activity studies (Hodes, 1976). Putting aside issues of structure redundancies, through elimination and further multiplication of actives based on occurrence probability, the result of the first study concluded that the use of SSA through Hode's method was comparable to more complex pattern recognition methods. It required, however, further testing on larger scale experimental sets to be able to deem them usable in a commercial sense (Hodes, 1976). Hodes continued his work on SSA in 1976 with a larger scale study, specifically on a novel anti-tumour drug design. The results showed a limited justification in the selection of appropriate compounds through variations of methods imposed in the experiments. The research, however, was undermined by issues of the biasing of compounds selected for the training sets. These factors resulted in the performance being unrealistically positive (Hodes, 1976).

Hodes (1981) published a more detailed, larger scale study on the anti-tumour screening programme, particularly on the search for a new drug responding to mouse lymphocytic leukaemia or P388. The study involved some 120 highly active compounds and 2000 moderately active compounds, while 33,000 more compounds were deemed inactive. Further selections for a training set were made, comprising 80% of the compounds above, in order to

eliminate any doubts about bias. Screening performance was validated by both the computer and the chemist selection on the agreement of actives, which was largely similar. Hodes concluded that the results of the computed selection of compounds were favourable in this sense. However, he did not discount several limitations due to problems of biasing, chemical structures, feature choice on fragments, redundancies and the difficulty of application (Hodes, 1981).

2.10.2 CASE and MULTICASE

Klopman (1984) introduced a program to study the relationship between the structures and the biological activity of organic molecules called Computer Automated Structure Evaluation (CASE). The program was based on the artificial intelligence concept with an algorithm that automatically identifies molecular fragments with a high probability of relevancy in terms of the biological activity in question. CASE was considered as a knowledge-based system. One of the original objectives of CASE was to intrinsically maximise the effort to focus on active molecules from the mass of inactive ones. The basic workflow of CASE is: (1) the generation of all possible connected fragments for every molecule in the dataset; (2) the determination of active and inactive molecule distribution; (3) the evaluation of each active or inactive molecule distribution concerning whether it is specifically in the active molecules, in the inactive molecules or if the distribution has occurred simply by chance.

In subsequent studies, it was discovered that CASE had several limitations apart from its success in the automated prediction of relevant molecules, based on their inherent biological activity response. Klopman (1992) noted the lack of sensitivity to geometrical differences and their inability to be handled in a hierarchical manner. Klopman later presented a new program called MULTICASE. It involved the use of logical and hierarchical features to select descriptors for the improved, automated prediction of relevant molecular substructures. CASE and MULTICASE have both been applied to a wide variety of problems including the prediction of physiochemical properties, the generation of quantitative structure-activity models (in particular toxicology, mutagenicity and biodegradation) and the analysis of large data sets (Leach & Gillet, 2007).

Recently, the CASE Ultra algorithm was introduced by Chakravarti, Saiakhov and Klopman (2012). It is the latest methodology, which is mainly based on the MULTICASE methodology. The CASE Ultra algorithm was implemented in the CASE Ultra Expert System. It is a

fragment based QSAR machine learning method and has recently been applied in the field of chemical toxicity (Chakravarti, Saiakhov & Klopman, 2012). CASE Ultra generates predictive models automatically by learning from training sets which consist of both active and inactive compounds. It does this by scanning each individual compound to find structural alerts, also called positive alerts related to activity. The final set of positive alerts is further used to build a local QSAR model. Several advantages of this new algorithm are its ability to utilise the capacity of modern hardware and software. It also allows the handling of large training sets and is known to be unlimited with regards to its learning-path determinations (Saiakhov, Chakravarti, & Klopman, 2013).

2.10.3 SLASH

The SLASH program is an implementation of SSA, developed for medicinal chemists in Zeneca Pharmaceuticals in 1998 as an acceptable molecular descriptor in identifying a lead compound. The objective of the programme was to analyse a large number of compounds, identify complete functional groups and determine their significance in relation to activity (Cosgrove & Willett, 1998). SLASH examined the performance of scoring functions by using larger and more useful functional group fragments, incorporating four weighting schemes: (1) the SAF scheme by Cramer et al. (1974); (2) the SAS scheme by Redl et al. (1974); (3) the R2 schemes by Robertson and Sparck-Jones (1976); and (4) the Number of Standard Deviations (NSD) by Hodes et al. (1977). One of the objectives of SLASH is to determine if the use of more sensible functional group fragments in a substructural analysis can improve the performance of any of the fragment scoring functions (Cosgrove & Willett, 1998). Similar to Ormerod (1988), Cosgrove (1998) tested three statistical measures within SLASH to assess the effectiveness of each scoring function in SSA: Percent misplaced (PM), Error Score (ES) and comparison of rankings (Cosgrove & Willett, 1998) as discussed in Section 3.5.3.

2.10.4 Other applications of SSA

A further example of the application of SSA was carried out by Capelli et al., (2006). They employed SSA and 1024 Daylight fingerprints to create a set of GlaxoSmithKline compounds biased towards Ligand-Gated Ion-Channel Ligands (LGIC). In the study, they used the R2 weighting scheme to calculate the score of each compound by utilising the information on the activity state which are LGIC (active) and non-LGIC (inactive). As a consequence, they found that the scoring function effectively discriminated the known LGIC from the non-LGIC training set, and efficiently ranked the 550k in a house test set.

Another application is described by Anzali et al. (2001), who employed SSA as an algorithm for use in computer systems, known as the Prediction of Activity Spectra for Substances (PASS). In addition to the use of the SSA algorithm, PASS also provides chemical structure information, guided to improve accuracy in the differentiation between “drug” and “non-drug” compounds. Lederle also conducted an experiment on the use of substructural analysis to calculate the independence of atom-pair and structure-activity properties as either active or inactive, for each given compound. In the study, Carhart et al. (1985) employed SSA to calculate the score for each compound by using the information of the presence of a particular substructure in a compound. It uses such information to estimate the probability of the biological activity of new compounds being potential hits. The study applied the same principle that was introduced by Cramer, but with a different approach. Carhart et al. formed the weighting based on a heuristic and used the appearance of an atom-pair in the structures as a molecular descriptor to be analysed. The heuristic technique is also known as the trend vector analysis, whereby the occurrence of atom pairs of different types is correlated with biological activity. The atom-pair is a substructure containing two non-hydrogen atoms with the interatomic bond separation. Each unique atom pair is assigned a score, as shown by Equation 2.18.

$$\frac{ACT(I) - \left(TOT(I) * \frac{NACT}{N} \right)}{N} \quad \text{(Equation 2.18)}$$

Permeability of the blood–brain barrier (BBB) is a system that guards the brain from harmful substances, which circulate in the blood of the central nervous system via capillary cells. Mensch, Oyarzabal, Mackie and Augustin (2009) reported on the use of the SSA method in the analysis of BBB, where, SSA was applied to categorise the unknown BBB permeability of the molecules over a pool of known BBB+ and BBB- molecules. First, the molecules are fragmented into a possible fragment size that is defined by the user, and then the summation and mean value of the fragment score of each compound is determined. Finally, the compounds are ranked in descending order, where the highest numbers of compounds, which have the highest potential to permeate the BBB, were placed at the top.

2.11 SSA and Naive Bayesian Classifier (NBC)

SSA has been studied in considerable detail by researchers at the National Institutes of Health in an extended programme to develop novel anti-cancer agents, and also by workers at Lederle (Carhat et al., 1985; Hodes et al., 1976; Hodes et al., 1977; Hodes et al., 1981).

However, it is only in the last few years that the approach has become widely used. Although this was perhaps not recognised when the approach was first introduced, SSA is an example of a Naive Bayesian Classifier (NBC). NBC by definition is a type of probabilistic classification algorithm, based on Bayes theorem, which carries a strong assumption of independence when characterising structurally the object(s) to be classified. The NBCs approach can be used to classify a trained data set into two different classes of active and inactive, based on their probability value of in(activity) calculated using Bayes` s theorem. For example, Xia, Maliski, Gallant and Rogers (2004) employed NBCs to categorise kinase inhibitors activity, in which they claimed the method was able to find a compound that was structurally diverse to known actives.

In principle, SSA and Naive Bayesian Classifier (NBC) are both machine learning methods which can be used for ligand-based virtual screening in drug discovery, and which can be shown to be mathematically related. The notion of similarity between SSA and NBC was noted by Hert et al. (2006). Taking note of Xia` s et al. (2004) NBC based equation is used in the Pipeline Pilot software; Hert et al. proved that it can reach a form of equation similar to Robertson-Sparck Jones' R1 method, as shown below:

$$P(A) = \frac{NACT}{N} \quad \text{(Equation 2.19)}$$

$$P(A|I) = \frac{ACT(I)}{TOT(I)} \quad \text{(Equation 2.20)}$$

$$\frac{P(A|I)}{P(A)} = \frac{ACT(I)/NACT}{TOT(I)/N} \quad \text{(Equation 2.21)}$$

The initial equation form of (Equation 2.19), as expressed by Xia et al., estimates the ratio of active compounds over the total compounds available. Specifically for a given fragment *I*, the ratio can be computed, as in (Equation 2.20), whereby the number of active fragments *I* is over total compounds containing fragment *I*. Extending (Equation 2.19) and (Equation 2.20) yield equation (Equation 2.21) which is similar to Robertson-Sparck Jones' R1(*I*) method.

$$P(A|I) = \frac{ACT(I)}{TOT(I) + 1} \left[\frac{NACT(I)}{N(I)} \right] \quad (\text{Equation 2.22})$$

Hert further demonstrated that by expanding equation (3) above, it can reach the form of the Avidon scheme (Equation 2.22). Furthermore, in the Pipeline Pilot, the log function is used to sum up the fragment weights for a compound, which yield the final probability of activity for it. In regard to Avidon weighting scheme, the sum of the fragment weights is the final score. Based on the equations above (Equation 2.22), Hert et al. (2006) also showed that the SSA weighting scheme R2 is mathematically related to NBC.

NBC's approach has been previously studied under similarity searching and has been applied to screening set selection and the HTS dataset analysis. For instance, Bender, Mussa, Glen, and Reiling (2004) used NBC to classify compounds into two different classes of active and inactive, based on their probability value of in(activity), calculated using Bayes's theorem. Furthermore, Bender, Mussa, Glen, and Reiling (2004) reported on the use of NBC searching method based on atom environments as molecular descriptor to show the best results in their investigation of molecular searching techniques. Their experiment in particular measured the performance of the retrieval rates of active compounds derived from the MDDR database. They concluded that the performance was better in the case of diverse structures and on a large size data set. In another application, Ertl, Roggo, and Schuffenhauer (2008) used SSA to analyse the similarity of molecules to structural characteristic based on Bayes's theorem. This was done by measuring the natural product likeness rather than by measuring the overall activity of compounds.

Glick, Klon, Acklin and Davies (2004) applied NBC to enrich noisy HTS data of a five compounds mixture. The experiment found that NBC successfully ranked the large number of compounds (up to 1 million in minutes) by using a desktop computer. Additionally, in 2006, Glick, Jenkins, Nettles, Hitchings and Davies carried out a further investigation on the use of NBC. In the experiment, Glick et al. (2006) investigated the performance of NBC in the screening of novel active compounds applied to the HTS data with increasing levels of noise. In their analysis, Glick et al., found that NBC can improve the enrichment of high-level noisy HTS data, and is therefore useful and applicable to use NBC for this purpose.

On the other hand, previous studies have shown that NBC unsuccessfully increased the enrichment of High-Throughput Docking (HTD) results. In their study, however, Klön, Glick and Davies (2004) reported that the combination of NBC and consensus scoring approaches successfully improved the HTD data. This can be achieved by employing the consensus scoring after HTD, and later NBC uses the list of compounds gained from the consensus scoring as input to improve the enrichment. For example, Rogers, Brown and Hahn (2005) reported that it is useful to employ NBC with ECFP fingerprints to rank samples in the test set for screening. The advantages include speed, automation, ease and the low cost of the computation required for analysing the HTS data. Langer et al. (2009) also reported that, when active versus inactive categorical data against a given target is available, the NBC can be envisaged. As a result, a large compound collection can be screened in order to differentiate potentially active molecules from inactive ones.

2.12 Performance evaluation and validation for SSA

Only a few documented studies of the SSA have been conducted by researchers since the introduction of the method. Such available studies, however, have explored the evaluation of the SSA performance with respect to structural based molecular descriptors, mainly on 2D fingerprints. This is in contrast to various available studies. These evaluated different machine learning methods, such as the popular similarity searching, data fusion scoring and binary kernel discrimination. Considering the diverse nature of chemical structures, and noting the heterogeneity of compounds in a dataset (consider HTS screened data), the main theme of this study involves the predictive analysis of SSA application. A form of quantification of SSA's performance was thus needed (as for any given evaluation programme), it essentially involves statistical evaluation methods for the weighting schemes in question, or SSA's overall performance in general. A number of statistical evaluation methods have been conducted by various authors (Ormerod, 1989; Cosgrove, 1998; Wilton, 2003) in order to analyse the variability and performance of the weighting schemes.

Table 2.4 highlights a summary of previous works on the performance evaluation of SSA by various researchers. Ormerod et al. reported an extensive study, which compared the effectiveness of a number of established fragment weighting schemes. Their work was originally within the SSA scope of weighting scheme evaluations. In this study, ten individual databases with roughly 100-200 compounds were used to perform SSA with fourteen different weighting schemes. All fragment weights performed less well predictively than

retrospectively. Taking into account both the predictive and retrospective analysis of the study, the R2 weighting scheme is clearly to be the most effective, with SAS consistently being the least effective (Ormerod, Willett, & Bawden, 1989).

Cosgrove et al. (1998) aimed to develop a program called SLASH, which moves a step further in the preparations of the fragments to be used in SSA. For instance, a test was conducted in their study which confirmed the validity of the performance of weighting schemes in the following order: Robertson Spark-Jones R2 > SAF > NSD > SAS. The most recent evaluation of SSA was carried out by Wilton et al. (2003), whose work involved a comparative study of ranking-based methods in Virtual Screening. He employed and analysed individual techniques such as Similarity Searching (SS), Substructural Analysis (SSA) and Binary Kernel Discrimination (BKD). In the study, R1 shows the highest retrieval rate of active compounds derived from the NCI AIDS dataset in the top 1%, and 5% by using BCI and Unity fingerprints respectively. In contrast, R2 shows the highest number of active compounds retrieved from the Syngenta database in the top 1%, and 5% using BCI and Unity fingerprint respectively. In all the evaluation studies, the majority authors highlighted the fact that the most effective results were obtained with the Robertson-Sparck Jones R2 method.

2.13 Conclusion

This chapter discusses the elements in substructure analysis (SSA). Molecular descriptors used to represent the information of chemical compounds are emphasised. Furthermore, several popular weighting schemes used in the SSA searches have been explained. Each of the components of SSA plays an important role in the performance of the search. The next chapter describes the methodology used to develop new weighting schemes in the SSA. This includes the justification on the selection of databases, fingerprints, weighting schemes adopted in this research.

Table 2.4: Summary of past performance evaluation programs on SSA

Description / Analysis / Results						
Researcher(s)	Evaluation	Dataset	Weighting Schemes	Method of study	Performance evaluation	Conclusion/Results
Ormerod et al. (1989)	Evaluation of 14 different fragment weighting schemes for Substructural Analysis.	10 small-sized 2D-fingerprints datasets (a few hundred compounds each) Data Compounds Chemmut 115 Dataprak 196 Nitroso 145 Amino 141 Benzo 115 Ames 114 Nitrocyc 111 Acrid 209 Stero 114 Barbit 160	SAF, SAS, Hodes SD and PR (including analog versions SDI and PRI), Robertson-Sparck R1 to R4, AVID, WT1, WT2.	Retrospective and Predictive	Chi x2, percentage misplaced, error score	Robertson-Sparck Jones R2 method most efficient, but not by a great margin compared to other schemes.
Cosgrove et al. (1998)	Application and evaluation of Substructural Analysis method with the use of a more sensible fragment group (SLASH program).	Atom-pair fragment group. 3 sets of datasets: <ul style="list-style-type: none"> • 25 000 molecules (305 actives, 2 195 inactives and produced 784 458 fragments) • 28 456 molecules (1 822 actives, 26 634 inactives and produced 979 256 fragments) • DAYLIGHT fingerprints for comparison of result. 	Robertson-Sparck R2, SAF, SAS and NSD.	Retrospective and predictive	Decile ranking	Agreement with Ormerod on R2's best performance among the tested weighting schemes. Otherwise, atom-pairs group considered limited in ability to predict activities of new compounds.

Wilton et al. (2003)	Evaluation of SSA (chosen weighting schemes) with other machine learning methods (Binary Kernel Discrimination (BKD) and Support Vector Machine (SVM)).	<p>1) NCI AIDS dataset, containing compounds checked for anti-HIV activity, 2D-fingerprints descriptors (1 129 actives and 34 862 inactives)</p> <ul style="list-style-type: none"> • Training Set (200 actives and 200 inactives) • Test Set (35 591 compounds) <p>2) Syngenta, which contains 132 784 molecules (7 127 actives and 125 657 inactives)</p> <ul style="list-style-type: none"> • Training Set (713 actives and 713 inactives) 	AVID, Robertson-Sparck R1 and R2, WT2.	Retrospective	Top percentage (cumulative) ranking recall	Highlighted BKD's performance efficiency over other methods, but Robertson-Sparck's R2, shown as the best weight function among schemes in SSA.
----------------------	---	--	--	---------------	--	---

Chapter 3

Research Methodology

3.1 Introduction

This chapter describes the methods used to investigate and analyse all experiments conducted in this study. Four experiments were carried out and are outlined in the thesis as follows: the first is a comparison of different existing weighting schemes in SSA (as explained in Chapter 4); second, the use of a genetic algorithm (GA) as a weighting scheme for the SSA (Chapter 5); third, the use of genetic programming (GP) for SSA (Chapter 6); and finally an investigation of the application of data fusion to both the GA and GP-based SSA (as explained in Chapter 7). Thus, the purpose of this chapter is to describe the experimental design (datasets, fingerprints, and evaluation methods) that covers all experiments above. The elements, methods and procedures in all experiments are similar, as are the statistical and evaluation methods used to quantify the performance of each SSA techniques in question.

3.2 Datasets

For all experiments conducted in the study, three large datasets were used to simulate the virtual screening experiments and for the evaluation of the tested methods. The datasets used are as follows: (i) the MDL Drug Data Report database (MDDR); (ii) the World Of Molecular Bioactivity database (WOMBAT) (World Of Molecular Bioactivity, 2007); and (iii) the European Bioinformatics Institute's ChEMBL database (version 18, otherwise known as ChEMBL). The databases above were chosen as they have been used extensively by University of Sheffield researchers, and many other research groups for various chemoinformatics-based studies.

3.2.1 MDDR

The MDDR jointly produced by Accelrys and Prous Science (available from Accelrys Inc. at <http://accelrys.com/products/databases/bioactivity/mddr.html>). The final MDDR database version is volume 32, year 2010 and holds over 180,000 biologically active compounds. Well-defined derivatives are also featured, as well as structures and information on pharmacological classes for the compounds. It was reported that there were approximately

10,000 new compounds added to the database yearly up to the 2010 version. The information of these compounds was gathered from published journals, patent literature, congresses and meetings focusing on released and under development drugs. MDDR used a qualitative method to distinguish between active and inactive compounds. A compound was classified as active if it showed a specific activity, otherwise it was considered as inactive. In this study, MDDR version 1995 was used. It contains 102,540 compounds as shown in Table 3.1(a). For our experiments in the following chapters, eleven activity classes were selected from the MDDR database that were first described by Hert et al. (2004) and that were devised in collaboration with Novartis (Novartis, 2012).

3.2.2 WOMBAT

WOMBAT is a well-known, small molecule, chemogenomics database and was developed by Sunset Molecular Discovery. The database holds structures collected from papers published in important drug-discovery journals, such as the Journal of Medicinal Chemistry. Every year, the database was amended twice, and each time over 10,000 new biological activity structures were added to the database. Unlike MDDR, WOMBAT is quantitative, whereby the activity of a compound is based on the drug potency value. The value is quantified using the $-\log IC_{50}$ scale, also known as pIC_{50} . The scale is introduced by converting the half maximal inhibitory concentration (IC_{50}) value. The IC_{50} is used to quantify the inhibition effectiveness of a molecule affecting chemical process or life in living organisms. The activity of a compound is classified by comparing the activity value with a threshold value. The threshold value was set as pIC_{50} at 5.0. For each activity class, compounds with $pIC_{50} \geq 5.0$ were classified as active in a particular class. Meanwhile, compounds with $pIC_{50} < 5.0$ were classified as inactive and removed from that class. Fourteen activity classes from the WOMBAT dataset, as described by Gardiner et al. (2009), were used throughout the studies presented in Chapters 4 to 7. 14 activity classes were selected from this database. They are similar to the 11 activity classes from the MDDR dataset, with several additional activity classes. The 14 activity classes identified in this study are shown in Table 3.1(b).

3.2.3 ChEMBL

Experiments of the GA and GP-based SSA were mainly carried out on the MDDR and WOMBAT databases, while a verification test was later performed on a larger, more current dataset. For this purpose, ChEMBL database was selected for such test. ChEMBL database is created by the European Molecular Biology Laboratory's – European Bioinformatics Institute

(EMBL-EBI), providing over a million molecules for scientists working in both academia and industry. ChEMBL is an open-data database which is available via <https://www.ebi.ac.uk/chembl/>. The database consists of 2D structures and calculated properties (i.e. log P, molecular weight, and Lipinski parameters). It also features binding, functional and ADMET bioactivities. Much of the ChEMBL data sources were extracted from over 48,000 papers in 47 or more journals, particularly from the Bioorganic and Medicinal Chemistry Letters, the Journal of Medicinal Chemistry and the Journal of Natural Products. EMBL-EBI released several versions of the ChEMBL database, the latest which is version 21, made available on the 29th March 2016 and contains 1,928,903 molecules and 11,019 protein targets.

In this study, ChEMBL version 18 (hereby labelled as ChEMBL) was used to study our GA and GP-based SSA methods, containing a total of 1,352,681 molecules. To exclude any bias in the performance measurement, only activity classes similar to those in WOMBAT and MDDR databases were retrieved. Similar to WOMBAT datasets, the bioactivity for ChEMBL is quantitative, as the molecule is considered to be active if the calculated activity value is more than a threshold value, and is considered inactive for the rest. Molecules were retrieved based on three properties: (i) homo sapiens target organism; (ii) compounds with $pIC_{50} \geq 5.0$; and (iii) compounds with a confidence score equal to 9. In the ChEMBL database, the confidence score is a score value that reflects the target type assigned to a particular assay and the assurance that the target assigned is the correct target for that assay. By using these properties, it was possible to extract 15 activity classes from ChEMBL, as shown in Table 3.1(c).

Table 3.1 contains information on the experimental activity classes from the three databases. These include the name of each activity class, the number of active molecules and the Mean Pairwise Similarity (MPS) values for each class. The MPS value in each row was calculated by comparing each member of an activity class with all of the other members of that class. The MPS values were obtained by quantifying the similarity values between each molecule for each activity class using the standard UNITY 2D fingerprints (Tripos, 2015) and the Tanimoto coefficient. The average of the similarity values was then computed in order to acquire the level of molecular diversity for each activity class (Gardiner et al., 2009; Hert et al., 2004). Based on the MPS values, as shown in Table 3.1(a) and Table 3.1(b), the Renin inhibitors (RNN) was identified as the most homogeneous set, while Cyclooxygenase

inhibitors (COX) as the most heterogeneous. From these, it was possible to classify the activity classes into two heterogeneity sets, namely (1) homogeneous datasets, in which the MPS is equal or more than 0.40; and (2) heterogeneous, wherein the MPS value is less than 0.40. Table 3.1 also provides the number of unique ring systems for each activity class. This is defined by the number of scaffolds found among drugs, also known as the Murcko scaffold. The study of this scaffold was first carried out by Bemis and Murcko (1996), who identified significant ring systems and chains linking two or more rings together. In another study, Brown (2009) found that scaffolds can be used in molecular diversity selection and molecular classification. Figure 3.1 below demonstrates the Murcko scaffold used in this study. A ChEMBL molecule is represented by the Figure 3.1(a). Such molecule is translated to its Murcko scaffold equivalent as shown in Figure 3.1(b).

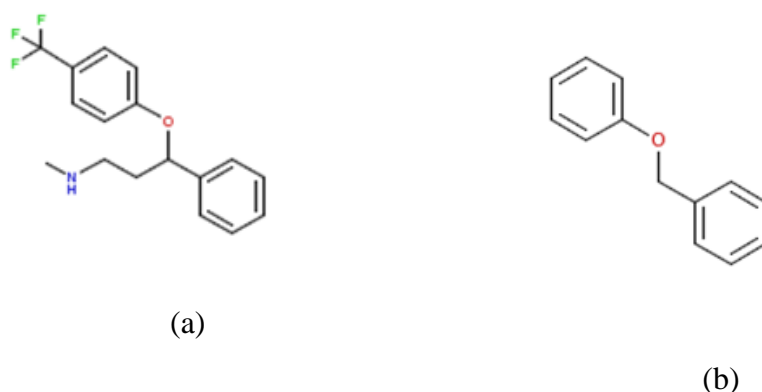


Figure 3.1: (a) A molecule from the ChEMBL database and (b) Its corresponding Murcko scaffold

3.3 Fingerprints

The molecules from MDDR, WOMBAT and ChEMBL datasets were characterised via a dictionary-based fingerprints known as the MDL fragment description. The MDL structural keys used in this study was originally developed for a substructure search (Olah et. al., 2004). The MDL keys consist of 166 bit keysets, based on 166 publicly available MDL MACCS structural keys. The structural keys are important fragments listed in a dictionary used to encode molecules in a bit-string. Each bit is associated with a structural key and it denotes the presence or absence of one of the keys or substructure.

Another type of fingerprint is the Daylight fingerprints which contain indexing of all the structural patterns present in a molecule. A hashing function is applied to generate a set of

numbers from these patterns. Once the numbers for all patterns have been generated, the fingerprints are folded to obtain a fixed bit-string. Folding has several limitations as the same bit may be set by multiple patterns. This bit can potentially become irrelevant and lose information, which is otherwise known as bit collision. Based on the limitation of the Daylight fingerprints, the MDL fingerprints were considered to be more chemically meaningful, as it is recognised as a low resolution, well-established descriptor and has often been used as a standard to evaluate the performance of fingerprints (Heikamp & Bajorath, 2011).

The MDL fingerprints were used to identify the combination of fragment weights to generate the best possible ranking of the molecules in a database. The MDL fingerprints were generated using SciTegic's Pipeline Pilot software to produce structural descriptors or fragments for all compounds. Pipeline Pilot protocols were used to retrieve the MDL fingerprints from the MDDR database. The protocol, as illustrated in Figure 3.2, involves the process of converting a Daylight SMILES notation found in the property list to a molecular representation. The MDL public key fingerprint component was then used to convert molecules into 166-bit MDL fingerprints, denoting the present fragments as '1' and the absent fragments as '0' in each of the 166 fragments.

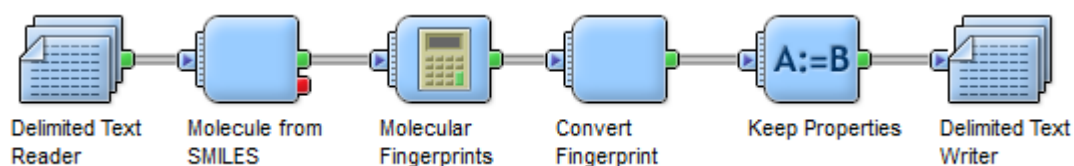


Figure 3.2: Pipeline Pilot workflow for fingerprint

3.4 Test set and training set

Predictive analysis is a standard approach implemented in machine learning methods, as outcomes of such methods often have the possibility of being biased to their input datasets. In a predictive analysis, a dataset is divided into training and test sets, in which the former is used to build an analysis model, while the latter is used to validate the model generated. In the case of SSA, and for each activity class in each database, an analysis is first performed on the training set, generating a set of fragment weights for all fragments. The obtained fragment weights are then directly applied to the test set to calculate the activity or inactivity score for all compounds in the prediction set. In total, three separate analyses were carried out

pertaining to each database used in the experiment. In each dataset, if an active compound for a particular activity class is active, it will be inactive for the other classes. For each activity class, 10% of the active and inactive compounds were selected randomly from each database to create the training set. Such size for a training set was considered for reasons of running-cost effectiveness, whereby 10% size constitutes to a large enough proportion of active and inactive compounds. It is also to ensure that any methods experimented in this research are challenged on its tolerance to the composition of the training and test set relationship.

The remaining compounds were grouped as the predictive test sets. This is described as follows:

- i) Predictive MDDR: 10% of active compounds and 10% of inactive compounds were selected from MDDR as a training set, making a total of 10,254 compounds available for analysis. The test set consisted of the remaining 90% of the dataset.
- ii) Predictive WOMBAT: 10% of active compounds and 10% of inactive compounds were selected from WOMBAT as a training set, making a total of 13,812 compounds available for analysis. The test set comprised of the remaining 90% of the dataset.
- iii) Predictive ChEMBL: 10% of active compounds and 10% of inactive compounds were selected from ChEMBL as a training set, making a total of 135,268 compounds available for analysis. The test set comprised of the remaining 90% of the dataset.

Table 3.1: (a) MDDR, (b) WOMBAT and (c) ChEMBL activity classes considered in this study

Activity class	Abbreviation	Number of Actives	Number of scaffolds	MPS
5HT3 antagonists	5HT3	752	417	0.35
5HT1A agonists	5HT1A	827	450	0.34
5HT Reuptake inhibitors	5HT	359	181	0.35
D2 antagonists	D2	395	258	0.35
Renin inhibitors	RNN	1130	554	0.57
Angiotensin II AT1 antagonists	AT1	943	464	0.40
Thrombin inhibitors	THRM	803	425	0.42
Substance P antagonists	SUBP	1246	586	0.40
HIV protease inhibitors	HIVP	750	461	0.45
Cyclooxygenase inhibitors	COX	636	282	0.27
Protein kinase C inhibitors	PKC	453	171	0.32

(a)

Activity class	Abbreviation	Number of Actives	Number of scaffolds	MPS
5HT1A agonists	5HT1A	592	224	0.40
5HT3 antagonists	5HT3	220	117	0.38
Acetylcholinesterase inhibitors	ACHE	503	220	0.37
Angiotensin II AT1 antagonists	AT1	724	253	0.44
Cyclooxygenase inhibitors	COX	965	220	0.32
D2 antagonists	D2	910	324	0.37
Factor Xa inhibitors	FXA	842	328	0.39
HIV protease inhibitors	HIVP	1128	473	0.44
Matrixmetalloprotease inhibitors	MMP	694	280	0.44
Phosphodiesterase inhibitors	PDE	569	270	0.36
Protein kinase C inhibitors	PKC	142	31	0.57
Renin inhibitors	RNN	474	253	0.59
Substance P antagonists	SUBP	558	186	0.43
Thrombin inhibitors	THRM	421	196	0.42

(b)

Activity class	Abbreviation	Number of Actives	Number of scaffolds	MPS
Serotonin 1a (5-HT1a) receptor	5HT1A	1483	641	0.37
Serotonin 3a (5-HT3a) receptor	5HT3	213	90	0.35
Serotonin transporter	5HT	2447	687	0.34
Acetylcholinesterase	ACHE	739	400	0.36
Type-1 angiotensin II receptor	AT1	106	60	0.52
Cyclooxygenase-1	COX	139	63	0.28
Dopamine D2 receptor	D2	1858	815	0.35
Coagulation factor X	FXA	1502	603	0.39
Human immunodeficiency virus type 1 protease	HIVP	2157	904	0.43
Matrix metalloproteinase-1	MMP	395	157	0.40
Phosphodiesterase 4a	PDE	254	100	0.31
Protein kinase C alpha	PKC	211	76	0.42
Renin	RNN	982	291	0.45
Neurokinin 1 receptor	SUBP	847	316	0.43
Thrombin	THRM	838	472	0.35

(c)

3.5 SSA weighting schemes evaluation methods

As the training sets were mainly used to determine fragment weights (subsequently directly applied to corresponding test set), a score for each molecule from the test set was calculated by adding the weights of those fragments present its MDL fingerprint. The molecules were ranked in descending order based on the calculated score, relative to the measure of the probability that the compound would be active. The best fragment weights were expected to successfully rank all active compounds at the top and all inactive towards the bottom of the ranking. In the past, Cosgrove and Willett described three different statistical measures that can be used to evaluate the rankings resulting from the various SSA weighting schemes (Cosgrove et al., 1998). The techniques are:

- i) Chi squared X^2 : The value representing the difference in distributions of actives and inactives in the top and bottom halves of the ranked list of compounds. If there was no association between the score calculated for a structure and its activity, 50% of the actives would be expected to appear in the top half of the ranked list with the remaining 50% in the bottom half. Similarly, 50% of the inactives would be expected to appear in the top half of the ranking and 50% in the bottom half. The observed distribution could then be compared with this null distribution using the calculated X^2 values.
- ii) % misplaced or decile ranking: The number of active compounds in each decile of the activity ranking is identified and two rankings compared by the distribution of active compounds within the deciles. If a dataset has been analysed perfectly, the first part of the ranking will consist just of actives and the second part just of inactives. In general, however, some of the actives will be displaced and occur in the lower part of the ranking, with some of the inactives appearing in the upper part of the ranking. For example, consider 10 compounds, 6 of which are active (A) and 4 of which are inactive (I). The perfect situation would be

A A A A A I I I I,

for the compounds when ordered in descending score value. If the actual ordering gave

I A A A A I I I A,

then the percentage misplaced is 2/10, i.e., 20%.

- iii) Error score: Takes account not only of the number of compounds misplaced but also the positions of the misplaced compounds in the ranking, i.e., how far their rank is

from the dividing line between the active and inactive compounds in the ideal ranking. Consider a possible ranking of the small dataset

III AAA I AAA .

Thus, for this ranking, the error for the inactives is 12, i.e, 5+4+3, and for the actives is 9, i.e, 4+3+2. The Error score is the given by the mean error, i.e., by (9+12)/10. Thus, the Error score for this ranking is 2.1.

However, the Enrichment Factor (EF) was used to compare the rankings resulting from the weighting schemes investigated in our study.

3.5.1 Enrichment Factor (EF)

In evaluating the SSA search method, the effectiveness aspect was focused on, which quantifies whether the actual output meets the desired output or otherwise (Edgar et al., 2000). Several techniques can be applied to evaluate the effectiveness of existing SSA weighting schemes and developed evolutionary algorithms based on SSA. Machine learning experiments are often evaluated using area under the curve (AUC) values, i.e., the area under a receiver operating characteristic (or ROC) curve. However, this performance criterion is less appropriate for evaluating virtual screening experiments since it considers the entire ranking of a database when calculating the effectiveness of a ranking. In fact, methods of virtual screening require only the analysis of a small fraction of the molecules that occurs at the top of the ranking to be considered for further biological screening. Rather than using AUC values, the screening performance was hence measured by the number of actives for the top 1% of the ranked test set (i.e. 1% enrichment value).

The enrichment values are computed as the actual obtained number of active compounds at a specific cut-off value, divided by the number of actives that would be retrieved if compounds were picked from the database at random. Thus, the enrichment factor compares how much the retrieval rates are better than the random model-based retrieval. For instance, for a given group of 750 active compounds, and considering that the top 1% was picked at random, the expected number of active compounds would be 7.5. Hence, a random model-based enrichment of active compounds at the top 1% of the samples would yield 7.5 active compounds. Again, this is based on the random assumption. Consequently, to compute the enrichment factor for a given trend, consider the case in which 200 active compounds are found in the top 1%. The enrichment factor would be the actual obtained number of active

compounds divided by the number of actives expected based on random selection, which is 200/7.5. Therefore, in the top 1% of the dataset, the enrichment value is denoted as 26.7 in this case.

3.5.2 Analysis of diversity

In this study, apart from the enrichment factor, the performance of the virtual screening was also measured by identifying the number of distinct Murcko scaffolds at the top 1% actives. This serves as a simple measure of structural diversity. Diversity analysis was conducted on the experimental results to quantify its ability to identify novel bioactive compounds from a diverse space of possible compounds. The analysis obtained was compared to the results obtained from evaluating weighting schemes. The top 1% rank positions of compounds from the test set in the analysis were chosen for diversity analysis. The weighting scheme with the highest retrieval rate, diversity and unique scaffolds for each activity class were identified. Mean diversity analysis is calculated as the complement of mean similarity analysis using the Tanimoto coefficient. The equation retrieves fragment presence information of two compounds by using 2D fingerprints (defined as 166-bits MDL 2D fingerprints in this case) and computes a value associated with the Tanimoto coefficient, given by Equation 3.1:

$$Diversity_{AB} = 1 - \left[\frac{c}{a + b + c} \right] \quad (\text{Equation 3.1})$$

where a is the number of "on" bits in compound A,

b is the number of "on" bits in compound B,

c is the number of "on" bits in both compounds A and B.

Based on the above equation, the Tanimoto coefficient calculates the ratio between an intersecting set and the union set. The ratio of one signifies maximum diversity and the ratio of zero denotes maximum similarity. This method was chosen as it has been used extensively in the past for identifying / singling out structural similarities. In this analysis, it was possible to identify the number of unique scaffolds of ranked compounds retrieved in the top 1%. These were obtained through the use of Pipeline Pilot software. In order to generate a variety of unique scaffolds from the retrieved compounds, the *BemisMurckoAssemblies* scaffold type was utilised, as described by Bemis and Murcko (1996).

3.5.3 Statistical tests

Two different statistical sets of methods were carried out to evaluate the statistical significance of the experimental results. The statistical tests were used to identify the performance of weighting schemes outlined in Chapters 4, 5 and 6. They were also used effectively on the performance of data fusion methods application discussed in Chapter 7.

3.5.3.1 Kendall's W analysis

The degree of agreement between the weighting schemes rankings is measured by calculating the Kendall coefficient of concordance, known as the W value (Siegel, 1956). This coefficient provides a means of quantifying the degree of association between k variables or k sets of rankings of similar objects. Accordingly, Kendall's W calculates the agreements between rankers as it evaluates and ranks a number of subjects according to particular characteristics. The concept is that n subjects are ranked (0 to $n-1$) by each of the rankers, and the statistic evaluates how much the rankers agree with each other. Kendall's W ranges from 0 to 1, where 0 indicates no agreement and 1 indicates complete agreement. This analysis was performed for all of the experimented methods conducted in chapters 4 to 7.

Specifically, the weighting schemes from each database are ranked in decreasing order of effectiveness of virtual screening for a specific activity class. This is repeated for each class so that there are e.g. 11 rankings for the MDDR dataset. The degree of agreement between the rankings in the top 1% of the ranked compounds is measured by calculating the Kendall Coefficient of Concordance, W . This coefficient provides a means of quantifying the degree of association between sets of rankings of the same objects. If there is an agreement between the rankings of the weighting schemes, it can be concluded that there is a statistical significant result for the null hypothesis, H_0 . This predicts the probability that the rankings are not associated, and can thus be rejected. In this analysis, 0.001, 0.01, 0.1 and 0.5 were selected as the significance level. Therefore, if the probability p value is equal to or less than 0.001, it is then necessary to reject the null hypothesis and then can give overall ranking. However, if the p value is more than 0.01, then the computed results are considered insignificant. The equation that has been used to compute the degree of variance among the ranks is given by Equation 3.2:

$$W = \frac{12 \sum R_i^2 - 3k^2 \times N (N + 1)^2}{k^2 \times N (N^2 - 1)} \quad (\text{Equation 3.2})$$

where k is the number of ranks; for example, 11 activity class in MDDR dataset;
 N is the number of objects being run; for example 9 weighting schemes were
evaluated in this study;
and R_i^2 is the sum of the squares sums of ranks for each of the N objects.

The significance of the W was computed using a X^2 distribution, with a degree of freedom $df = N-1$, for which the equation (Equation 3.3):

$$X^2 = k(N - 1)W \quad (\text{Equation 3.3})$$

If the size of the samples is larger ($N > 7$) then the chi square and the probability p values were identified by referring to the chi square distribution table; otherwise, the table of critical values was used to identify the probability (Siegel and Castellan, 1988). Whenever W is larger than the critical values, this result would be considered significant and thus the null hypothesis would be rejected.

3.5.3.2 Wilcoxon signed rank test

In addition to Kendall's W test, the Wilcoxon signed rank test was also performed in Chapters 5 to 7. This nonparametric test is designed to evaluate the difference between two conditions where the samples are correlated (Ott & Longnecker, 2015). In this study, the test was used to compare the performance between weighting schemes. The statistical test was used in this study to measure paired samples to identify whether two weighting schemes are statistically significantly different or whether one of them is better than another. Rankings of differences of larger and smaller values are used in conjunction with a test statistic W . In this test, all the top 1% active recalls of activity classes in all datasets (i.e. MDDR, WOMBAT and ChEMBL) are collected and their significance of difference tested using the Wilcoxon signed rank test. A null hypothesis, H_o , is defined as where the median difference is zero. This means that our default assumption is that both results of weighting schemes are significantly identical. The alternate hypothesis, H_1 , is defined as the median difference being positive at a significance level of $p = 0.01$. The ranks are given the positive (+) or negatives (-) signs of the

corresponding deviations. The computation of the Wilcoxon signed rank test involves five steps, these are explained below:

- i) The magnitude of difference between two weighting schemes is determined by calculating the signed difference $DIFF_i$ of each pair, X_i and Y_i , in observation. The difference is calculated by subtracting Y_i from X_i (i.e. $DIFF_i = X_i - Y_i$). This gives more weight to a pair which shows a larger difference than a smaller one. If the two scores of any pair N are equal, where $DIFF_i = 0$, then such pairs are discarded from the analysis. N is equivalent to the number of pairs whose differences show a sign (i.e. + or -) where ties are ignored, in this study.
- ii) Ignoring the sign (i.e. -, +) of the difference, all the $DIFF_i$ values are ranked subsequently, with 1 being the smallest. This rank (i.e. R_i) is used to compute the test statistics.
- iii) Each rank (R_i) is labelled according to its sign of difference ($DIFF_i$) to indicate which rank is positive (+) or negative (-) from the $DIFF_i$. The signed rank is used to calculate the sum of positive and negative ranks. The positive ranks are calculated by summing the entire positive (+) ranks, while the negative ranks are calculated as the sum of the negative ranks.

$Ri_{positive}$ = the sum of the positive $DIFF_i$'s ranks

$Ri_{negative}$ = the sum of the negative $DIFF_i$'s ranks

- iv) To compute the probabilities H_0 being true, the W value was calculated to evaluate the hypothesis, since the size of N used is less than 20, then the W value was used to evaluate the hypothesis, if $N \geq 20$ the z value can be used. Calculate W by comparing the value of $Ri_{positive}$ and $Ri_{negative}$, the smallest compared value is assigned as W . For instance, $Ri_{positive} = 8$, while $Ri_{negative} = 15$, thus, W , the smaller of $Ri_{positive}$ and $Ri_{negative}$, is 8.

$W = \text{Smaller of } Ri_{positive} \text{ and } Ri_{negative}$

- v) Finally, it is necessary to refer to the table of critical values of W . This serves to gauge the level of rejection of the test statistics in order to arrive at the alternate hypothesis. Using the information of the number of differences, N ; a probability value with the

lowest value of the significance level of 0.01, rejects the null hypothesis H_0 , if the value of W is less than or equal to the critical value of $W_{critical}$ (Ott & Longnecker, 2015). For example, to determine the critical value of $N = 11$ for the significance level $\alpha = 0.01$ with the test statistic is $W = 3$, looking at the critical value table, the critical value of $W_{critical}$ is 5, thus the null hypothesis is rejected because the calculated W is less than the $W_{critical}$, $3 < 5$.

If $W \leq W_{critical}$ then reject H_0

3.6 Hardware

Several computer machines were made available over the course of this research, in order to perform the investigated methods and also to gauge their computing performance. The computer hardware specifications are listed in Table 3.2. It is stressed that programming and development of methods investigated in this thesis was completely done via MATLAB, a rapid application development (RAD) productivity software package with a custom high level language tailored for the software. Therefore, there are clearly other potential applications (such as code redundancy reductions or even a change to optimised low-level languages like C++) which can reduce computational cost and increase programming efficiencies, but this was not explored further. MATLAB also features a parallel computing toolbox which includes the graphic unit acceleration option. These options were also not explored during this research; therefore, the advantage of multi-core processes or graphic unit additions on evolution-based SSA at the time of writing could not quantify. At the very least, the runtime benchmark analysis presented for both GA and GP programs can indicate the level of resource expectation when screening a multitude of dataset sizes. It may also be possible to use different parameterisations to the ones used in these setups.

Table 3.2: Computer’s hardware specification used to run the GA-based SSA. (a) Server setup; (b) Multimedia-intensive workstation; (c) Office-level workstation; and (d) Laptop setup

(a)

Machine ID	Server_1
Make / Model (if applicable)	Linux server
Processor	Intel (R) Xeon (R) CPU E5-1650 v3 % 3.5 Ghz (12 cores)
Memory (RAM)	32.0 GB
Operating system	Red Hat Linux Enterprise

(b)

Machine ID	Workstation_1
Make / Model (if applicable)	Custom built
Processor	AMD FX (TM) 8350 8 Core 4.00 GHz
Memory (RAM)	16.0 GB
Operating system	Windows 7 64-bit Professional, Service Pack 1

(c)

Machine ID	Workstation_2
Make / Model (if applicable)	Dell / OptiPlex 320X
Processor	Intel (R) Core (TM) i5 CPU 650 @ 3.20 GHz
Memory (RAM)	4.0 GB
Operating system	Windows 7 32-bit Enterprise, Service Pack 1

3.7 Conclusion

This chapter has described the method used in this investigation of the SSA weighting schemes using evolutionary algorithms. This chapter also discussed the databases used and the statistical methods applied to evaluate the effectiveness of the developed evolutionary algorithms-based SSA compared to the existing SSA weighting schemes. Thus, the following chapter describes the evaluation of different existing weighting schemes in SSA using the three large datasets, i.e. MDDR, WOMBAT and ChEMBL as discussed above.

Chapter 4

The Comparison of Different Weighting Schemes in Substructural Analysis Using Large Datasets

4.1 Introduction

Several weighting schemes are readily available in SSA, as discussed in Chapter 2. A majority of the schemes are relatively similar to one another in terms of the equations. In the current literature, studies have been conducted on the analysis of the effectiveness of such weighting schemes. Their evaluations, however, only focused on small sets of compounds (amounting to hundreds to a few thousands only) and in limited activity classes (Ormerod, 1989; Cosgrove, 1998; Wilton, 2003). In this chapter, an updated analysis of the effectiveness of the established SSA weighting schemes for measuring the predictive performance of a given biological activity class using large datasets was presented. Predictive analyses using a randomly selected portion of the dataset to analyse the predictive performance of SSA in the test set, based on applications in the training sets were conducted. Experiments to analyse the effectiveness of the established SSA weighting schemes in determining the activity of the molecules were also carried out.

4.2 Experimental details

For the experiments, three large datasets were used, namely WOMBAT, MDDR and ChEMBL. The details of the datasets and the analysis method applied in this study are explained in Chapter 3. The molecules in MDDR, WOMBAT and ChEMBL were represented by the MDL fragment descriptor (contains 166 bits). More information about the MDL fingerprints and explanation on the workflow used to generate the MDL fingerprints are described in Chapter 3.

4.3 Experimental procedure

Figure 4.1 illustrates the procedure used in conducting the comparative experiment. Here, the SSA method was applied for predictive analyses, consisting of three general steps. First, the weight calculation of the individual fragment was carried out using ten different weighting

schemes. Second, the process involved identifying score of each compound in the test set. Finally, it is necessary to rank the compounds in the test set by descending score value order.

4.3.1 Weighting schemes

Given the abundant choice of various weighting schemes available under SSA, ten weighting schemes were selected for evaluation on the basis of their performance in previous comparative studies by different authors. The selected weighting schemes are: SAF and SAS, introduced by Cramer et al. (1974); the relevance weighting schemes R1, R2, R3 and R4 introduced by Robertson and Sparck Jones (1976); Avidon et al.'s (1982) derived weighting schemes, particularly AVID, WT1 and WT2; and NBC method from Pipeline Pilot. A detailed explanation of these weighting schemes is described in Chapter 2.

4.3.2 Benchmarking SSA performance against NBC Pipeline Pilot

In addition to the SSA weights, the implementation of the NBC-based Pipeline Pilot scheme as an additional benchmark against the SSA weighting schemes was also included. The NBC method was performed using the Pipeline Pilot software and the procedure is described in detail in both Figures 4.2 and 4.3. For the initial NBC model using training sets, the predictive sets as input and the NBC model is generated using the "*Learn Good Molecules*" option. It is used in order to examine and distinguish the "*good*" and "*baseline*" compounds (Figure 4.2). The generated NBC models were used to screen a new candidate of the input test set and sorted the test set using the model's value using the "*Enrichment Plot Viewer*" option (Figure 4.3). The plot is used to identify how rapidly hits are found in the sorted compounds list. The NBC results were compared to all the SSA weighting schemes for benchmark analysis.

4.4 Analysis of SSA weighting schemes

The performance of the weighting schemes was evaluated via enrichment factor analysis and statistical-based methods (enrichment curve and Kendall's W) to highlight the prediction effectiveness for each scheme. The measures are based on the number of active molecules retrieved in the top 1% of the test set, as explained in Chapter 3.

Table 4.1 lists the total number of actives in the test set and the number of actives in the top 1% using the selected weighting schemes for every designated activity class of three databases. The equivalent enrichment factor of the actives is also summarised in Table 4.2. The best result for each activity class is lightly shaded.

The entire comparable results were obtained with the MDDR dataset as presented in Tables 4.1(a) and with WOMBAT dataset as shown in Tables 4.1(b). In the predictive experiment, all weights retrieved more than 10% of active compounds, except SAS weight, which, can be seen in 5HT3, 5HT1A, 5HT, D2, RNN and HIVP classes. In general, more than 50% of the active compounds were retrieved within the top 10% of the ranked compounds in all activity class analysis. Similar observations can be seen in cases with the ChEMBL dataset. Tables 4.1(c) shows that the four relevance weighting schemes show good retrieval rates for 1% searches compared to SAS, which is shown to be significantly lower in terms of performance.

4.4.1 Enrichment curve analysis

The effectiveness of screening via SSA is illustrated diagrammatically using enrichment curve plots for all SSA weighting schemes. The cumulative active compounds recall rates based on the enrichment factor of actives in the ranked compounds for each weighting scheme were plotted. Figures 4.4 to 4.8 show several examples of enrichment plots from predicted test set instances for the first 10% ranked compounds, based on five selected activity classes from MDDR, WOMBAT and ChEMBL datasets. These are the 5HT3, COX, D2, RNN and PKC classes respectively. The figures also include a random and ideal model, where the former serves as the baseline guide for the active compound retrieval rate distributed equally throughout the entire ranking. The latter shows the best possible performance of active compound retrieval at the top ranks. The value of the enrichment factor within the top 10% levels' samples are plotted and the number of active compounds in the top 1% of each of the eleven MDDR, fourteen WOMBAT and fifteen ChEMBL activity classes for all ten weighting schemes are captured for further analysis (Kendall's W test).

Based on the enrichment curve analysis for provided plots, it can be seen that the R3 and R4 schemes are shown to outperform other weighting schemes in most classes tested over the three databases. Clear examples are from the WOMBAT 5HT3 (Figure 4.4b), MDDR D2 (Figure 4.6a) and ChEMBL D2 (Figure 4.6a), MDDR, WOMBAT and ChEMBL RNN (Figure 4.7a-c), and WOMBAT PKC (Figure 4.8b) classes. There are cases, however, when the NBC scheme surpasses the R3 and R4 scheme, as shown in the MDDR 5HT3 (Figure 4.4a) and MDDR COX (Figure 4.5a). R1 and R2 schemes were also shown to come on top against the R3 and R4 schemes in the ChEMBL 5HT3 (Figure 4.4c) and WOMBAT D2 (Figure 4.6b). A majority of R1, R2 and NBC schemes generally show mid and occasionally top-tier performance results in most classes tested, while SAF, WT1, AVID, WT2 schemes exhibit

mediocre performance schemes for a majority of classes in all three databases. The SAS and AVID schemes are shown to be consistently the worst performing schemes in most classes. It is worth noting that the AVID schemes managed to retrieve just less than 50% active compounds at the top 10% ranked percentile. These can be seen in examples of MDDR and ChEMBL 5HT3 (Figures 4.4a and 4.4c), MDDR COX (Figure 4.5a), MDDR and WOMBAT D2 (Figure 4.6a-b). Note that a particular ChEMBL RNN activity class (Figure 4.7c) indicates that the AVID scheme is disappointingly lower than even the random model threshold up to the top 10% ranked list.

In general, more than 50% of the active compounds were retrieved within the top 10% of the ranked compounds in all the classes of all databases analysed, except for the SAS scheme. It was also observed that the four relevance weighting schemes (R1, R2, R3 and R4) exhibit good retrieval rates for 1% searches compared to other available schemes.

Based on the enrichment curve analysis for MDDR, WOMBAT and ChEMBL datasets as discussed above, two observations can be made. First, all the relevance weighting schemes based on Robertson-Sparck Jones work were shown to consistently produce the highest, or among the highest distribution rate. These are clearly shown in Tables 4.1 and 4.2 for almost every activity class from the three databases. Second, based on the results presented in Figure 4.9, to 4.11, the homogeneous activity class, i.e., RNN, AT1, PKC, MMP and SUBP gave excellent results for actives retrieved in MDDR, WOMBAT and ChEMBL datasets respectively, using the SSA R4 weighting function.

4.4.2 Kendall's W analysis

Kendall's W tests were carried out based on the number of active compounds in the top 1% of the ranked compounds of each dataset. Table 4.3 shows the list of ranks for each weighting scheme for each activity class of MDDR, WOMBAT and ChEMBL test sets. The weighting scheme with the number of actives retrieved for each activity class is recorded. From these values, the rank for each weighting scheme was noted for the purpose of identifying the performance of the schemes. Thus, in these studies the weighting scheme subjects are ranked (0 to $N-1$), so the rank value is from 0 to 9. For example, MDDR, R3 produces the highest number of actives retrieved in the ATI activity class. Therefore, R3 was ranked as number 9 and the weighting scheme, SAS, with the lowest number of actives in the class will have rank number 0. The final two columns on the right in Table 4.3 are the mean of the rank obtained of

all activity classes for each weighting scheme. This produces a listed rank position for each scheme. In the case of tied rank positions, a correction factor is used to resolve such disputes. The correction factor is defined as shown in Equation 4.1.

$$T_j = \sum_{i=0}^{g_j} t_i^3 - t_i \quad (\text{Equation 4.1})$$

From the equation, t_i is the number of tied ranks in the i^{th} groups of tied ranks; while g_j is the number of groups of ties in the set of ranks for judge j . The correction factor is computed by adding over all groups of ties found in all j judges. For the tied rankings, their positions will be re-assigned based on the correction factor calculated above (Siegel and Castellan, 1988).

The rankings of the weighting schemes are measured in three steps. First, the ranking of the weighting schemes in the top 1% of samples are identified, these are based on the number of active compounds retrieved from the enrichment curves of each activity class. Second, the degree of agreement between the ten rankings is computed. This produces the list of ranks for each weighting scheme for each activity class of three databases (MDDR, WOMBAT and ChEMBL), as shown in Table 4.2. From these values, the value of W in the top 1% of the ranked compounds, together with the probability value, is computed. Thirdly, the mean value of rank for each weighting scheme is calculated. The rank for each weighting scheme is generated by the Kendall's W test, based on the number of actives in the top 1% over all the activity classes of each database, as presented in Table 4.2. From the results, the overall ranking of the weighting schemes in the top 1% for the predictive analyses are identified.

Predictive analysis

The determination of the Kendall's W test for the eleven, fourteen and fifteen activity classes in MDDR, WOMBAT and ChEMBL databases, respectively, reveals that there is a strong agreement between the ten rankings of the weighting schemes. It may be concluded that there are statistically significant differences between the performances of the various weighting schemes, which is significant at the 0.01 level. The determination of the Kendall's W test for the MDDR can be seen in Table 4.2(a). From the data in Table 4.2, it is apparent that there is a strong correlation between the ten rankings of the schemes using 1% of the MDDR, WOMBAT and ChEMBL databases. The computed value of W for the MDDR is 0.75, and

this yields a value of $X^2 = 74.28$ at the $p < 0.01$. The overall ranking of the weights for the MDDR database is:

$$R4 > R3 > R2 = NBC > R1 > WT1 > SAF > WT2 > AVID > SAS$$

For the analysis using the WOMBAT database, Table 4.2(b) also provides the rank position of the ten weighting schemes computed using the Kendall's W test, which gives the value of $W = 0.68$, and $X^2 = 85.47$. This test is highly significant, with $p < 0.01$. The resulting ranking of the ten SSA weighting schemes is:

$$R4 = R3 > R1 = R2 = NBC > AVID > SAF > WT2 > WT1 > SAS$$

Similar observations can be seen when using the ChEMBL dataset. Table 4.2(c) presents highly significant at the 0.01 level and it reveals a strong association between the ten rankings of the schemes of all cases. The computed value for W is 0.54, which yields a value of $X^2 = 75.99$. The overall ranking of the weighting schemes for the ChEMBL dataset is:

$$R4 > R3 > NBC > R1 > R2 > WT2 > SAF > AVID > WT1 > SAS$$

Overall, based on the Kendall's W results of the three databases, as discussed above, it is shown that all the results are somewhat similar to each other, with the four relevance weights, $R3$, $R4$, $R2$ and $R1$, at the top of all rankings. SAF , $WT2$, $AVID$ and $WT1$ performs well; but SAS shows a poor performance. Furthermore, based on the Kendall's W test, it may be concluded that there are statistically significant differences between the performances of the various weighting schemes in this study.

In addition, when considering the mean rank for all databases (i.e., MDDR, WOMBAT and ChEMBL), Table 4.3 shows a highly significant agreement of performance of the various SSA weighting schemes in each database at $p < 0.01$ level. The calculated value for W is 0.94, with a value of $X^2 = 25.34$. The resulting ranking of the SSA weighting schemes is:

$$R4 > R3 > NBC > R2 > R1 > SAF > AVID > WT2 > WT1 > SAS$$

4.4.3 Analysis of the SSA R4 fragment weights

The relevance weighting schemes, termed R4, as originally derived by Robertson and Sparck-Jones (1976), was modified by Ormerod et al. (1989). It serves to accommodate their use for the nature of chemical substructures and was used for comparison with the GA weights. Recall that the R4 weight is given by the Equation 2.13. The equation demonstrates that the R4 weight evaluates the ratio between the “active odds” for a particular fragment. For example, the proportion between the number of active compounds in which the fragment is present and the number in which the fragment is not present (or absent) and its “inactive odds”. Robertson and Sparck-Jones (1976) stated that R4 weights yield a comparatively better performance than other relevant weights since full, active and inactive compound information is available in the equation. $ACT(I)$, $NACT$, $INACT(I)$ and $NINACT$ particulars are known for each fragment. On the other hand, the R4 weight was created based on Assumption 2 and Ordering 2 (as discussed in Chapter 2). This method is more realistic and reliable compared to Assumption 1 and Ordering 1. The R4 equation shown above incorporates the logarithm function as it is based on the summation of weights. The logarithm is undefined for a zero value, and hence a prior value needs to be assigned in place of any term in the equation equalling zero (Robertson and Sparck-Jones, 1976). Following Ormerod, in order to prevent arithmetic problems, any zero-valued component in the formula was substituted with the prior value of 0.0000001.

To analyse the effect of the prior value m to the R4 weight, fourteen fragments were chosen that contain the following features: (i) Fragments that represent the low, moderate and high active compound frequency, and (ii) Fragments that represent a low and high presence in the overall compound dataset. Table 4.4 shows the results of SSA R4 weighting scheme applied to the training set of COX activity class from the MDDR database, based on the predictive analysis. This consists of 10,253 compounds (N), of which 63 are active ($NACT$). A number of SSA R4 weighting values for each fragment are presented; taken into account are the different prior values of m used, which were 0.0000001, 0.01, 0.05, 0.1 and 0.5. For instance, the weight of fragment ID 1 in the fourth column from the right in Table 4.4 shows that the SSA R4 weight equals 2.2089 when zero-valued components were exchanged with 0.01. It should be noted that the fragments above carry with them zero-valued components in the equation ($ACT(I)$, $TOT(I)$ and $INACT(I)$ for example). The zero values depicted in bold and shaded, as listed in Table 4.4. The results were then reordered with respect to the $ACT(I) / TOT(I)$ ratio, which defines the probability of the compounds comprising a particular fragment being active.

To analyse the weight, the $ACT(I)/N$ (the probability that an active compound contains the fragment) and $TOT(I)/N$ (the probability that a compound contains the fragment) ratios were also computed. From the Table 4.4, for the case where $m=0.0000001$, negative weights were obtained when $ACT(I)/N < 0.1$. SSA R4 also gave a negative weight to fragments ID 19, 13, 17, 74, 67, 127 and 154; it is presumed that the probability the fragments are present in an inactive compound is higher.

Further observations from Table 4.4 show that fragment ID 17 gave a much lower weight compared to all other fragment weights when assigning m as 0.0000001; instead of either 0.01, 0.05, 0.1 or 0.5 to all the zero-valued components. In this case, components $ACT(I)$, $TOT(I)$, and $INACT(I)$. Thus, molecules which contain fragments ID 17 will be pushed further down the ranking when $ACT(I)$ is substituted with 0.0000001, compared to when it is replaced by 0.5. For fragment ID 166, the replacements of any zero-valued element with a smaller prior value of 0.0000001 causes it to impose a larger R4 fragment weight. This result is confirmed by the fact that fragments ID 166 should indeed be emphasised, since the probability that an active compound containing this fragments ($ACT(I)/N$) is higher than for other fragments. From these results, it can be concluded that a prior value of 0.0000001 improves the ranking of compounds, especially when there are zero-valued elements involved in the determination of the fragment weights.

Based on the observations in Figure 4.12, a different m value produces a varied degree of smoothing effect on the term weights computation. A large m value of 0.1 caused larger smoothing of the R4 weights compared to the small m value of 0.0000001. This can be seen in Figure 4.12, where the figure shows the distribution of fragment weights obtained using SSA R4 weighting. This analysis confirms the perception that the prior value m acts as a smoothing correction to the relevant weighting scheme. This perception has been expressed in the information retrieval literature by Hiemstra (2001). For the case of the SSA R4 weighting scheme, however, applying larger than necessary smoothing correction values reduces the capability of SSA R4 to differentiate the active compounds from their pool. As a consequence, this may reduce the performance of the SSA R4 scheme. This is also observed in Table 4.5 where the mean and standard deviation values of the overall 166 fragments weights for prior values of 0.0000001, 0.01, 0.05, 0.1 and 0.5 were computed. It is evident that the assignment of higher prior values implies a smaller variation of individual fragment weights, hence the weaker influence of such individual fragments on the determination of a compound's overall

score. When comparing SAF weights with SSA R4 weights, a variation between individual SAF fragment weights is smaller than individual SSA R4 weight. The SAF weight, therefore, unsuccessfully indicates a fragment with a strong association with the activity and a fragment which is unrelated to activity.

When comparing SAF weights with SSA R4 weights, the SAF weights failed to incorporate information regarding the difference in the number of active or inactive compounds containing a particular fragment. For example, the fourth column in Table 4.4 depicts the SAF weight for fragment ID 166 to be 0.0062 when $ACT(I) = 63$, and the SAF weight for fragment ID 164 to be 0.0064, when $ACT(I) = 62$. In this case, the latter weight should receive a smaller score than the former, since the latter fragment appears in 62 active compounds, while the former fragment, appears in 63. The difference is higher by 1 than the latter fragment, i.e. Fragment ID 164. This shows that Fragment ID 166 indicates a stronger association to the activity than Fragment ID 164. It can also be seen from the Table 4.5 that SAF fragment weights show a small variation between individual weights, which is 0.06 when compared to the individual SSA R4 weights, where the variance for the SSA R4 weights is 7.66. This therefore indicates that the SAF weights were unsuccessfully differentiated between a fragment that had a strong association but was unrelated to the activity.

Overall, the above tests were carried out as a means of observing the effect of varying prior value and to quantify its influence on the SSA R4 method. Our tests confirmed that the prior value of 0.0000001 chosen by Ormerod improved the performance of SSA R4 scheme, which was subsequently applied to the SSA R4 analysis in the experimentations.

4.5 Discussion

Previous studies have determined that the most effective SSA weight function is R2, while SAS has consistently been the least effective scheme (Ormerod, 1989; Cosgrove, 1998; Wilton, 2003). Ormerod (1989) also included the R4 scheme in her evaluation of various SSA weighting schemes, but still found R2 to be generally better. Based on our predictive study using the MDDR, WOMBAT and ChEMBL databases, however, it was found that the Robertson-Sparck Jones's R4 scheme is clearly superior to other schemes when used in both heterogeneous and homogeneous datasets.

The results were obtained based on the number of actives in the top 1% and mean rank in the predictive studies. From the results, R4 weights consistently retrieve the highest number of active compounds in the top 1% compared to other SSA methods. However, the result was not much different compared to the other relevance weights (i.e. R3, R1 and R2). This is due to their weighting equations, which use all the information about the dataset; they incorporate all five variables, the presence and absence of fragments information (i.e. $ACT(I)$ and $INACT(I)$), and the number of compounds with regards to active, inactive and the total (i.e. $TOT(I)$, $NACT$, $NINACT$).

Furthermore, R3 and R4 schemes exhibit very similar results in retrieval rates. This is because both weights are related to the use of the term 'odds', in which R3 represents the ratio between the active 'odds' for the fragment (the ratio between the number of active compounds when it is not present and the number in which it is present). R4 meanwhile, represents the ratio between the presence of fragments in active odds and the absence of fragments in inactive odds.

Based on the experiments, the R1 scheme shows a close relationship to R2, in fact they are very similar in performance. Both weights use the information of the presence of fragment I ($ACT(I)$) in active and inactive compounds ($NACT$, $NINACT$) in their equations. R1 and R2 are also related to the use of proportion, where R1 evaluates the ratio of the proportion of present fragments in active compounds to the proportion of the present fragments in the entire set of compounds. R2, on the other hand, represents the proportion of active compounds over inactive compounds. From the data in Table 4.2, a similar performance can be observed between NBC Pipeline Pilot and R2 schemes for both MDDR and WOMBAT classes. There are, however, marginal differences between the two schemes when performed on ChEMBL classes. These results support the previous research finding that the SSA weighting scheme R2 is mathematically related to NBC Pipeline Pilot as reported by Hert et al. (2006).

These relevance weights are based on two independence assumptions (Assumption 1, Assumption 2) and ordering principles (Ordering 1, Ordering 2) as discussed in the literature chapter. Robertson and Sparck Jones (1976) state that Assumption 2 is more realistic than Assumption 1 and Ordering 2 is correct while Ordering 1 is not. Moreover, among the relevance weights, R4 is most likely to be the best since the weight is based on Assumption 2

and Ordering 2. From these analyses, it was found that R4 yields the best results; therefore these results support the claim made by Robertson and Sparck Jones, as stated above.

Another general trend is observed where there are similar trends in retrieval between weighting schemes SAF, WT1, WT2 and AVID. This can be attributed to similarities in their weighting equations and the use of variables as well. For instance, the SAF equation is just the proportion of active compounds containing fragment *I* ($ACT(I)$), against the total number of compounds containing fragment *I* ($TOT(I)$). Other schemes, such as Avidon's WT1 for example, utilise the fragment's presence information in inactive compounds, namely the variable $INACT(I)$. The WT2 scheme incorporates all three variables ($ACT(I)$), ($INACT(I)$) and ($TOT(I)$) in its equation. It could be argued that the weights produced by the schemes may only result in a similar weighting strength carried over through the ranking process, thus yielding similar ranks.

4.6 Conclusion

In this chapter, the investigation of the performance of the SSA weighting schemes was conducted. From the observation during the predictive studies, the four relevance weights (R1, R2, R3 and R4) performed very well compared to other weighting schemes. However, between the relevance weights, R4 generally produced a higher retrieval rate in this predictive study, while SAS weighting schemes yielded the weakest result. Based on our findings, it was decided that the R4 scheme should be used in the predictive approach (using a smaller training set).

RNN activity representing the most homogeneous sets gave excellent results of active compounds retrievals using the SSA in this study; other homogeneous classes (HIVP, AT1, SUBP and THRM) also had good results in this analysis. On the other hand, heterogeneous classes (5HT1A, 5HT3 5HT, COX, FXA, PDE4 and D2) were found to deliver poor results. The experiments described in this chapter proved that the ten fragment weighting schemes performed better for the homogeneous activity classes when compared to the heterogeneous activities. A strong justification for this is that homogeneous classes tend to have more common identical fragments in the active compounds. This then translates to a high value of fragment weights and consequently a high probability that a compound is active. This also corresponds to the assumption of fragment weighting schemes, where all fragments in a given structure have a degree of influence to the likelihood of the compounds being active.

The results reported here are based on the predictive experiments using the ten presented SSA weighting schemes via large datasets obtained from MDDR, WOMBAT and ChEMBL databases. The predictive results obtained here will be compared to the genetic algorithm (GA) approach to weighting schemes via the predictive experiment. Chapter 5 describes an extension of the work presented in this chapter to explore the feasibility of a GA-based weighting analysis compared to best SSA weighting schemes found in this study, which is R4 in terms of the improvement in predictive performance. The objective is to validate whether the GA-based weighting determination yields similar or improved results when compared to the best SSA weighting scheme, i.e., the SSA R4 equation.

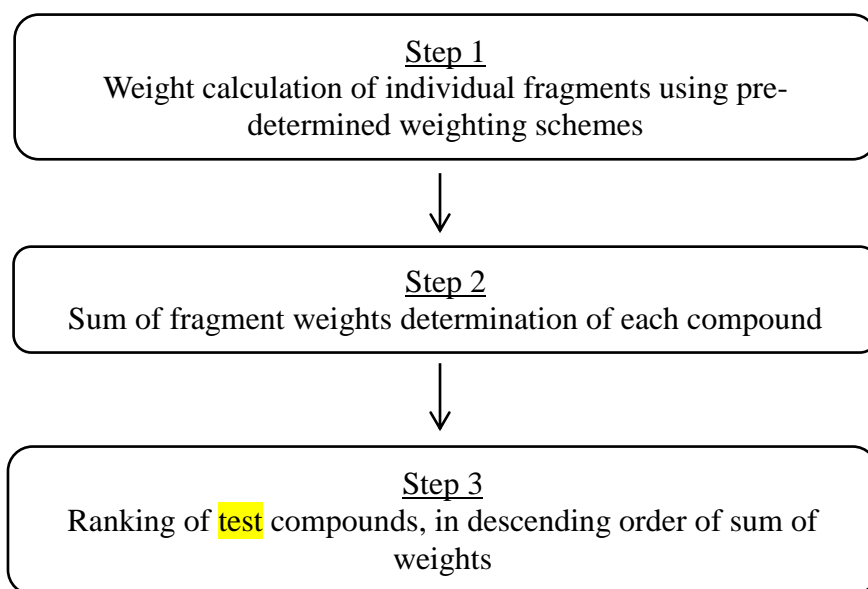


Figure 4.1: Methodology of experimental procedures conducted in predictive analysis

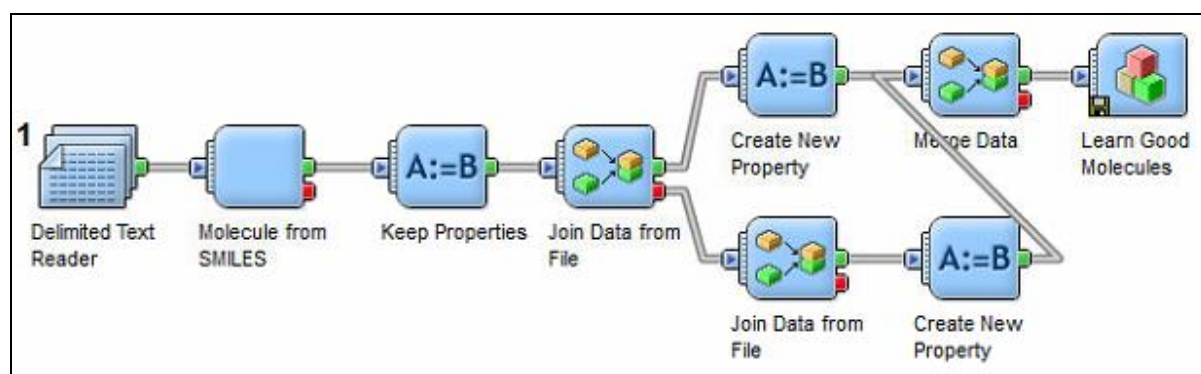


Figure 4.2: Pipeline Pilot workflow to generate NBC models using training sets

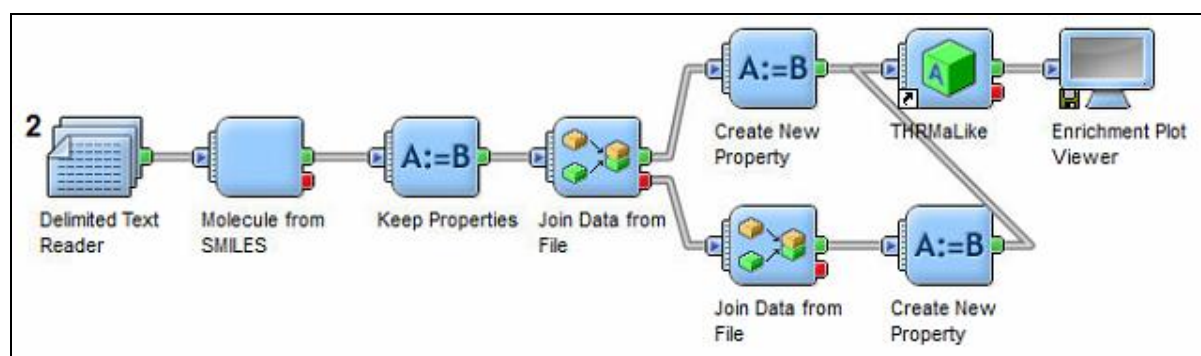


Figure 4.3: Pipeline Pilot workflow for new candidate screening using generated NBC models

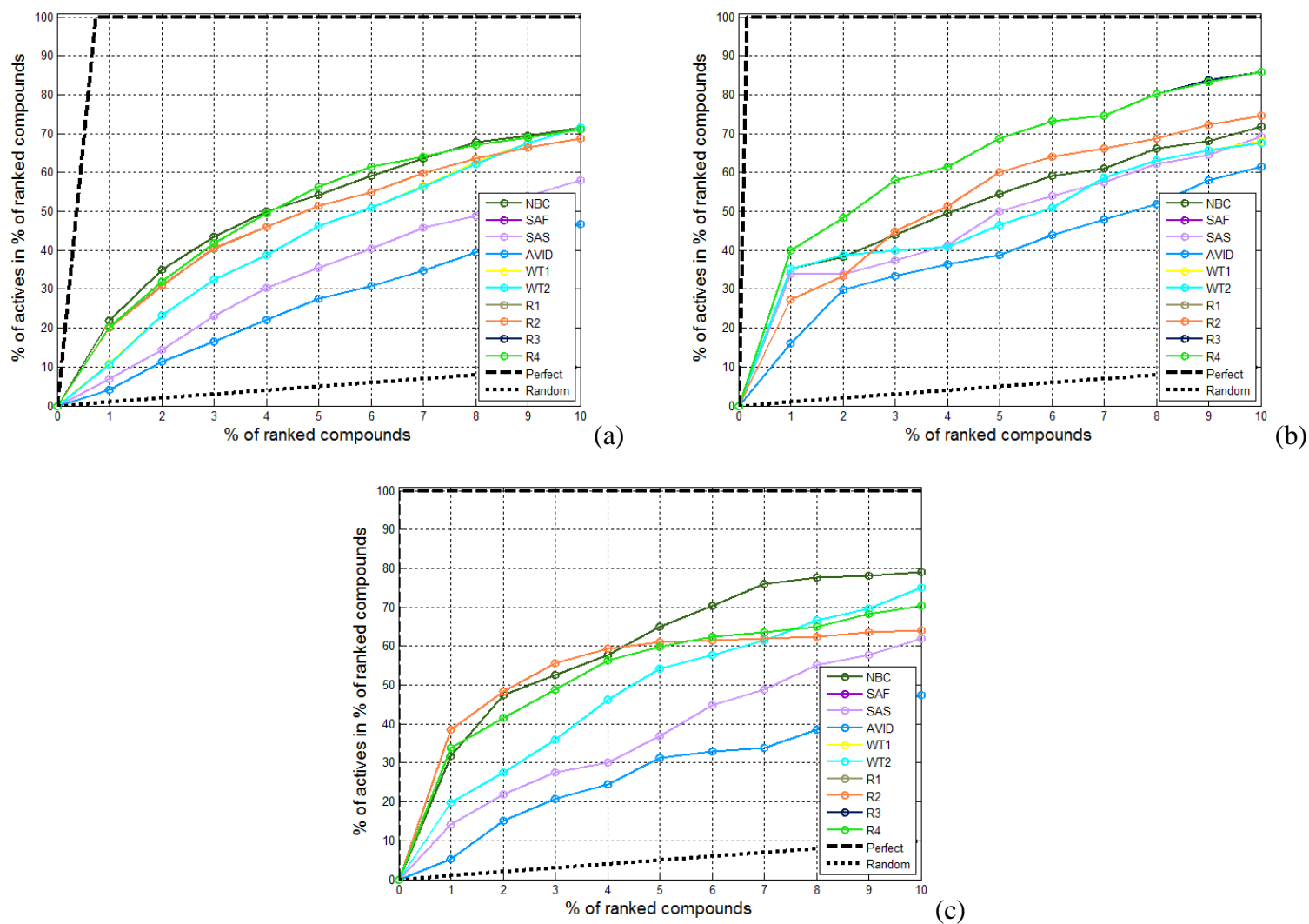


Figure 4.4: Cumulative recall plots of the various SSA weighting schemes for the 5HT3 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset

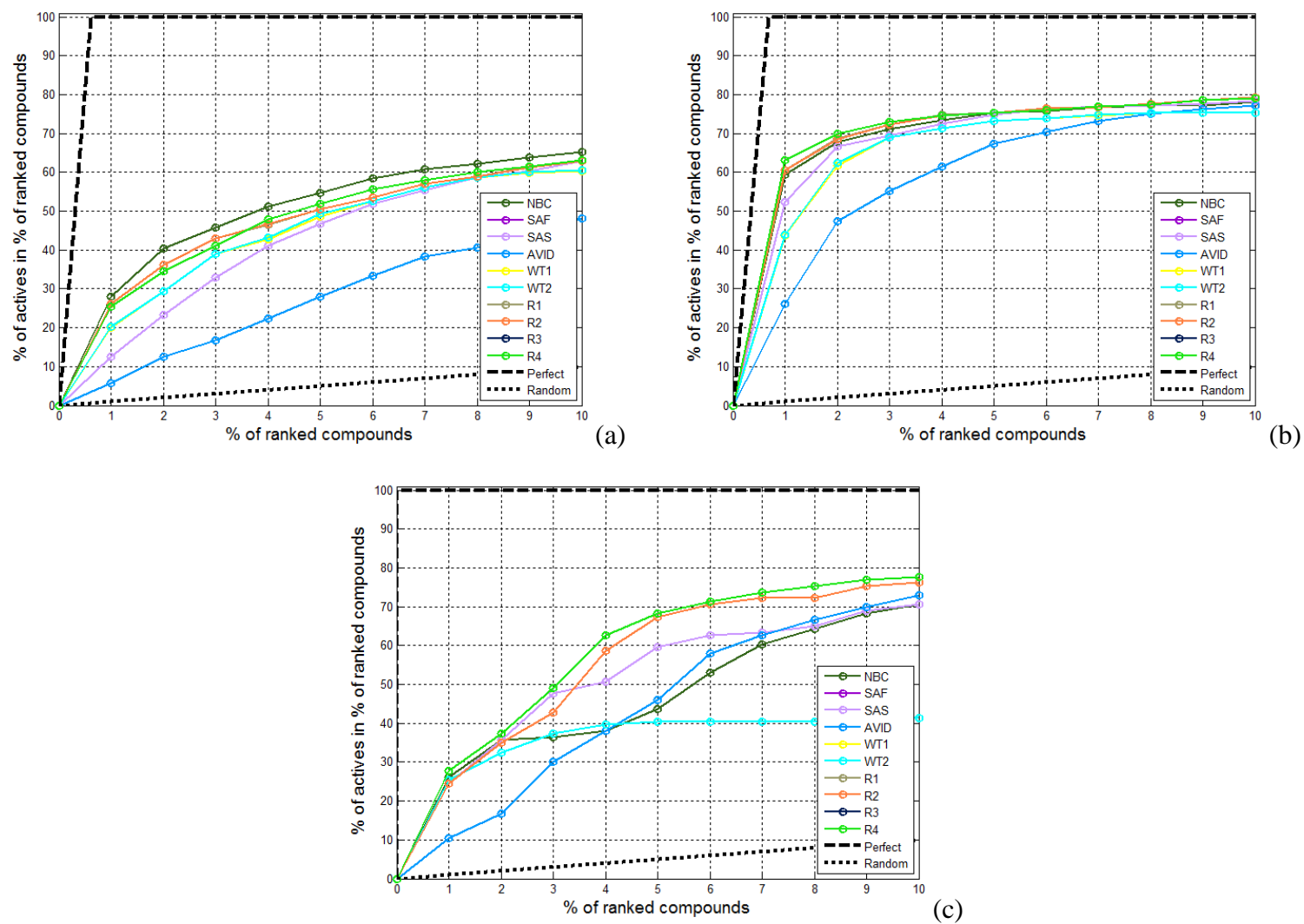


Figure 4.5: Cumulative recall plots of the various SSA weighting schemes for the COX activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset

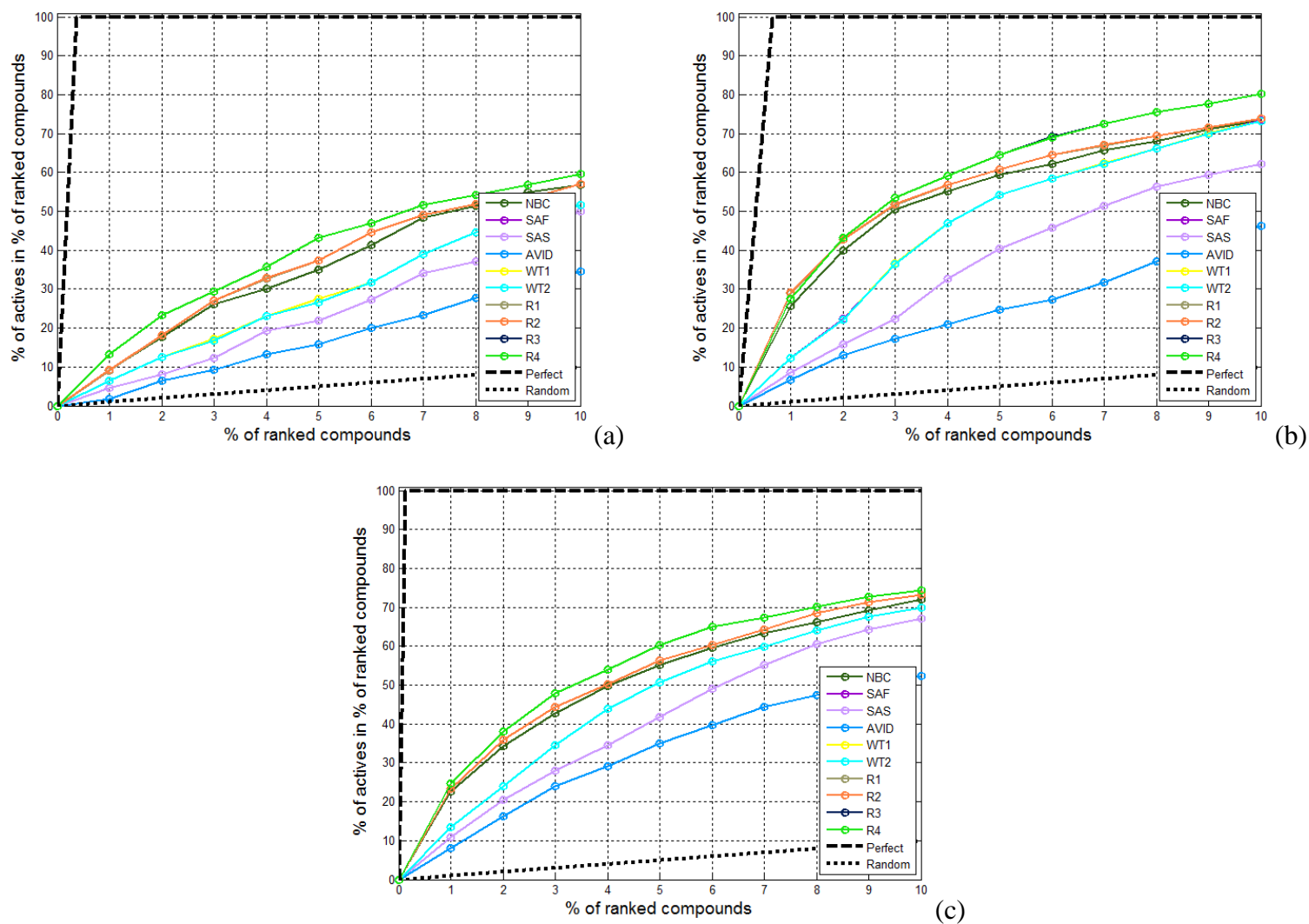


Figure 4.6: Cumulative recall plots of the various SSA weighting schemes for the D2 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset

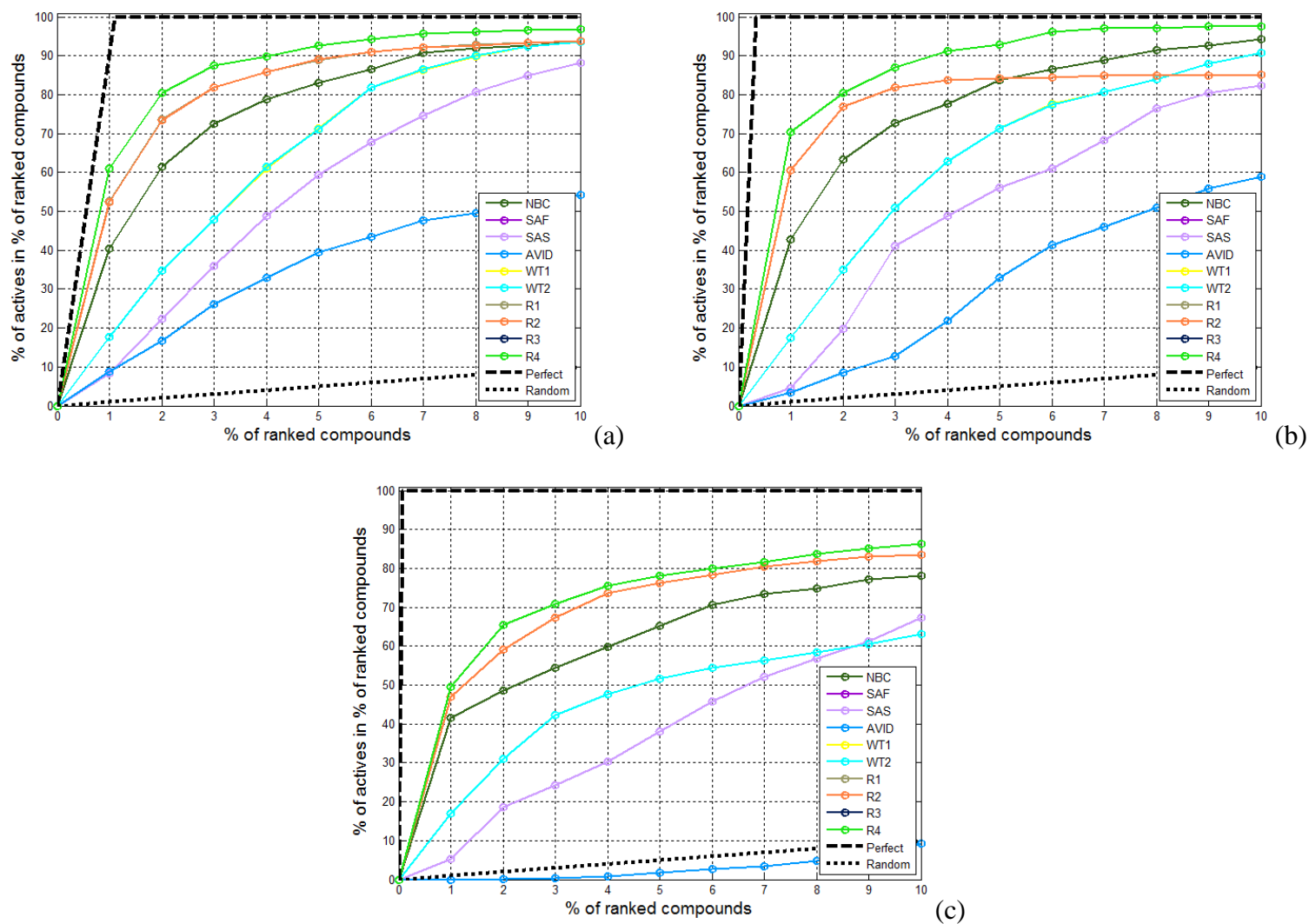
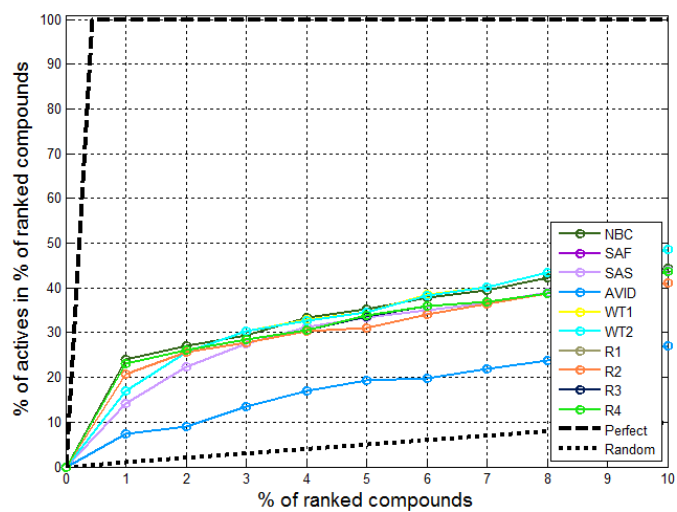
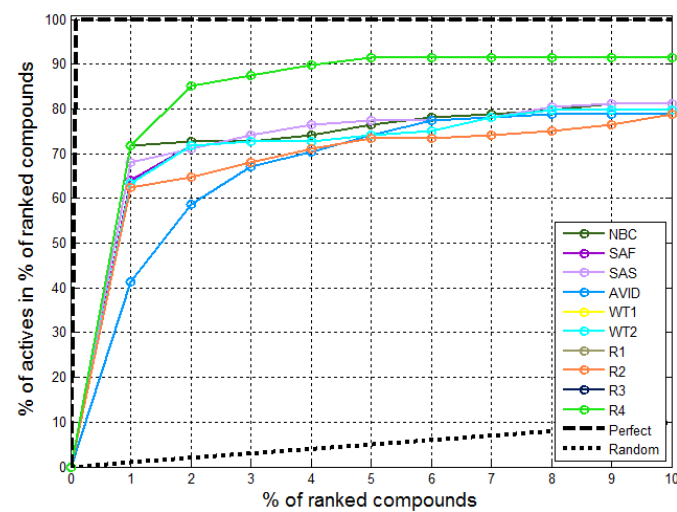


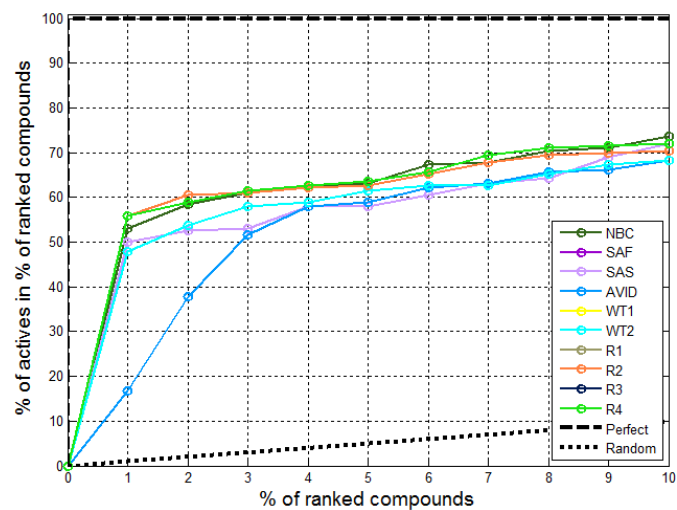
Figure 4.7: Cumulative recall plots of the various SSA weighting schemes for the RNN activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset



(a)



(b)



(c)

Figure 4.8: Cumulative recall plots of the various SSA weighting schemes for the PKC activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset

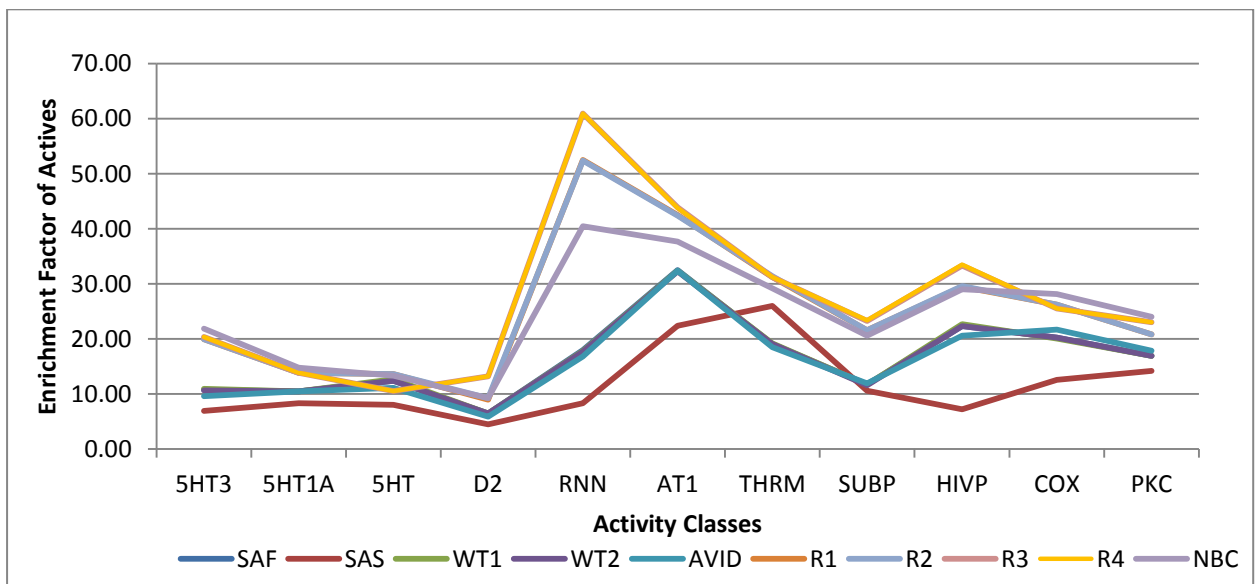


Figure 4.9: Comparison of the eleven MDDR activity classes based on the enrichment factor of actives in the top 1% of the rankings

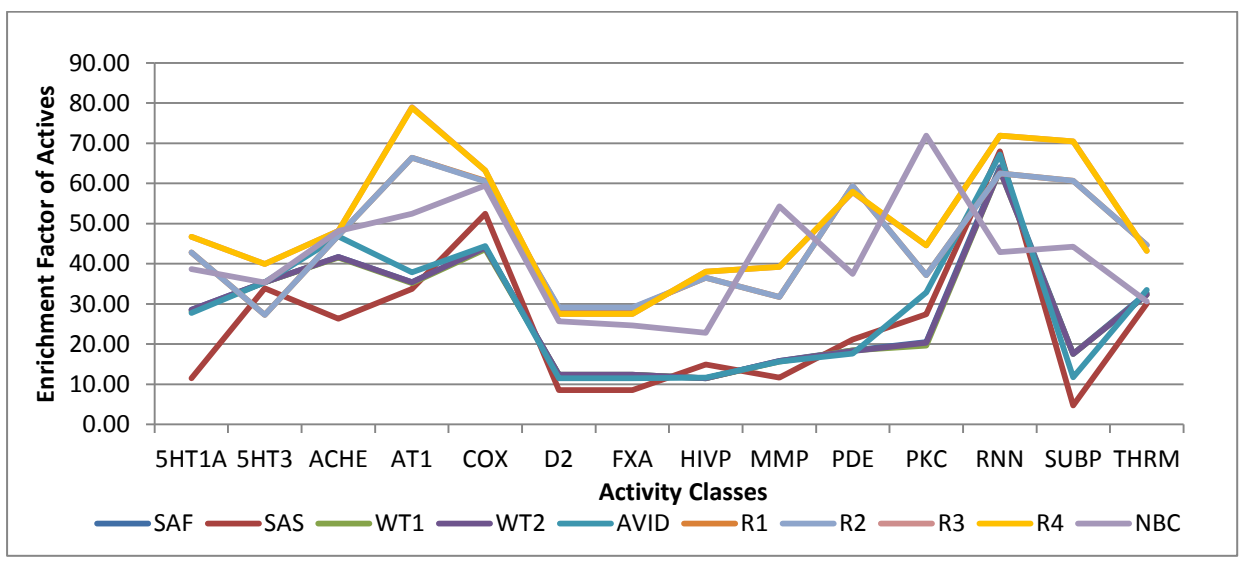


Figure 4.10: Comparison of the fourteen WOMBAT activity classes based on the enrichment factor of actives in the top 1% of the rankings

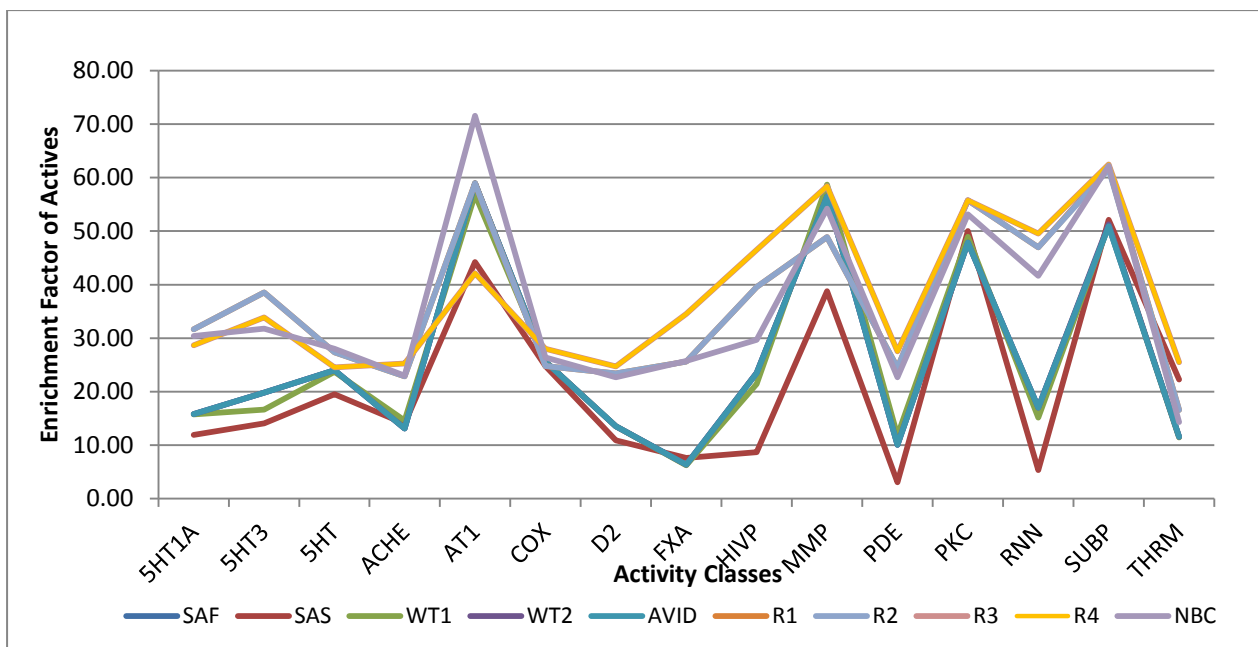


Figure 4.11: Comparison of the fifteen ChEMBL activity classes based on the enrichment factor of actives in the top 1% of the ranking

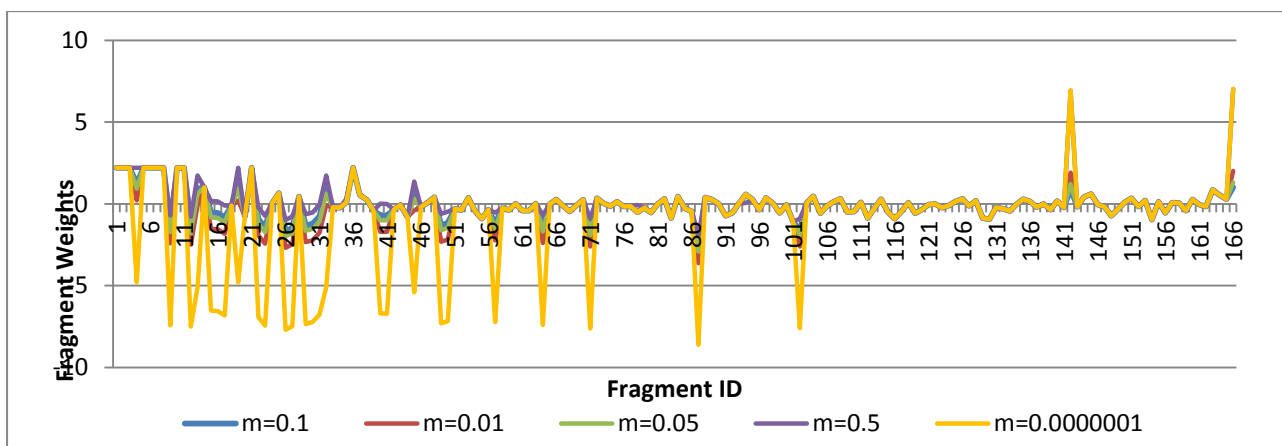


Figure 4.12: Comparison of fragment weights of 166 fragments, at m equals 0.0000001, 0.01, 0.05, 0.1 and 0.5. The SSA R4 weights are computed using the training sets of predictive analysis of COX activity class in the MDDR database

Table 4.1: Enrichment factor of actives retrieved in the top 1% of the ranked compounds of
(a) The eleven activity class in MDDR (b) The fourteen activity class in WOMBAT dataset,
and (c) The fifteen activity classes in ChEMBL dataset

Activity Class	Actives	Actives Retrieved									
		SAF	SAS	WT1	WT2	AVID	R1	R2	R3	R4	NBC
(a)											
5HT3	677	10.78	6.94	10.93	10.64	9.60	19.94	19.94	20.38	20.38	21.86
5HT1A	744	10.48	8.33	10.48	10.48	10.48	13.84	13.84	13.84	13.84	14.78
5HT	323	12.38	8.05	12.69	12.38	11.15	13.62	13.62	10.53	10.53	13.31
D2	356	6.46	4.49	6.46	6.46	5.90	8.99	9.27	13.20	13.20	9.27
RNN	1017	17.99	8.36	17.80	17.70	16.81	52.51	52.41	60.96	60.96	40.51
AT1	849	32.51	22.38	32.39	32.27	32.39	42.52	42.40	43.93	43.82	37.69
THRM	723	19.09	26.00	19.23	19.09	18.53	31.26	31.40	31.26	31.26	29.18
SUBP	1121	11.78	10.62	11.69	11.60	11.95	21.59	21.59	23.28	23.37	20.61
HIVP	675	22.37	7.26	22.67	22.37	20.59	29.48	29.63	33.33	33.48	29.04
COX	572	20.28	12.59	20.10	20.28	21.68	26.22	26.22	25.52	25.52	28.15
PKC	408	16.91	14.22	16.91	16.91	17.89	20.83	20.83	23.04	23.04	24.02
(b)											
5HT1A	533	28.52	11.44	28.14	28.52	27.77	42.78	42.78	46.72	46.72	38.65
5HT3	198	35.35	33.84	35.35	35.35	35.35	27.27	27.27	39.90	39.90	35.35
ACHE	453	41.72	26.27	41.50	41.72	46.80	47.02	47.02	48.12	48.12	48.12
AT1	652	35.43	33.74	35.12	35.43	37.88	66.41	66.41	78.99	78.83	52.45
COX	869	43.96	52.47	43.61	43.96	44.42	60.64	60.53	63.18	63.18	59.49
D2	819	12.33	8.55	12.33	12.33	11.48	28.94	29.06	27.47	27.47	25.64
FXA	758	11.48	14.91	11.48	11.48	11.61	36.54	36.54	37.99	37.99	24.67
HIVP	1015	15.76	11.63	15.67	15.76	15.67	31.72	31.72	39.21	39.21	22.76
MMP	625	18.40	21.12	18.40	18.24	17.60	59.36	59.36	57.76	57.92	54.24
PDE	536	20.52	27.43	19.59	20.34	32.84	37.13	37.13	44.59	44.59	37.50
PKC	128	64.06	67.97	63.28	63.28	67.19	62.50	62.50	71.88	71.88	71.88
RNN	427	17.56	4.68	17.56	17.56	11.71	60.66	60.66	70.49	70.49	42.86
SUBP	502	32.67	30.08	32.27	32.27	33.47	44.23	44.23	43.62	43.62	44.22
THRM	379	20.32	19.26	20.32	20.32	21.11	29.82	29.82	52.77	52.77	30.61

Activity Class	Actives	Actives Retrieved									
		SAF	SAS	WT1	WT2	AVID	R1	R2	R3	R4	NBC
(c)											
5HT1A	1335	15.81	11.91	15.73	15.81	15.81	31.69	31.69	28.69	28.69	30.41
5HT3	192	19.79	14.06	16.67	19.79	19.79	38.54	38.54	33.85	33.85	31.77
5HT	2202	24.01	19.47	23.69	23.97	24.01	27.33	27.33	24.51	24.51	28.02
ACHE	665	13.06	13.96	14.56	13.06	13.06	22.82	22.82	25.23	25.23	23.01
AT1	95	58.33	43.75	56.25	58.33	58.33	58.33	58.33	41.67	41.67	71.58
COX	125	25.40	24.60	25.40	25.40	25.40	24.60	24.60	27.78	27.78	26.40
D2	1672	13.51	10.88	13.51	13.51	13.51	23.37	23.37	24.69	24.69	22.67
FXA	1352	6.36	7.62	6.21	6.36	6.36	25.59	25.59	34.54	34.54	25.74
HIVP	1941	23.43	8.65	21.42	23.48	23.43	39.55	39.55	46.50	46.50	29.68
MMP	356	56.46	38.76	58.71	56.46	56.46	48.88	48.88	58.43	58.43	54.21
PDE	229	10.04	3.06	11.79	10.04	10.04	24.45	24.45	27.51	27.51	22.71
PKC	190	47.89	50.00	48.95	47.89	47.89	55.79	55.79	55.79	55.79	53.16
RNN	884	16.97	5.32	15.16	16.97	16.97	46.55	46.55	49.95	49.95	41.63
SUBP	762	50.85	52.03	51.11	51.25	50.85	61.73	61.47	62.39	62.39	62.20
THRM	754	11.52	22.25	11.39	11.52	11.52	16.56	16.56	25.43	25.43	14.32

Table 4.2: Kendall's W analysis for the top 1% actives retrieved of the ranking for (a) Eleven activity classes in MDDR, (b) Fourteen activity classes in WOMBAT, and (c) Fifteen activity classes of the ChEMBL database

(a)

SSA Weighting Schemes	Activity Class											Mean Rank	Rank Position
	5HT3	5HT1A	5HT	D2	RNN	AT1	THRM	SUBP	HIVP	COX	PKC		
R4	7.50	6.50	1.50	8.50	8.50	8.00	7.00	9.00	9.00	5.50	7.50	7.14	1
R3	7.50	6.50	1.50	8.50	8.50	9.00	7.00	8.00	8.00	5.50	7.50	7.05	2
R2	5.50	6.50	8.50	6.50	6.00	6.00	9.00	6.50	7.00	7.50	5.50	6.77	3
NBC	9.00	9.00	7.00	6.50	5.00	5.00	5.00	5.00	5.00	9.00	9.00	6.77	4
R1	5.50	6.50	8.50	5.00	7.00	7.00	7.00	6.50	6.00	7.50	5.50	6.55	5
WT1	4.00	2.50	6.00	3.00	3.00	2.50	3.00	2.00	4.00	1.00	2.00	3.00	6
SAF	3.00	2.50	4.50	3.00	4.00	4.00	1.50	3.00	2.50	2.50	2.00	2.95	7
WT2	2.00	2.50	4.50	3.00	2.00	1.00	1.50	1.00	2.50	2.50	2.00	2.23	8
AVID	1.00	2.50	3.00	1.00	1.00	2.50	0.00	4.00	1.00	4.00	4.00	2.18	9
SAS	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.36	10

(b)

SSA Weighting Schemes	Activity Class														Mean Rank	Rank Position
	5HT1A	5HT3	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
R3	8.50	8.50	8.00	9.00	8.50	6.50	8.50	8.50	6.00	8.50	8.00	8.50	5.50	8.50	7.93	1
R4	8.50	8.50	8.00	8.00	8.50	6.50	8.50	8.50	7.00	8.50	8.00	8.50	5.50	8.50	7.93	2
R1	6.50	0.50	5.50	6.50	7.00	8.00	6.50	6.50	8.50	5.50	0.50	6.50	8.50	5.50	5.86	3
R2	6.50	0.50	5.50	6.50	6.00	9.00	6.50	6.50	8.50	5.50	0.50	6.50	8.50	5.50	5.86	4
NBC	5.00	5.00	8.00	5.00	5.00	5.00	5.00	5.00	5.00	7.00	8.00	5.00	7.00	7.00	5.86	5
AVID	1.00	5.00	4.00	4.00	3.00	1.00	3.00	1.50	0.00	4.00	5.00	1.00	4.00	4.00	2.89	6
SAF	3.50	5.00	2.50	2.50	1.50	3.00	1.00	3.50	2.50	2.00	4.00	3.00	3.00	2.00	2.79	7
WT2	3.50	5.00	2.50	2.50	1.50	3.00	1.00	3.50	1.00	1.00	2.50	3.00	1.50	2.00	2.39	8
WT1	2.00	5.00	1.00	1.00	0.00	3.00	1.00	1.50	2.50	0.00	2.50	3.00	1.50	2.00	1.86	9
SAS	0.00	2.00	0.00	0.00	4.00	0.00	4.00	0.00	4.00	3.00	6.00	0.00	0.00	0.00	1.64	10

(c)

SSA Weighting Schemes	Activity Class															Mean Rank	Rank Position
	5HT1A	5HT3	5HT	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
R4	5.50	6.50	5.50	8.50	0.50	8.50	8.50	8.50	8.50	7.50	8.50	7.50	8.50	8.50	8.50	7.30	1
R3	5.50	6.50	5.50	8.50	0.50	8.50	8.50	8.50	8.50	7.50	8.50	7.50	8.50	8.50	8.50	7.30	2
NBC	7.00	5.00	9.00	7.00	9.00	7.00	5.00	7.00	5.00	3.00	5.00	5.00	5.00	7.00	4.00	6.00	3
R1	8.50	8.50	7.50	5.50	6.00	1.00	6.50	5.50	6.50	1.50	6.50	7.50	6.50	6.00	5.50	5.93	4
R2	8.50	8.50	7.50	5.50	6.00	1.00	6.50	5.50	6.50	1.50	6.50	7.50	6.50	5.00	5.50	5.87	5
WT2	3.00	3.00	2.00	1.00	6.00	4.50	2.50	2.00	4.00	5.00	2.00	1.00	3.00	3.00	2.00	2.93	6
SAF	3.00	3.00	3.50	1.00	6.00	4.50	2.50	2.00	2.50	5.00	2.00	1.00	3.00	0.50	2.00	2.77	7
AVID	3.00	3.00	3.50	1.00	6.00	4.50	2.50	2.00	2.50	5.00	2.00	1.00	3.00	0.50	2.00	2.77	8
WT1	1.00	1.00	1.00	4.00	3.00	4.50	2.50	0.00	1.00	9.00	4.00	3.00	1.00	2.00	0.00	2.47	9
SAS	0.00	0.00	0.00	3.00	2.00	1.00	0.00	4.00	0.00	0.00	0.00	4.00	0.00	4.00	7.00	1.67	10

Table 4.3: Kendall's W analysis for the top 1% based on the average of enrichment factor actives in the top 1% from the MDDR, WOMBAT and ChEMBL databases

SSA Weighting Schemes	Databases			Mean Rank	Rank Position
	MDDR	WOMBAT	ChEMBL		
R4	7.14	7.93	7.30	7.46	1
R3	7.05	7.93	7.30	7.43	2
NBC	6.77	5.86	6.00	6.21	3
R2	6.55	5.86	5.87	6.17	4
R1	6.77	5.86	5.93	6.11	5
SAF	2.95	2.79	2.77	2.84	6
AVID	2.18	2.89	2.77	2.61	7
WT2	2.23	2.39	2.93	2.52	8
WT1	3.00	1.86	2.47	2.44	9
SAS	0.36	1.64	1.67	1.22	10

Table 4.4: Fragment weights computed using R4 function for fourteen fragments (1, 2, 5, 19, 13, 17, 74, 67, 38, 53, 166, 164, 127 and 154) at m equals to 0.0000001, 0.01, 0.05, 0.1 and 0.5. The weights were derived using training sets in the predictive analysis

Fragment ID	ACTS(I)	TOTS(I)	(%)	(%)	SAF weight	INACTS(I)	Q	R	S	R4 weight = (Q/S)				
			ACTS(I)/N	TOTS(I)/N	ACTS(I)/TOTS(I)		ACTS(I)/R	NACT-ACTS(I)	INACTS(I)/ (NINACT - INACTS(I))	when m is equals to	$m = 0.0000001$	$m = 0.01$	$m = 0.05$	$m = 0.1$
1	0	0	0.00000000001	0.00000000001	1.000000000	0	0.000000002	63.0000	0.00000000001	2.2088	2.2089	2.2092	2.2095	2.2123
2	0	0	0.00000000001	0.00000000001	1.000000000	0	0.000000002	63.0000	0.00000000001	2.2088	2.2089	2.2092	2.2095	2.2123
5	0	0	0.00000000001	0.00000000001	1.000000000	0	0.000000002	63.0000	0.00000000001	2.2088	2.2089	2.2092	2.2095	2.2123
19	0	1	0.00000000001	0.0001	0.000000100	1	0.000000002	63.0000	0.0001	-4.7912	0.2089	0.9081	1.2095	1.9112
13	0	2	0.00000000001	0.0002	0.000000050	2	0.000000002	63.0000	0.0002	-5.0923	-0.0922	0.6070	0.9084	1.6101
17	0	104	0.00000000001	0.0101	0.000000001	104	0.000000002	63.0000	0.0103	-6.8127	-1.8126	-1.1133	-0.8120	-0.1102
74	6	1288	0.0006	0.1256	0.0047	1282	0.1053	57.0000	0.1439	-0.1358	-0.1358	-0.1358	-0.1358	-0.1358
67	12	2383	0.0012	0.2324	0.0050	2371	0.2353	51.0000	0.3032	-0.1102	-0.1102	-0.1102	-0.1102	-0.1102
38	9	870	0.0009	0.0849	0.0103	861	0.1667	54.0000	0.0923	0.2567	0.2567	0.2567	0.2567	0.2567
53	14	1142	0.0014	0.1114	0.0123	1128	0.2857	49.0000	0.1245	0.3608	0.3608	0.3608	0.3608	0.3608
166	63	10088	0.0061	0.9839	0.0062	10025	630000000.0000	0.0000	60.7576	7.0157	2.0157	1.3168	1.0157	0.3168
164	62	9714	0.0060	0.9474	0.0064	9652	62.0000	1.0000	17.9405	0.5386	0.5386	0.5386	0.5386	0.5386
127	25	4767	0.0024	0.4649	0.0052	4742	0.6579	38.0000	0.8704	-0.1216	-0.1216	-0.1216	-0.1216	-0.1216
154	18	8020	0.0018	0.7822	0.0022	8002	0.4000	45.0000	3.6572	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611

Table 4.5: Statistics for 166 fragment weights for m equals 0.0000001, 0.01, 0.05, 0.1 and 0.5.

The SSA R4 fragment weights are derived using the training set of predictive analysis of COX activity classes in the MDDR database

	Prior Value					
Data Statistics	0.0000001	0.01	0.05	0.1	0.5	SAF
Min	-8.62	-3.62	-2.92	-2.62	-1.59	0.00
Max	7.02	2.21	2.21	2.21	2.47	1.00
Mean	-0.92	-0.19	-0.09	-0.05	0.07	0.07
Median	-0.15	-0.12	-0.11	-0.11	-0.07	0.01
Standard Deviation	2.77	1.07	0.90	0.85	0.91	0.25
Variance	7.66	1.13	0.81	0.72	0.83	0.06

Chapter 5

Genetic Algorithm Approach to Substructural Analysis

5.1 Introduction

This chapter describes one of the two central themes of this thesis, which involves the use of one specific evolutionary algorithm for the purpose of SSA. This is the Genetic Algorithm (GA), focusing on 2D fingerprints analysis. The chapter reviews earlier implementations of the GA related to the field of chemoinformatics. It discusses the GA in detail from definitions to descriptions of components that form the technique, and subsequently preparations required in order to apply the GA-based SSA. To assess the performance of GA for SSA, two main experiments were carried out. The first was to identify the best parameterisation set of the GA to be used, as the GA has a number of options and alternatives that may define the method's success. The second experiment deals with analysing the true performance of the GA-based SSA on selected molecule databases and their activity classes. Comparisons are made against the SSA traditional weighting schemes, as reviewed in Chapter 4, Section 4.3. A conclusion is drawn as to whether the GA-based SSA outperforms the earlier schemes.

The GA is a machine learning method, mainly based on the principle of “survival of the fittest”, following the theory of biological evolution (Goldberg & Holland, 1988). GA was first introduced by John Holland and has since been widely adopted in various disciplines (Mitchell, 1998; Goldberg & Holland, 1988). The GA is basically a search heuristic method used to generate useful solutions for optimisation and search problems from a multiple set of possible solutions mimicking biological chromosomes. The standard GA implementation does not require complex mathematical implementation; instead it relies on simple genetic operators, specifically the selection, crossover and mutation operations.

5.2 Fundamental components of GA for SSA

Figure 5.1 describes the basic workflow of the GA. The GA begins with the initialisation of a population made up of chromosomes, usually represented in bit strings, real number arrays or character strings. Such chromosomes are related to a set of parameters or variables decoded from the solution required.

The chromosomes are stimulated to repeatedly go through cycles of genetic operations which alter the chromosomes' elements through a series of crossover and mutation operations. A fitness function represents the chromosomes' suitability as a solution, in which either a perfect, or non-improving solution equates to a stopping criterion. One can think of the best possible stopping criterion to be ideally and more practically, the solution that satisfies the fitness objective. Alternatively, reaching the maximum number of GA iterations can also be considered as another stopping criterion to signal an end to the GA program. The components of the GA are described in more detail below.

5.2.1 Encoding of chromosomes

Encoding of chromosome is an important part of the GA. Encoding is the way to represent the solution. The choice of chromosome representation in genetic algorithms depends on the variables of the optimisation problem being solved. For example, in the case of a minimisation problem of a linear equation, the chromosomes are represented by a series of possible integers representing equation variables. This is conducted in order to quantify for possible numbers that fit the equation. Similarly, for an optimisation problem within a circuitry design, the chromosomes are logically formed by binary flags of different circuit paths or gates. The chromosome, once decoded according to the specifics of the coding strategy, can then be used to calculate the suitability of the candidate solution to the problem. This can be done by performing the fitness function that has been specifically defined for the problem domain.

Several types of chromosome encoding are available. The most common is known as binary encoding formed with the binary numbers 0 and 1. This is the easiest form of encoding that works with genetic operations. Other number-based encoding includes octal or hexadecimal encoding which refers to a different number system than the binary one. Apart from number-based encoding, strings or permutations can also be used for chromosome encoding, which are usually specific to recognition-based problems. Tree encoding is another type of encoding, but this is almost exclusively not for GA (instead being meant for GP). The choice of a suitable encoding scheme depends on the problem specification and the type of fitness function required in relation to the problem.

In the case of 2D fingerprints, molecules are represented by a dictionary of fragments denoting the presence or absence of individual fragments in a given molecule. For the SSA,

such fragment information was used to calculate a series of weight values, based on the various weighting schemes available, and these weights are in the form of integers. On this basis, it was necessary to adopt the chromosomes in the GA-based SSA in order to have a similar weight value combination. The integer encodings of chromosomes seem to be the logical choice to represent the fragment weight of a given fragment.

5.2.2 Fitness criterion

The fitness criterion is an important component of the GA. Otherwise known as the objective function. It is usually represented in the form of a function or equation. The fitness function serves two main purposes. Its most important function is to evaluate an individual's suitability as the preferred solution to be produced by the GA. Considering that most GA programs begin with a population set of randomised solutions, the fitness function is calculated for each individual in the population so as to guide the selection of potential solutions. The selection continues to evolve in subsequent genetic evolutions. A good fitness function is critically important in achieving a good success rate with the GA (Koza, 1992). To design a suitable fitness function, an accurate understanding of the problem definition is required. The fitness function can relate either to a maximisation or minimisation problem. Complex optimisation problems may require a multi-objective fitness function rather than a single-objective based one. In principle, such a design of an objective function requires a form of mathematical representation which can be described by the GA program. In the case of fingerprint-based SSA approach of the GA, the definition of the fitness function used is described in Section 5.4.3.1.

5.2.3 Chromosome selection methods for genetic operations

Genetic operations are performed on chromosomes via mating and mutation methods. These operations form the core of the GA program. Prior to genetic operations, one or two parent chromosomes are first selected to undergo such procedures. Such selections are critical in determining how the entire chromosome population will progress. In its most basic form, selection for a parent chromosome should be random in nature. The selection, however, may also be guided by a certain element of influence, such as fitness suitability or diversity of chromosomes' characteristics. The two fundamental selection processes related to restrictive random chromosomes selection as described below.

Roulette wheel selection

In this method, chromosomes are first ordered in a formation that mimics an actual roulette wheel device. Each chromosome takes up a share of the wheel based on their fitness score. Chromosomes with higher fitness are given a larger share of the wheel, and thus they have a higher probability of being selected. Next, a random number from the range of 0 to the total sum of accumulated fitness is then generated. This number then traverses the wheel and stops at the wheel portion that encompasses the number, representing the selected parent chromosome. A further example is shown in Figure 5.2, where a set of chromosomes is listed, along with their fitness values and their percentage of fitness, calculated from the total fitness (Figure 5.2a). When sorted in descending order, a random number is generated to select the wheel quotient which represents the selected chromosome (Figure 5.2b). Following the example from Figure 5.2, the chromosome C_5 is shown to have the largest fitness value compared to the other chromosomes: thus holding a larger share of the wheel. It should also therefore have a higher probability of being selected than the other chromosomes.

Tournament selection

Tournament selection mimics a tournament style approach in which a series of randomly chosen chromosome pairs are pitted against one another. The winner then progresses to the next round until only one chromosome remains. Following the example in Figure 5.3, four chromosomes were randomly selected and grouped in two pairs (Figure 5.3a). Each pair then competes with one another using their fitness value and the winner progresses to the next round (Figure 5.3b). The process is repeated again until one chromosome remains, as shown in Figure 5.3(c). Zhong et al. (2005) reported the preference and advantage of the tournament selection. It is beneficial as it has the ability to converge chromosomes faster into desired solutions, since the selection allows for greater consideration of mid to lower ranked chromosomes. These can survive the selection process more efficiently compared to the roulette wheel method.

5.2.4 Evolutionary operators

There are two main genetic operations within the GA, specifically the crossover and mutation operations. The crossover operation requires two parent chromosomes to be selected and mated together to produce an offspring chromosome. The mutation operation, however, causes an individual element of one chromosome to be randomly mutated to another value. The schematic of the two operations above are described in Figure 5.4. For the crossover

operation, it is possible to consider a population of 4 chromosomes encoded in binary (Figure 5.4a). Two chromosomes are selected at random and a portion of each of the selected chromosomes is taken and recombined to produce an offspring chromosome (Figure 5.4b). Various crossover methods have been described and proposed (Holland, 1975; Davis, 1985; Vekaria & Clack, 1998; Kaya, 2011; Kaya, Uyar & Tek, 2011; Kuczkowski, Kolendo, Jaworski & Smierzchalski, 2012; Mendes, 2013). Here, three common crossover methods associated with evolutionary algorithms are described. The first is known as the *One-point Crossover*. In this method, two selected parents (Parents 1 and 2) are combined using a randomly generated combination point. The points lie anywhere between the second elements to the second last element of a chromosome index. This example is shown in Figure 5.5(a). The ordering of the combination (either parent 1 first, then 2, or vice versa) is also determined randomly. The second crossover method is the *Two-point Crossover*, where the two selected parents are recombined with two random recombination points, as opposed to one (Figure 5.5b). Finally, the third common crossover method is the *Uniform Crossover*, which functions by recombining a similar ratio of genes from both parents (Figure 5.5c).

The resultant offspring then undergo a mutation process, which identifies a random position index and subsequently replaces it with a new randomised value. In the case of a binary string such as the above, the highlighted fragment bit of “0” is simply replaced with a “1”. The final reproduced and mated chromosome is then evaluated and inserted back into the population list. To mimic the natural process of biological mutation and evolution, the frequency of these operations can be controlled by assigning a rate of occurrence, ranging from 0 to 1. 0 refers to a complete lack of chance (or a non-occurrence) for an operation to be performed, while 1 signifies an absolute occurrence of such operation at every single evolutionary stage. A high rate of crossover and the added "randomness" factor brought about by mutation help to increase the chances of reaching global convergence. The process thus effectively determines a suitable solution in the long run.

5.2.5 Chromosome's principle of elitism

Elitism is mainly concerned with the preservation of fit chromosomes within each generation which is carried over to the next generation without any modification. These are usually in the order of the most effective solution to the least. These are preserved and brought forward into a subsequent series of genetic evolutions. It is primarily a method of preserving the desired solutions before they are unwittingly mutated into a lesser form, due to genetic

operations. Two main elitism models are available, known as the simple-state model (otherwise referred to as generational-state); and the steady-state model.

Simple-state preservation model

The simple-state model assumes that offspring from parent chromosomes are largely better than the parents themselves; as illustrated in Figure 5.6. The offspring chromosomes often replace a majority, and in extreme cases, all parents between each generation. For example, an elitism of two chromosomes ensures that only the best two chromosomes are retained in the next stage of chromosome evolution, while the rest are replaced by offspring. A major advantage of this preservation model is that it is possible to reduce the time it takes to converge outside the local maximum search space and into the global maximum space more rapidly. This is mainly due to the replacement of a large number of chromosomes between the evolutions. The drawback, however, is that close-to-ideal chromosomes might be replaced too early in the evolution series. This can effectively reduce the chance of arriving at desirable solutions at a later stage. In the extreme case of an absolute simple-state model (where no chromosomes are preserved between evolution), each generational evolution enforces the chromosome to search for its solutions in the global space. The reason is that no localised maxima are recorded (except for traces of good parent chromosomes). An ideal simple-state mode would usually retain at least 1 or 2 parent elitist chromosomes, which are then carried forward into the next GA evolution.

Steady-state preservation model

An alternative model, the steady-state model replaces only a few chromosomes during each GA evolution. Only the last few parents are usually omitted and replaced by more promising offspring chromosomes, as depicted in Figure 5.7. This may affect the GA's potential to either expand the search space, as it may require further iterations of evolution to arrive at a preferred solution, unless a solution is achieved earlier on. A major advantage of the steady-state model, however, is that it can retain the core of fittest chromosomes within each generation, and thus they are not accidentally discarded during intensive rounds of GA iterations. This is in sharp contrast to the simple-state model. This employs almost a complete replacement of parents with their offspring at each stage of the evolution.

5.3 Previous works in GA

A considerable amount of literature has been published on applications of GAs in more general chemical fields (Clark, 2000). GAs have been applied successfully in many fields including chemoinformatics and computational chemistry. These include the application of the GA in pattern recognition analysis of multivariate chemical data, protein-ligand docking, the identification of novel antibacterial peptides, conformational analysis, molecular graphs, pharmacophore-mapping, de novo design, hyperstructures, the generation of QSAR models and library design (Willett, 2000; Lavine, Davidson & Moores, 2002; Brown, McKay, Gilardoni & Gasteiger, 2004; Wang, Krudy, Xie, Wu & Holland, 2006; Fernandez, Caballero, Fernandez & Sarai, 2011; Fjell, Jenssen, Cheung, Hancock & Cherkasov, 2011). Here, the focus is on previous GA-related studies carried out in Sheffield to exemplify the range of problems put to the test with the GA.

Brown, Jones, Willett, and Glen (1994) were probably the first to work on utilising a GA for 2D chemical matching of query substructures and chemical database structures, as well as investigating the GA for the generation of hyperstructures. A chemical hyperstructure is a single structure representation of a library, which is generated by the sequential overlapping of each molecular graph in the library to the current hyperstructure. The overlapping is carried out so as to minimise the size (in terms of numbers of nodes and edges) of the resulting hyperstructure. In their research, the GA was found to be less effective than conventional search methods, but it showed good potential in terms of hyperstructures construction and generation.

Wild and Willett (1996) worked on evaluating the effectiveness of the GA in the alignment of molecular electrostatic potentials (MEP) in 3D chemical structure databases for similarity searching. This was followed by Holliday and Willett (1997), who investigated the GA to identify pharmacophores through a developed program called MPHIL (Mapping Pharmacophores In Ligands). In 1997, Jones, Willett, Glen, Leach and Taylor used the GA to develop an automatic ligand docking program, known as GOLD (Genetic Optimisation for Ligand Docking). The program has been used extensively in drug discovery for identifying binding space and ligand conformational space (Jones, Willett, Glen, Leach & Taylor, 1997). Gillet, Willett and Bradshaw (1998) later used a GA to calculate weights of activity profiles determined by SSA methods, based on high-level structural molecular features. Part of the

objective was to investigate the GA's capability and potential to enhance drug discovery and/or design utilisation of such structural features. The authors experimented with GA parameters such as operator weights, population size, chromosome length, mutation rate and number of iterations. This was conducted to affect the way evolution converged on two sets of SSA weighting schemes (specifically Cramer's SAF and Robertson and Sparck-Jones's R2). They found a good discrimination between active and inactive compounds when combining the SSA with a GA, enabling the weights to have more predictive capability and an influence when ranking the compounds. However, the approach still requires further enhancement and development in tandem with the structural features and bioactivity information available.

Bayley, Jones, Willett, and Williamson (1998) employed a GA to calculate the structures of proteins using NMR restraints. It was discovered that the algorithm effectively calculated structures in which the calculated distance was similar to the geometric distance. Several authors have also investigated GAs to solve the ligand docking problem for screening chemical databases (Jones & Willett, 1995; Jones, Willett, & Glen, 1995a; Jones, Willett, & Glen, 1995b; Jones, Willett, Glen, Leach, & Taylor, 1997; Jones, Willett, Glen, Leach & Taylor, 1999). Studies by Gillet, Willett, and Bradshaw (1999) and Gillet, Khatib, Willett, Fleming, and Green (2002) analysed the use of a multi-objective GA to derive an optimum scoring function for combinatorial library design in the program MoSELECT. In MoSELECT, a typical library design scenario would be to design a library that is not just structurally diverse, cheap to synthesise but that also has drug-like physicochemical properties. Further evidence of the GA performance was also found in another study, achieving successful results in comparing a surface for protein-docking via the application of a GA (Gardiner, Willett & Artymiuk, 2003; Gardiner, Willett & Artymiuk, 2001; Poirrette, Artymiuk, Rice & Willett, 1997).

Cottrell, Gillet, Taylor and Wilton (2004) worked on applying multi-objective optimisation technique of the GA on pharmacophores methods. The aim was to generate valid multiple pharmacophores hypothesis from a series of overlay hypotheses, allowing for flexibility and diversity in the established structure-activity relationships. An extension of this work was further carried out by Gillet (2004), in which the multi-objective GA was explored for the design of effective combinatorial libraries that represents different knowledge and

compromised objectives. The multi-objective optimisation concept is discussed in further details in Section 5.4.3.1.

The successful results of GAs in the studies discussed above led us to use GAs in developing a method to predict biological activity. The following section aims to identify whether a set of optimised weights can be generated and compared to the existing substructural analysis weighting schemes. The objective is to provide a way of evaluating the effectiveness of GAs, as suggested in past studies.

5.4 Experiment details

The main objective of the experiment is to utilise the stochastic approach of the GA when determining suitable fragment weights for compound ranking. Results obtained from the GA are evaluated against Robertson and Sparck-Jones's R4 weighting scheme for the SSA. This is known to be the most consistent and often the highest performing scheme, as reported in Chapter 4. Findings from the evaluation of the GA-based SSA may answer the following research question: whether an evolutionary algorithm such as the GA is able to improve the upper-bound to the performance of the SSA. As a first step, the requirements and setups are discussed in order to prepare for the experimentations.

5.4.1 Dataset

The datasets used for the GA experiments were similar to those reported in Chapter 3. This is to ensure that results obtained by the GA-based SSA can be benchmarked against traditional SSA weighting schemes as discussed in Chapter 3. They comprise of eleven, fourteen and fifteen activity classes from the MDDR, WOMBAT and ChEMBL databases, respectively (Table 3.1). The training sets of each activity class contained 10% active and 10% inactive molecules as the input dataset to be used in the GA. The remaining 90% of the data were subsequently classed as the test set. They were thus used to evaluate the predictive performance of the training set.

5.4.2 Hardware

The GA program developed was executed on a number of concurrent hardware devices with different architecture. It was necessary to use these efficiencies in run-time and reduced computational costs. Also used was the hardware as listed in Table 3.2. A run-time performance benchmark and analysis is presented in Section 5.7.6.

5.4.3 Algorithm implementation

Following Chapter 4, MATLAB was used for the GA investigation for its ease of programming and rich access to existing libraries, utilities and data structures. MATLAB is packaged with its own global optimisation toolbox, which includes implementations of the GA tools. It was decided, however, to code a GA program from scratch to ensure that all functions and algorithms could be monitored properly. In this way, it was possible to introduce appropriate code optimisations tailored specifically towards fingerprint-based GA implementation. Appendix A lists the full pseudo code of the GA-based SSA program written in MATLAB.

Algorithm 5.1: Main GA program

```
1: Initialise a population of chromosomes
2: Evaluate the fitness of individuals
3: repeat
4:   Select best individuals to be used for genetic operations
5:   Generate offspring individuals using crossover and mutation
6:   Evaluate fitness of individuals
7:   Replace parent chromosomes in the population with offspring chromosomes
8: until stopping criterion satisfied
```

Algorithm 5.1 highlights the main GA program developed in MATLAB which conforms to the standard GA workflow. The program consists of several important steps. The first is the initialisation of a chromosome population, which is a representation of properties to determine the solution required from the GA. The main workflow of the GA program is shown in detail in Table 5.1. The chromosomes here are randomly defined by default, but special circumstances can be imposed on these chromosome assignments, such as discussed in Section 5.2.1. Each chromosome individual is assigned a fitness score to represent its suitability as a solution candidate. For the GA-based SSA, the fitness determination is explained in detail in Section 5.4.3.1 below. The bulk of the program is processed through multitudes of genetic evolution phases, consisting of genetic mating and reproduction operations (Section 5.4.3.3) through continuous genetic changes and fitness reassessment. The program ends when a suitable solution is found.

5.4.3.1 Suitable fitness function for SSA-based GA

The success of a GA is primarily based on the design of a suitable fitness function that describes the problem to be solved. In the case of biological activity prediction for 2D fingerprints, the chromosomes represent a series of weights corresponding to a particular fragment substructure. The chromosome for the GA is a vector containing N integers, where the i -th element is the fragment weight for the i -th bit in the fingerprint; and the fitness function for the GA is the number of active molecules that occur in the top-1% of a ranking of a training set of active and inactive molecules when the molecules are ranked using the set of N weights encoded in a chromosome. The GA, which uses single-point crossover and single-bit mutation, is run for a pre-set number of generations or until the weights have stabilised, thus providing an estimate of the best possible SSA weights that can be obtained using that training set. The resulting weights can then be applied to a separate test set. A score for a test set molecule is computed by summing (or otherwise combining) the weights of those bits that are set in its fingerprint, this sum representing the overall probability of the molecule being active given that it contains a particular pattern of bits. Ideally, the right weighting combination should maximise the number of active molecules situated in the top ranked portion of the given dataset. Hence, the number of active compounds found in the top, defined cut-off of the ranked dataset (usually in the top 1%) is considered to be the objective of the GA-based SSA.

An extension of this approach is the multi-objective optimisation technique, which takes into account multiple source of information or conditions to be considered as objectives to be optimised (Nicolaou & Brown, 2013). Researches in chemoinformatics applying multi-objective optimisation were extensively performed by Cottrell, Gillet, Taylor and Wilton (2004), and Gillet (2004), as previously described in Section 5.3. Such optimisation approach was also explored in detail by others, notably Li, Yang and Liu (2014, 2015a, 2015b), in which they addressed challenges of evolutionary algorithms in applying multi-objective optimisation problems. Their work focused on the popular Pareto efficiency, a concept to achieve optimal solution from multiple objectives without introducing a dominant solution for one objective, known to be a common problem.

For the purpose of simplicity and to prove the effectiveness of the GA-based SSA, however, a single-objective fitness definition is preferred for the experimentation. The corresponding fitness function should therefore be defined as the number of active compounds in a ranked

compound list. This is based on a given set of fragment weights. In mathematical terms, this can be represented as Equation 5.1 below:

$$F = (\sum_{i=1}^n a) \quad \text{(Equation 5.1)}$$

Where i equals the rank position;

And n equals the index at X percent in the molecules data;

And a is the molecule activity state, which is either 0 for inactive, and 1 for active

To calculate the fitness score of a given chromosome, the chromosome consisting of randomised fragment weights is applied to the molecule dataset. Each fragment's individual scores are then added together to represent the molecule total score. The molecules are subsequently ranked based on their scores in descending order. The molecule's activity state is highlighted. From Equation 5.1, the ranked molecules' activity states were observed to calculate the number of active molecules present in the first X percent of the ranked list, usually in the top 1 percent of the data. Another variation of the equation above can also be performed, for example, in the top 2 percent or 10 percent instead. Hence, the fitness function used in our GA is simply the number of active molecules ranked in higher threshold rank position. This is also emphasised in Algorithm 5.2 below.

Algorithm 5.2: Fitness function evaluation of individuals

```
1: for each molecule
2:   for each fragment
3:     Multiply fragment's presence with chromosomes' fragment weight
4:   end
5:   Sum the fragments' scores to get molecule score
6: end
7: Sort molecules based on the molecule score in descending order
8: Fitness is number of active molecules in top X percent of ranked molecules
```

A simpler example of the fitness function for the GA-based SSA is further demonstrated, as shown in Table 5.1. Consider a molecule dataset comprising of five molecules, defined by a dictionary of five fragments as in Table 5.1(a). In the example, the molecule M₁ contains the

second and fifth fragments F_2 and F_5 respectively. The molecules' activity states are defined in Table 5.1(b), where 0 denotes an inactive molecule and 1 is for an active molecule. The table states that molecules M_1 , M_3 and M_5 are active, while the rest are inactive. An initial population of chromosomes is generated with the initial weight values assigned by a random-number generator (Table 5.1c). Each chromosome is then used to compute the sum-of-weights for each molecule, as shown in Table 5.1(d). For example, M_1 contains F_2 , F_4 and F_5 , so its sum-of-weights using C_1 is the sum of W_2 , W_4 and W_5 , i.e. 3; using C_2 the sum is 16; and so on for C_{3-6} . Considering just C_1 , the sums-of-weights for molecule M_{1-6} are 3, 13, 7, 13 and 2 respectively; hence the fragment weights represented by chromosome C_1 results in the following ranking of the training set:

$$M_2 = M_4 > M_5 > M_3 > M_1$$

On the other hand, C_2 yields the ranking:

$$M_3 > M_1 > M_2 > M_5 > M_4$$

Taking the example dataset above and observing only the top three ranked molecules to represent the fitness criterion, the fitness value for chromosome C_1 is calculated as 1 out of 3 since the top two ranked molecules are inactive ones. Similarly, chromosome C_2 scores a fitness value of 2 out of 3, or 0.667. If the fitness scores of all the chromosome sets are calculated, chromosome C_3 will obtain the perfect fitness value based on the ranking below. In this case, the active molecules are ranked in the first three positions:

$$M_1 = M_3 > M_5 > M_2 > M_4$$

5.3.3.2 Weight polarity to overcome overfitting

Hawkins (2004) refers to overfitting as a problem in which a statistical model describes a random error or noise instead of the underlying relationship. It is a known problem in relation to machine learning methods and received special attention in the study of its causes and of methods to overcome this problem (Santos, Sabourin & Maupin, 2009; Hawkins, 2004; Domingos, 2012).

In its simplest form, an overfitting case occurs when a solution from one dataset, trained with a machine learning technique, fails to arrive at both successful and similar conclusions when

applied to another dataset. This may occur when performed on another ‘unseen’ dataset. It is a classic case of the generalisation failure, in which a training dataset does not represent the complexities present in the general untrained data. As the GA belongs to a non-deterministic class of algorithms, the optimal solution found for the GA may vary each time the algorithm is rerun with the same input data and GA parameters. For the GA-based SSA, a particular feature was introduced for the purpose of regulating the weight distribution in the fragment dictionary, and ultimately to overcome overfitting. This was achieved by manipulating the frequency of fragment occurrence in both active and inactive molecules. It is argued that a fragment should be strictly active-influenced if such a fragment is present in more active molecules than inactive ones, and vice versa. This property is referred to as the weight activity tendency. To illustrate this feature, see Table 5.2. Several familiar variables common to most SSA weighting schemes (Chapter 3) were identified for use as weight regulation, and these variables are listed in Table 5.2(a). The variables *NACT* and *NINACT* represent the number of active and inactive molecules in the dataset respectively. *ACT(I)* is defined as the number of active molecules containing fragment F_i while *INACT(I)* is defined as the number of inactive molecules containing fragment F_i . From these determinations, two ratio values can be calculated: the rate of actives *ROA* is defined as Equation 5.2 and the rate of inactives *ROI* is defined as Equation 5.3. The equations of both rates are given below:

$$ROA = \frac{ACT(I)}{NACT} \quad (\text{Equation 5.2})$$

$$ROI = \frac{INACT(I)}{NINACT} \quad (\text{Equation 5.3})$$

ROA is defined as the total active molecules (*ACT*) containing fragment F_i against *NACT*, while *ROI* is defined as the total number of inactive molecules (*INACT*) over *NINACT*. These are shown in Table 5.2(b). In the case of fragment F_1 , its *ROA* is determined as 0.333, against a *ROI* of 1. This therefore means that the fragment F_1 has a larger presence in the inactive molecules group. For the fragment F_2 , the *ROA* and *ROI* are 0.667 and 0 respectively, thus conveying the weight polarity as having a greater active molecule presence. It is argued that the weight tendency can be further used as a representation of fragment influence, and this can be interpreted in terms of positive and negative weight values. This behaviour is observed to be similar to the fragment weights distribution when using the SSA R4 weighting scheme.

Table 5.2(c) shows the weight values and their polarity when calculated using the SSA R4 scheme. It was observed that both the SSA R4 weight polarity correlate to the simple *ROA* and *ROI* case in Table 5.2(b). Based on this, a strong justification can be made for combining the weight polarity criterion and constraining the GA operation itself. First, the SSA R4 weighting scheme was selected to derive fragment polarity restriction. The reason is that it is formulated algebraically and is more sophisticated than the *ROA* and *ROI* examples shown above. Second, it is clear that the initialisation of chromosomes does not necessarily have to be purely random. The SSA R4-based weight polarity restriction can be applied during the initialisation of chromosomes itself. Table 5.3 shows the example of the fitness determination process; but this time with a SSA R4-weight polarity based values. In this example, the weight introduced was randomised and constrained to its SSA R4 equivalent polarity, as in Table 5.3(a). The subsequent scoring and ranking, as shown in Tables 5.3(b-d), allows for the active compounds to be placed in the top ranking much more easily. This is achieved by lowering the irrelevant fragment scores. Figure 5.8 highlights an updated workflow of the GA, with inclusion of the weight polarity feature. The GA flowchart highlights two additional processes in the workflow: (i) weight polarity determination, and (ii) enforcement of weight polarity. All of the other operations remain the same, including the fitness function definition. The process is also described in Algorithm 5.3 below.

Algorithm 5.3: Modified GA workflow based on fragment polarity limitation

- 1 : Determine fragment polarity based on SSA R4 weights.
 - 2 : Initialise a population of chromosomes based on fragment polarity
 - 3 : Evaluate the fitness of individuals
 - 3 : Repeat
 - 4 : Select best individuals to be used for genetic operations
 - 5 : Perform crossover
 - 6 : Perform mutation
 - 7 : Evaluate fitness of new individual
 - 8 : Replace the worst chromosomes in the population with offspring chromosomes
 - 9 : until stopping criterion satisfied
-

5.4.3.3 Selection of chromosomes and genetic operations

The fundamental implementation of chromosome selections and genetic operations covered in the earlier sections of this chapter is described in greater detail, for use in GA-based SSA. Selection acts as the driving force in a GA by directing the genetic search towards promising regions in the search space. Selection often chooses more fit individuals in analogy to Darwin's theory of evolution – survival of fittest (Fogel, 1995). In practise however, every individual should have a chance to be selected into the mating pool, although such chances of an individual can be weighted or biased, dependent on its fitness. Selection pressure is an informal term associated with the selection scheme which indicates the probability of better individuals to be favoured over the average ones. A higher selection pressure increases the likelihood for a better individual to be selected. An extreme selection pressure, however, may cause the chromosome population to be stuck in a local maximum (and decreases diversity). A lower selection pressure on the other hand may lead to slower convergence rate and prolonged evolutions to reach optimal solutions. In principal, selection pressure drives the GA to improve the population fitness over successive generations. The convergence rate of a genetic algorithm is largely determined by the magnitude of the selection pressure, with higher selection pressures resulting in higher convergence rates and vice versa.

Selection schemes can be classified into two broad categories, namely, proportional-based, and ordinal-based selection methods. Proportionate selection method picks out individuals based on their fitness values relative to the fitness of the other individuals in the population. Roulette wheel selection is an example of proportionate selection. Ordinal-based selection method meanwhile selects individuals based on their rank within the population, and the tournament selection is a prime example of this method. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. Here, selection begins by sorting all chromosomes based on their fitness value in descending order, from the largest fitness valued chromosomes to the smallest at the bottom of the rankings. A total fitness is calculated by adding the fitness of the sorted chromosomes cumulatively. Each chromosome is then assigned a segment of the roulette wheel, where the segment is proportional to the value of the fitness of the chromosome. A higher fitness value, for example, constitutes to a larger portion of the roulette wheel. Mimicking a roulette wheel spin, a random number is generated whereby the number limit is the total cumulative fitness. The chromosome corresponding to the segment

on which roulette wheel stops, is then selected. This will be the parent chromosome used further for genetic operations.

Roulette wheel is the simplest selection approach. The rate of evolution depends on the variance of fitness's in the population. The average fitness of the population for i th generation in roulette wheel selection is calculated as:

$$\overline{FRW}_{i,j} = \frac{\sum_{j=1}^N FRW_j}{N} \quad (\text{Equation 5.4})$$

where i varies from 1 to ngen and j varies from 1 to N .
Therefore, the probability for selecting the j^{th} string is

$$PRW_j = \frac{FRW_j}{\sum_{j=1}^N FRW_j} \quad (\text{Equation 5.5})$$

where N is the population size and FRW_j is the fitness of individual j .

Apart from the simplest random-based chromosome selection, the roulette wheel approaches are implemented as shown in Algorithms 5.4.

Algorithm 5.4: Roulette wheel selection

```

Set  $k=1, j=1, i=ngen$ 
  While  $k \leq mpool$ 
    Begin
      While  $j \leq N$ 
        Begin
          Compute  $FRW_{i,j}$ 
        End
        Set  $j=1, S=0$ 
        While  $j \leq N$ 
          Begin
            Compute  $S=S+FRW_{i,j}$ 
          End
          Generate random number  $r$  from interval  $(0,S)$ 
          Set  $j=1, S=0$ 
          While  $j \leq N$ 
            Begin
              Calculate  $c_j = c_{j-1} + FRW_{i,j}$ 
              If  $r \leq c_j$ , Select the individual  $j$ 
            End
          End
        End
      End
       $k=k+1$ 
    End
  End

```

Algorithm 5.5: Tournament selection

```
1:  $P$  is population
2:  $t$  is tournament size
3:  $Best$  is individual picked at random from  $P$  depending on their fitness value
   with replacement
4: for  $i$  from 2 to  $t$  do
5:    $Next$  is individual picked at random from  $P$  depending on their fitness
   value with replacement
6:   If Fitness ( $Next$ ) > Fitness ( $Best$ ) then
7:     Select  $Next$ 
8:   return  $Best$ 
```

The tournament selection process follows the Algorithm 5.5. A pair of chromosomes is firstly selected at random, known as a pair-bracket, and their fitness values observed. The highest ranking of that pair-bracket is selected from the chromosome that has a larger fitness score. It is then carried forward to the next round. The process is repeated for another pair-bracket to obtain another optimum ranking. When two of these are selected, they are paired together in a new bracket for selection. The best chromosome in the final round is finally selected as the parent chromosome.

Following the selection of chromosomes, the immediate genetic operation is the crossover operation. This method is described via Algorithm 5.6 below. Once the two parent chromosomes are determined, a random index number is chosen with a maximum number limit of $n-1$, in which n is the size of chromosome1. The contents of the two chromosomes are swapped with each other, in which the swapping point is defined by the random number generated earlier. The crossover operation can be implemented either by producing only one offspring chromosome or both the modified parents together, equalling to two offspring chromosomes. For the implementation of the GA-based SSA, the generation of one child chromosome from the crossover operation is preferred, as this gives a greater degree of freedom for the creation of new offsprings without requiring both modified parents to be selected forcefully.

Algorithm 5.6: Crossover operation

- 1: Select parent chromosome1
 - 2: Select parent chromosome2
 - 3: Generate a random index number of up to maximum size of chromosome1 - 1
 - 4: Flip parent chromosome1 with parent chromosome2 from random index
Number to index of maximum chromosome size
-

The mutation operation is a simple one in which the chromosome (often having already undergone crossover reproduction) gets to select one particular fragment weight to mutate. From Algorithm 5.7, the fragment weight selection is done randomly and a new number is generated (within the constraints of weighting regularisation) to replace the old fragment weight.

Algorithm 5.7: Mutation operation

- 1: Select a chromosome
 - 2: Select a fragment weight to mutate
 - 3: Generate a random number from minimum weight to maximum weight value based on weight polarity
 - 4: Replace fragment weight with new generated number
-

5.5 Experimental procedure

The experiments were divided into two sections; the first experiment was performed to identify the best parameters from a varying number of options. The best determined set of parameters were then used by the GA to be executed for all activity classes from the three databases. It was necessary to list the possible set of values for each specific parameter to be rigorously tested under several categorised groups. For example, the parameters' population size and GA maximum iteration were grouped under the population and generation group. Genetic operations such as crossover and mutation rates were defined under the Evolution Control parameter group. The complete list of parameters tested and the parameter groups are discussed in Section 5.6.

The second experiment followed, and involved taking the best identified parameters as a default for the GA verification run for all activity classes, where the main purpose is to observe the performance of GA-based methods against traditional SSA weighting schemes. For benchmarking purposes, the Robertson-Sparck Jones's R4 was chosen as the weighting scheme for comparing SSA to the GA. All the GA experiments above required an initial run on the training sets, followed by predictive verification on the remnant test sets. This process was repeated three times for the parameterisation (where the worst results were selected to represent the parameter tested) and ten times for the GA benchmark experiments sets, which looked at correlation and reliability factors.

5.6 Experiment setup: Parameterisation of GA-based SSA

The stochastic and random nature of an evolutionary algorithm implies that the end result of an evolution may not always be the same. Thus it can be argued that GAs with different sets of parameters may possibly arrive at non-identical solutions. Similarly, a small change to one parameter can also give markedly different results. Roeva, Fidanova and Paprzycki (2013) among others previously investigated the influence of population size on the success of the GA, while Goldberg (1991) mainly observed GA parameters, such as crossover / mutation rates, crossover methods and parents selection to correlate to GA success. The results obtained by both authors were used as a starting point to our parameterisation experiment and further variations were explored to seek a possible fit to our search space problem.

To ensure that results obtained from the parameterisation test are consistent and noise-free, each GA parameter test is repeated three times to note any large occurring discrepancies. The worst performing result was chosen out of the three to effectively represent the results of each parameter tested. The predictive sets of the two activity classes from the MDDR database, namely the RNN and COX classes, were chosen as input datasets for the parameterisation experiments. RNN was selected, as it has the least structural diversity (is the most homogenous) of all the classes in the MDDR, WOMBAT and ChEMBL databases. This means that active compounds are more likely to be identical to other actives in the dataset. The COX class was identified to have the largest structural diversity (as being the most heterogeneous) of all the classes in the three databases.

Based on the GA program developed, and following various literatures on GA parameterisations, a key number of parameters required by the GA was identified, thus: (a)

fitness function of active compounds rate in a selected top ranked percentile; (b) chromosome population size; (c) chromosomes' weight range of integer values based on weight polarity; (d) maximum generation / evolution limits; (e) elite chromosomes, (f) crossover rate; (g) mutation rate; and (h) parent selection method for offspring generation. Each parameter is changed one at a time to systematically record the performance variation of parameters. This ensures that the effect of individual parameter variation is quantified as accurately as possible, and noise results are not mistakenly recorded. A set of initial, default parameter values were first defined, which consisted of a fitness function of the number of active compounds in the top one percent, a 200 chromosome population size, weight range between integer values of -100 to +100, 300 maximum iterations, a roulette wheel parent selection, elitism of 1 chromosome preservation between evolutions (mimicking simple-state model), a one-point crossover method of 0.95 probability rate and a mutation rate of 0.05, as the default GA parameters. Individual parameters being investigated were changed while keeping the other parameter set mentioned above intact. The parameters were performed and the results obtained, are discussed below. To distinguish the most effective parameter, the highest values were highlighted as shown in Table 5.4 to 5.8.

5.6.1 Fitness function

First discussed in Section 5.4.3.1, the fitness function for the GA-based SSA is defined as the number of active compounds found in a chosen cluster of ranked compounds. Such cluster can be based on the top 1% ranking, or a larger cluster such as the top 10%. The first parameterisation test was conducted here to assess the difference of using varying fitness function score based on either the active rates in the top 1%, or the top 10% ranking.

Figures 5.9 and 5.10 shows the cumulative recall plots of the different fitness functions used on both the MDDR RNN and COX activity class respectively. For the MDDR RNN class, applying a fitness function score based on the top 1% (Figure 5.9a) recorded improvement in actives retrieval rate over the SSA R4. This trend is seen in the other percentiles, apart from a drop specifically at the top 6% ranking. For the fitness function case based on the top 10% ranking (Figure 5.9b), however, the actives retrieval rate is seen to struggle against the SSA R4. In the case of MDDR COX class, the fitness function score based on the top 1% ranking (Figure 5.10a) managed improvement in actives rates over the SSA R4. This is true especially in the top 1% of ranked compounds, and in a majority of other percentile up to the top 10% ranking. Meanwhile, for the fitness function score based on the top 10% ranking case (Figure

5.10b), the GA is seen to struggle in the top 1% of ranked compounds. The active retrieval rates are improved, however, in other ranked percentiles. Table 5.4 further shows the actual actives recall rate in the top 1% for the different fitness function experimented. The fitness function score based on the top 1% ranking is shown to be superior to the one based on the top 10% ranking. This is true for both cases of MDDR RNN and COX classes. From these, the fitness function score of the number of active molecules in the top 1% ranking is chosen as the preferred fitness function definition.

5.6.2 GA weight range of chromosomes

Weight polarity feature for weight distribution and regularisation to overcome overfitting was discussed in Section 5.3.3.2. This follows experimentation with different values of minimum and maximum weight ranges in conjunction with chromosome values initialisation. It was identified that the weights should represent a series of integers values to reflect suitable scores for each individual fragment. A series of weight ranges with different combinations of either negative, or positive, or both number limits were investigated. From the results in Table 5.5, positive-only weights of both “0 to +10” and “0 to +100” performed very poorly, with the enrichment value at one percent recording much lower values than even the SSA R4 schemes. This is evident in the COX activity class, where no active compounds were ranked in the top 1% when using the weights range combination of 0 to 10. A larger positive-only weight range of “0 to +100” also achieved low results for test sets from both activity classes. Meanwhile, the use of both negative and positive weight values improved the active retrieval rates rapidly. It was observed that the weight range combination of “-100 to +100” provided the best performance of ranked compounds, while a larger negative and positive weight range combination did not yield any significant improvements in terms of actives retrieved. From this, the weight limit combination of “-100 to +100” were chosen, in conjunction with weight polarity restriction.

5.6.3 Population size and generations of evolution

Two parameters were tested under this category: (i) Chromosome population size, and (ii) maximum generation of genetic evolution. For the first parameter in the category, varying population sizes of 100, 200, 300, 400 and a maximum size of 500 chromosome population were tested, as shown in Table 5.6. In both cases of RNN and COX activity classes, a population size of 200, showed peak retrieval of active compounds in the top 1% of the ranked molecules data on test sets. Higher population sizes of 300 and above did not show

any considerable improvements, although a population of 500 chromosomes does achieve identical results to the 200 population size case. It is noted, however, that there was a difference in the run time which doubled the total GA running time of the 200 population size.

The maximum GA iterations parameter restricts the number of GA iterations leading to termination of the GA program itself when the most desirable fitness function is not reached. For this specific parameter, the test is done by executing a single instance of the GA imposed with a maximum GA iteration of 500. While running, the performance of active retrieved in the top 1% was recorded at observed iterations of 100, 200, 300 and 500. The worst performing GA instance of the three runs is selected to represent the parameter assessment. From the tests, it was confirmed that the maximum GA iteration of 200 recorded peak actives retrieval was in the top 1% for both cases of RNN and COX activity classes. Above this iteration, the rate of retrieval remained constant, although the GA instance of the COX class at 500 iterations recorded a slight increase in the enrichment rate. In general, however, the 200 maximum iteration parameter showed a marginal increase in active rates, given such a large iteration difference.

To further analyse the predictive performance of the GA method, the error plot is referred to in Figure 5.11, which tracks the error rate in relation to both the training set and its equivalent predicted test set, versus evolutionary iterations. The error rate is defined as the inverse of the fitness rate, or simply the rate of actives retrieved in the top one percent of the ranked molecules. For instance, if the fitness rate of actives in the top one percent is 0.7, then the error rate is the remnant, or 0.3. Similarly, the error rate of one fitness of 0.5, is also 0.5. The worst GA run from the iteration parameterisation test (as in Table 5.6) was selected and its retrieval performance plotted, up to the 500th iterations. For both the MDDR activity classes RNN (Figure 5.11a) and COX (Figure 5.11b), the figure shows consistent error rates between the training and test set, signifying a good correlation of the predictive performance especially in unseen datasets, such as the test set. It was demonstrated that for longer iterations, the GA did not yield any improved weighting scheme for improve the actives retrieval rates for both the training and test set data, which is also reflected in Table 5.6. It can be concluded that, considering runtime efficiencies, active retrieval and its predictive performances, the population size is set to be 200 chromosomes, and the GA maximum iterations determined as 200.

5.6.4 Elitism model

The distinction between a simple-state and the steady-state model was looked in further detail. Based on the default population list of 200 chromosomes, a variety of elite chromosome preservation methods were tested based on the elitism state. For example, a chromosome preservation of 0 represents the absolute case of a simple-state model, while a preservation of 199 chromosomes (out of 200) represents the absolute steady-state model. Note that a preservation of every chromosome in the genetic pool is not possible, as this means that no chromosomes are replaced at all between generations of evolution, defeating the purpose of the GA method as a means of natural selection and evolution.

The results of the elitism parameter test are outlined in Table 5.7. From the table, the absolute simple-state model (with zero preservation between iterations) was observed to be less effective than the same model with at least 2-parent preservation above. The simple state model with 2-parent preservation managed to score the highest retrieval rate for all values tested. Subsequent larger parent preservation did not yield any considerable improvements, while the steady-state model of 150 and 199 chromosomes preservation degraded the retrieval performance for test sets of the two activity classes. This was perhaps due to the nature of steady-state, which replaces fewer numbers of chromosomes, resulting in a slower convergence of the solution search space. To account for this requirement, another slightly different parameter set was investigated, specifically the steady-state model with one chromosome preservation but with a higher maximum iteration of 1000. It is noted that the retrieval performance still did not improve when it was increased to larger number of iterations.

5.6.5 Evolution control

A number of different parameterisation combinations were tested under the evolution control group, as listed in Table 5.8. Here, each individual parameter was changed one at a time, while others were consistently kept the same. This was so that any variance caused by the parameter change could be observed. Firstly, the parent selection method defines how two parents' chromosomes are selected to be used for the genetic operation or reproduction process. The main idea of parent selection is that better individuals get higher chances of being selected for reproduction purposes. Three methods are already established in the literature, namely the roulette wheel method, tournament selection and random selection; these three were extensively tested in our experiments. From the test results as shown in

Table 5.8, it was observed that for both activity classes, the roulette wheel method consistently obtained higher retrieval rates than the other two methods.

The crossover rate represents the probability that a crossover operation will be performed, with a possible range between 0 and 1. Here, a number of different rates were tested, ranging from 0.60 to 0.95. The difference in the performance of retrieval is marginal for such changes, although a slightly higher and consistent active retrieval was observed in the top 1% threshold of ranked dataset when using the crossover rate of 0.95. This was true in both cases of RNN and COX activity classes. The mutation rate is the percentage rate of mutation occurrence, and is represented in terms of a ratio from 0 to 1. For our test, a high active retrieval rate was recorded when using the mutation rate of 0.01 (or 1% in percentage terms). Test instances of the RNN and COX activity classes recorded higher actives via this mutation rate. Next, the crossover methods were tested between the three described in the earlier sections above, namely the: (1) one-point crossover, (2) two-point crossover; and (3) uniform crossover. It was confirmed that the one-point crossover method records the highest active retrieval rates for both RNN and COX activity classes in the test sets.

5.6.6 Final parameterisation selections

Extensive parameterisation tests were carried out in order to quantify the influence of each parameter for the optimisation of GA search. From the tests, several parameters which are critical to GA's performance in maximising active retrieval rates were identified, while other parameters were found to be less sensitive.

The GA weight range is found to be the most critical parameter as successful GA searches were only achieved when the weight value limits are extended to both negative and positive weight range limit. GA runs with positive-only weights failed to generate active recall rates which is even at least comparable to the SSA R4. The minimum weight value range acceptable was found to be of at least the -100 to +100 combination but any larger, extended limit did not yield considerable impact. The second critical parameter is the elitism model. From the tests, it was found that having no elite chromosomes, or on the contrary, too many elite parents degrade the performance of the GA. An elitism of 1 to 5 is seen to optimise GA-based results. For the population size and generation limit, the parameters were only considerably sensitive if small values were used instead, while having very large populations and a longer iteration did not yield any increase in active retrieval performance. Finally, for

the evolution control parameters which include the parent selection method, crossover and mutation rate, only fine tuning of the parameters were required to maximise active retrieval performance.

Based on the parameterisation results shown in this section, it was decided that the following GA parameters were appropriate for the next experiment in this chapter. For the GA performance benchmark against SSA, and given all the activity classes from all three databases, the finalised parameters are as follows: (1) *Fitness function score based on number of active compounds in the top 1%*; (2) *GA weight range of -100 to +100*; (3) *a population size of 200 chromosomes*; (4) *a maximum of 200 GA iterations*; (5) *Simple-state model with elitism of 2 chromosomes*; (6) *Chromosome parents selection using the roulette wheel method*; (7) *a crossover rate of 0.95*; (8) *a mutation rate of 0.01*; and (9) *the one-point crossover method*.

5.7 Experiment result: Analysis of performance of GA-based SSA

Based on the finalised parameters, the GA-based SSA was properly executed to quantify its performance relative to the R4 weighting scheme. The R4 scheme was found in general to be the most consistent scheme in the SSA, as discussed earlier in Chapter 4, Section 4.3. Several analyses and tests were also conducted to gauge the level of GA's effectiveness. These are categorised below:

5.7.1 GA robustness

To quantify the randomness factor in the solutions obtained from the GA method, a robustness test was performed by running and analysing results from multiple instances of the GA. This was done using the best set of parameterisations observed earlier in Section 5.6.6. The two activity classes RNN and COX from the MDDR database were used as input for the GA robustness test. For all ten runs of the GA-based SSA and their subsequent application of the weights to their respective test sets, retrieval rates of the two classes above were plotted as enrichment curves, shown in Figure 5.12 (a-b). Based on the plots, all ten GA runs were observed to obtain improved active retrieval rates in the top 1% of ranked compounds compared to SSA R4 results. This trend continued with small to fairly small deviations between each GA run for the rest of the remaining top ranked percentile, up to the top 10%. From here, it can be concluded that the GA-based SSA does manage to produce effective

consistent results, albeit with the small deviations mentioned, confirming the robustness of the GA results.

5.7.2 GA weights correlation and consistency of compounds retrieval

The measure of the relationships of all the GA weights generated by the 10 runs for each MDDR RNN and COX classes (Section 5.7.1) were investigated using Pearson's *r* correlation coefficient. The Pearson's *r* correlation coefficient was used to measure how well the generated GA weights are related in the multiple runs. The correlation results for both activity classes are presented in multi-correlation plot format.

For the RNN class, high correlation values with a minimum of 0.74 were observed between individual GA run instances, and for the COX class, a minimum correlation value of 0.77. The mean and standard deviation for the Pearson's *r* averaged over the 45 pairs of runs for each activity class were 0.75 and 0.025 (RNN) and 0.79 and 0.024 (COX). The level of consistency was further observed from the ten revised-GA runs. Table 5.9 shows that the first GA run (dubbed GA_run1) was assigned as the reference run. Its active molecules in the top 1% were identified and compared to those of the corresponding nine remaining runs. From the table, the upper value for each run denotes the number of different active compounds in the top 1% compound ranking, which were not present in the equivalent top 1% of GA_run1. The lower values shown in brackets describe the actual number of active molecules retrieved in the top 1% of the test set of the ten GA runs.

The most obvious observation is the relatively small difference in active molecules for the other nine GA runs compared to reference GA_run1 for both activity classes. For example, only 50 active compounds were found to be different in the top 1% (or the first 922 ranked compounds) test set ranking of GA_run2 when compared to GA_run1 for the RNN activity class (Table 5.9a). It is even smaller for the COX activity class (Table 5.9b), where the different active compounds retrieved from multiple runs ranged from 13 to 21. From the total of 167 active compounds retrieved in the top 1% of COX GA_run1, this accounts for roughly a 7-12% change for the other runs. These results show a strong indication of the level of consistency of performance obtained from multiple GA runs, even when there are variations in their individual weights.

5.7.3 Analysis of GA runs on all activity classes

Having confirmed the consistency of GA results and the strong correlation of the weights for the two MDDR classes, the actual performance of GA-based SSA was verified on all activity classes from the three databases via multiple runs of the GA for each class. Results of the GA ten runs are summarised in Table 5.10 for the classes from MDDR-dataset, the WOMBAT-dataset and the ChEMBL-dataset. The tables outline three important results, which are (i) the enrichment factor of active molecules in the top 1% (ii) the mean and standard deviation of the number of actives in the top 1% for the ten GA runs and (iii) the mean correlation and standard deviation between 166 weights using Pearson correlation coefficient for the ten GA runs. Based on Table 5.10, high correlation values were observed, in which the mean correlation of Pearson's r recorded a minimum of 0.74, and on average circa 0.78. Some classes were also observed to record a mean Pearson's r as high as 0.86. From the mean and standard deviation values for the ten GA runs, it can be seen that there is a high degree of consistency of the number of actives were retrieved in the top 1% of the ranked data.

Table 5.11 presents the GA-based SSA results of all activity classes from the three databases, based on the worst GA run. Actives retrieval performance, the diversity rate and unique scaffolds were observed and compared to the SSA R4 results. The highest value was shaded lightly. As shown in Table 5.11, all activity classes from the three datasets showed an improvement with the GA in the number of active molecules retrieved compared to the SSA. It was found that the GA-based SSA retrieved more than 200 molecules against the SSA R4 in several classes such as the ChEMBL-based 5HT and ChEMBL-based HIVP. A small improvement was also observed in the actives retrieval for WOMBAT-based PKC activity class, which scored only 93 for retrieval for GA versus 92 active molecules for SSA R4. It is deduced that the most likely possible cause of this is the fact that the WOMBAT-based PKC class contains the smallest number of active molecules, totalling 142 compounds out of the total 134,812 compounds for the whole of the WOMBAT database. An equivalent training set of 10% distribution makes up only a total of 14 active molecules against 13,481 inactive molecules. This is considered to be too small to be used in the sample test set, despite the slightly improved retrieval results.

5.7.3.1 Enrichment curve analysis

The effectiveness of screening is illustrated diagrammatically by the enrichment (or cumulative recall) plots shown in Figures 5.13 to 5.17, representing selected activity classes

from the MDDR, WOMBAT and ChEMBL datasets. The selected activity classes are the 5HT3, COX, D2, RNN and PKC and the worst GA run results are chosen to represent the classes. Each such curve shows the percentage of the actives retrieved in the top 1% of the ranked predictive test set up to the top 10% (since it is only the top-ranked molecules that are of interest in a virtual screening context). From the plots, three distinct trends were observed.

The first trend characterises the GA curves which are consistently above the SSA R4, up to the top 10% of ranked molecules. Example plots are from the MDDR and ChEMBL 5HT3 activity classes (Figure 5.13a, c). The MDDR COX (Figure 5.14a) and MDDR PKC classes (Figure 5.17a) both have a somewhat lesser impact for GA in the top ranking, but are still superior up to the top 10% of ranked compounds nonetheless. The second trend is defined where the GA curve sits above the SSA R4 in the first few portion of the ranked molecules, but approaches the SSA R4 curve in later part of the ranked list. An example is the WOMBAT COX (Figure 5.14b) class. The third trend defines the case where the GA curve sits above the SSA R4 in the first few portion of the ranked molecules, but crosses the SSA R4. Such examples are observed in the ChEMBL COX (Figure 5.14c), MDDR, WOMBAT and ChEMBL D2 (Figure 5.15a-c), and MDDR and WOMBAT RNN (Figure 5.16a, b) classes. A lesser variation of the third trend is found in the following classes, where the retrieval performance of the GA-based SSA is superior only in the top 1% or 2%, then crosses downwards against the SSA R4. The classes shown are in WOMBAT 5HT3 (Figure 5.13b), ChEMBL RNN (Figure 5.16c), and WOMBAT and ChEMBL PKC (Figure 5.17b, c) classes.

In general, based on the fact the worst GA run is used for comparison, the GA was found to be consistently and often markedly superior to the SSA R4. This in terms of the enrichment factor of actives retrieved in the top 1% ranked molecules region (full results indicated in Table 5.10). Some additional benefits include consistent superiority of the GA results in the later part of the ranked molecules for a number of activity classes as well.

5.7.3.2 Analysis of diversity

Diversity analysis was conducted on the GA results to quantify their ability to identify novel bioactive compounds from a diverse space of possible compounds. The methodology of this analysis is described in detail in Chapter 3. For this analysis, a breakdown of the diversity of actives retrieved using the GA-based weighting scheme and SSA R4 weighting scheme is

presented in Table 5.11. The table presents two important results, thus: (i) the number of distinct Murcko scaffolds in the top-ranked actives and (ii) the diversity rate (based on Tanimoto coefficient). From the table, the GA-based SSA consistently recorded a higher diversity of actives retrieved in the top 1% of ranked molecules for all three datasets used. Furthermore, the table also shows that the most heterogeneous activity classes in the top 1% for each database were found to be the COX (MDDR), PDE (WOMBAT) and COX (ChEMBL) when using the GA. Distinct scaffold results of both MDDR and ChEMBL results when using the GA method signifies a good improvement of the method when compared to SSA R4, except for the WOMBAT-based PKC class, which showed a somewhat similar result.

5.7.4 Wilcoxon signed rank test

Varied improvements were observed in the active recall rates of the GA results when compared to the SSA R4 method. To quantify the significance of the performance, results were evaluated using the Wilcoxon signed rank test as a statistical measure of the hypothetical test. In this test, all the top 1% actives of activity classes in the three datasets are collected and their significance of difference tested using the Wilcoxon signed ranks test. More explanation about the Wilcoxon signed rank test can also be found in Chapter 3.

The results of both GA and SSA R4 models were found to be significantly different at the 0.01 significance level for both MDDR and WOMBAT databases based on the W values for both which are 0, and at such significance level, the critical value $W_{critical}$ is 5 and 12 respectively. This is more than the value of W , and hence a significant result for both cases (Table 5.13 and Table 5.14 respectively). In the case of the ChEMBL database as shown in Table 5.15, the conclusion is drawn where both GA and SSA R4 results are significantly different based on the value of $W = 0$, which is less than the critical value i.e. $W_{critical}$ of 15 for $N = 15$, given the confidence level of 99% (0.01 significance level). In summary, it can be concluded that GA-based SSA is superior to SSA R4 method in all MDDR, WOMBAT and ChEMBL databases. Also, the null hypothesis which states that the results from the GA-based SSA and from the SSA R4 model are identical can be rejected.

5.7.5 Model validation with Y-randomisation

A popular approach to validating a predictive method is through the use of Y-scrambling test. In chemoinformatics, this validation is performed by taking a dataset consisting of descriptors

and bioactivities and randomising the bioactivities while retaining the descriptors, effectively scrambling the dataset and making it nonsensical. The main objective of this method therefore is to validate whether a particular method in question is able to produce outcomes that predict a model, even with deformed data. A model that has a strong correlation in terms of performance when applied to both scrambled and predicted datasets can be argued to be unreliable, as it is not responsive to valid bioactivities and descriptors relationship. Y-randomisation (otherwise known as the permutation test) was performed to determine statistical validity and the reliability of solutions generated by the GA. Klopman and Kalos (1985) first introduced the Y-randomisation test as a method for validating the existence of a chance correlation. This method takes an *N*-compound dataset consisting of both *X*-value and *Y*-identifier data and repetitively scrambles the *Y*-identifier while leaving the *X*-value portion intact. Each scrambling is followed by applying the machine learning method in question to the so-called scrambled data and recording the subsequent performance. In our study, one of the important variables to be analysed is the correlation value (usually Pearson's *r*) between the sets of 166 weights computed for each distinct pair of runs for the scrambled instances. The observed values should remain low so that one can remain confident about the relevance and reliability of the GA method proposed (Rucker, Rucker and Meringer, 2007).

For the scrambling test, an initial training set comprising 10% of active and 10% of inactive compounds from the MDDR-based RNN and COX activity classes were selected. The compound identifiers were scrambled, while leaving the fingerprints and compound activity state intact. The scrambling was repeated 100 times to generate 100 new, individual, scrambled training sets. Each of the scrambled training sets was then applied into the GA for prediction. The resultant weights given by the GA, based on the scrambled sets, were then applied to the test set.

Recall the ten repeated GA runs, as discussed in Section 5.7.1 (GA robustness test). For the purpose of model validation, the 10 GA instances are dubbed as the unscrambled GA runs. Subsequently, two meaningful variables to be observed were identified: (i) Pearson's *r* coefficient, based on the weight distribution when compared to the first run of the unscrambled GA result, hereby dubbed `unscrambled_GA_run1`; and (ii), the active molecule retrieval rate in the top 1% of the compound ranking. The hypothesis is that the scrambled GA runs should not match or correlate well with unscrambled GA runs, since in doing so, means that there is the presence of chance correlation using deformed data. To calculate

Pearson's r , the set of 166 weights of the unscrambled_GA_run1 were correlated with the sets of 166 weights obtained from the 100 scrambled GA runs. The remaining nine unscrambled GA runs were also correlated with the unscrambled_GA_run1. Similarly, the percentage of actives retrieved in the top 1% of active compounds was observed for both the scrambled and unscrambled GA results.

Figure 5.18(a) and 5.18(b) show the Y -randomisation plot for the two MDDR-based activity classes; the RNN and COX. The X -axis of the plot represents the retrieval rate in the top 1% ranking, while the Y -axis denotes the Pearson's r values versus the unscrambled GA_run1. From both figures, the ten unscrambled GA results performed positively, clustering in the top right side, which signals a similarly high active retrieval rate, while maintaining a good correlation with the unscrambled GA_run1. This is the opposite of the scrambled GA runs, which recorded very low correlation values of Pearson's r (none higher than 0.30). All of the scrambled results also failed to retrieve comparable actives in the top 1% of the ranked compounds for both classes, especially when compared to the unscrambled ones. The mean and standard deviation of the number of actives retrieved in the top 1% of the scrambled GA cases were 4.16 and 8.10 (RNN). Two standard deviations above and below the mean creates a range from -12.04 to 20.36. So, of the 100 runs, 94 runs fall into this range, which is 94% of the runs. Meanwhile, the mean and standard deviation of the numbers of actives retrieved in the top 1% of the scrambled GA cases for the COX were 5.31 and 6.96. Two standard deviations from the mean create a range from -8.61 to 19.23, which means that 94 runs fall into the range. Based on these results, it can be seen that the GA-based SSA is unable to arrive at chance correlations with deformed data.

5.7.6 Run-time benchmarks of GA-based SSA

Execution of the GA-based SSA was fairly intensive on a single computer, depending on data size and choice of parameters. Performance of the GA program with different hardware was monitored and documented, followed by run-time analysis, in order to understand hardware suitability in different data scenarios. The hardware used follows the ones listed in Table 3.2.

For the benchmark test, the RNN activity class was selected for all three database instances, with the same parameterisation set as finalised in Section 5.6.6. Table 5.12 lists the breakdown of run-time of the GA for individual iteration and subsequently the total run-time for a complete GA program. These are based on 200 maximum GA iterations. For the

MDDR-based RNN class, the 10% training set is made up of 10,254 compounds, while for the WOMBAT-based class, 10% training set is equivalent to a total of 13,812 compounds. The ChEMBL-based RNN activity class meanwhile is significantly larger for the case of a 10% training set, consisting of 135,267 compounds.

For the MDDR-based RNN activity class, as shown in Table 5.12(a), a GA run using 10% training set clocks in between 2.10 to 3.30 seconds per GA iteration, with a total run-time for the GA program averaging between 468.51 seconds to 660.38 seconds; these are equivalent to 7 and 11 minutes respectively. Table 5.12(b) shows the run-time breakdown for the WOMBAT-based RNN activity class via a 10% training set of the GA run. The average run time for a single iteration is between 2.77 and 4.10 seconds, with the total run-time of all 200 iterations averaging between 551.23 and 911.95 seconds (9 and 15 minutes), respectively. For the ChEMBL case, as shown in Table 5.12(c), it is highlighted that the machine WKST_01 was not able to execute the GA-based SSA due to its small memory limitation problem of having only 4GB DDR physical RAM. Between the SERVER and WKST_01 machines, the run-time of a single iteration was clocked-in at 34.20 and 40.10 seconds respectively. The total run-time of a complete GA run in the ChEMBL case was 7011.20 seconds for the SERVER machine and 8450 seconds for the WKST_01 machine; this translates to 116 and 140 minutes respectively.

Two objective opinions can be drawn from the above, whereby the primary requirement for a GA-based SSA is in the physical memory limitation of the particular hardware, especially if the size of the dataset is equivalent or larger than the ChEMBL database. This can be mitigated, though, via a smart code optimisation or a programmable data handling feature in order to accommodate the limited memory issue. In terms of run-time performance, a fairly mild influence of the machine's processor architecture was observed in increasing run-time efficiencies, but this should not be a critical factor in the consideration of hardware choice for running the GA-based SSA.

5.8 Discussion

Various analyses were performed to gauge the performance level of the GA-based SSA. The primary method of analysis was based on the active retrieval rate for each activity class in the three databases, which was compared to equivalent implementations of the SSA R4. This was determined as the most effective scheme out of the existing ones (as discussed in Chapter 4).

The screening results were analysed by comparing the SSA R4 results with the worst of the 10 GA runs executed for each activity class. Based on the analyses, it was observed that the GA method outperformed the SSA R4 in all cases of activity classes. The GAs executed for each activity class recorded high correlation values and consistency in terms of active compound retrieval as shown in the robustness and consistency tests. GA also showed a larger number of Murcko scaffolds and a higher diversity rate of actives in the top 1% when compared to the SSA R4 for all cases of activity classes. Permutation tests for model validation proved that the GA would not be able to generate a successful solution if trained from randomly generated datasets.

Cumulative recall plots showed that the GA is superior to the SSA R4 in the top 1% of the ranked compounds for the majority of classes. Most of the curves represent a GA trend that consistently outperforms the SSA R4 up to the top 10% ranking, but a number of classes demonstrate GA curve behaviour where they cross or dip down below the SSA R4 in later percentiles. A Wilcoxon signed rank test was also performed to measure the significance of the difference between the performances of the GA and SSA R4. The Wilcoxon test indicated that there is a significant difference at the $p < 0.01$ level in the performance between the worst GA and the SSA R4 results. The performance of the GA-based method was statistically proven to be better than that of the SSA R4 runs. In terms of suitability of the GA-based SSA in real world application, two tests performed are related for this assessment. Firstly, parameterisation tests have established the GA-based SSA's parameter sensitivity to be critical of only a number of specific parameter conditions. The most important parameters are the negative-to-positive weight range limit, followed by an acceptable elitism model. Other parameters did not significantly affect GA's performance as the ones listed above, in which only fine-tuning requirements were stressed. Secondly, results of the run-time benchmark indicates that the GA-based SSA require standard hardware resource with the primary emphasis on a large physical memory availability (depending on the size of compounds database). Both results above affirm the GA's practicality in real world applications.

5.9 Conclusion

Chapter 4 previously compared ten SSA weighting schemes and it was found that the Robertson / Sparck-Jones R4 scheme to be consistent in retrieving the highest (or among the highest) active molecules for the majority of activity classes in MDDR, WOMBAT and ChEMBL datasets. This chapter looked at expanding the SSA method by applying a GA-

based weighting scheme to determine the suitable set of fragment weights for any possibility of an upper-bound to the activity prediction of the SSA.

In order to address our second research objectives outlined in this thesis, the GA experiment results were benchmarked against the SSA method represented by the R4 weighting scheme. In this chapter, it can be concluded that the GA-based SSA method is superior to the SSA R4 scheme, as it successfully manages to provide uplift in the upper-bound of active retrieval performance in the top 1% of ranked molecules. The active compounds retrieved by the GA also show improved diversity rates and larger amount of scaffolds than its SSA R4 counterpart. Unlike the SSA, the GA-based approach is considered to be an inherently non-deterministic process. High correlation and consistency between multiple GA runs however means that the method is reliable and effective as an alternative weighting scheme to the SSA method.

The GA approach proposed in this chapter has proven to be able to improve on its main objective (active retrievals) when compared to the SSA existing weighting schemes. The aim of our GA was purely to maximise the number of retrieved active molecules without depending on any other optimisation supporting criteria. The key components of the proposed GA are chromosome randomisation and a continuous but often unpredictable series of evolutions to arrive at a preferable weighting scheme. Despite being a non-deterministic method, this study proved that the results obtained are more consistently effective than those obtained from existing, deterministic methods for generating such weights. It is therefore strongly recommended for the SSA to be further enhanced via the use of a GA in determining the best fragment weights combination. This finding hopefully will contribute to standard practise in ligand-based virtual screening and guide further enhancement in SSA.

The next chapter of this thesis focuses on exploring the validity of another type of evolutionary algorithm, known as the Genetic Programming (GP), which utilises program evolution to represent potential solutions for an efficient weighting scheme.

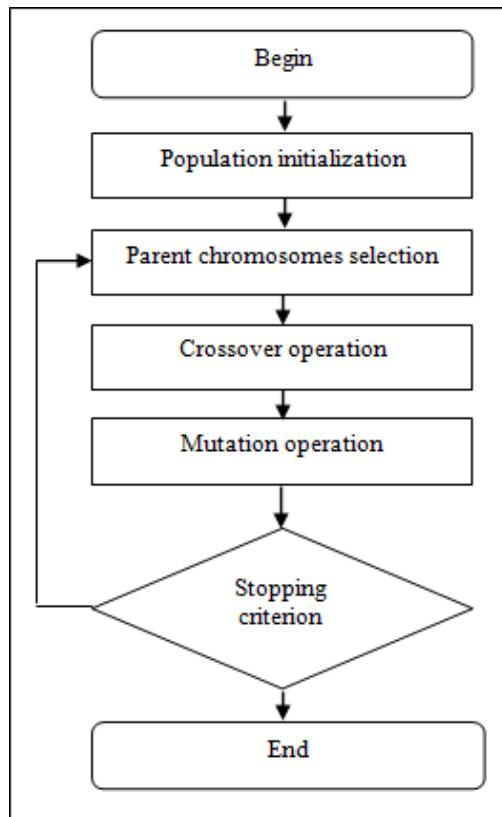


Figure 5.1: The basic genetic algorithm flowchart

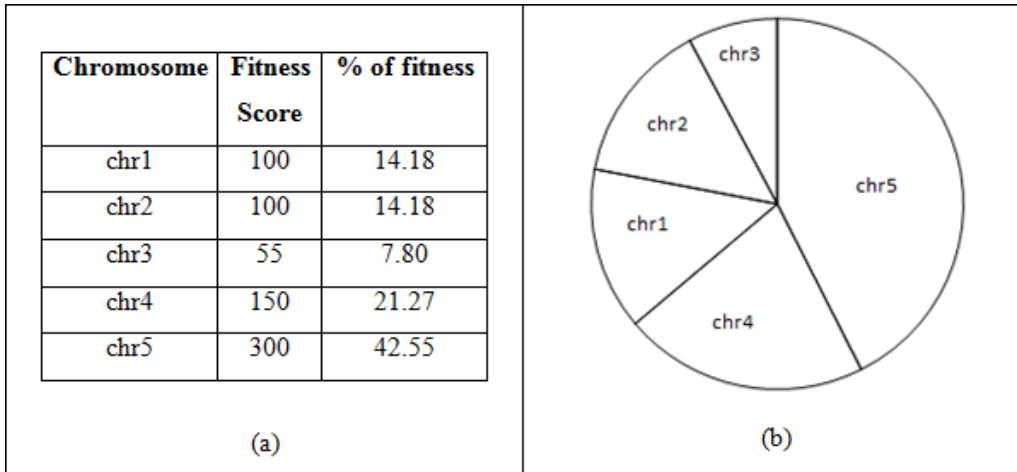


Figure 5.2: Roulette wheel selection after Goldberg (1987). (a) Outlines a set of evaluated chromosomes with different fitness scores, and their relative percentage of the total fitness.

(b) The chromosomes are sorted and fitted into a roulette wheel model where larger chromosomes take a bigger portion of the wheel. A random number generated ranging from 0 to 100% will iterate through the wheel until the value is achieved, thus selecting the parent chromosome

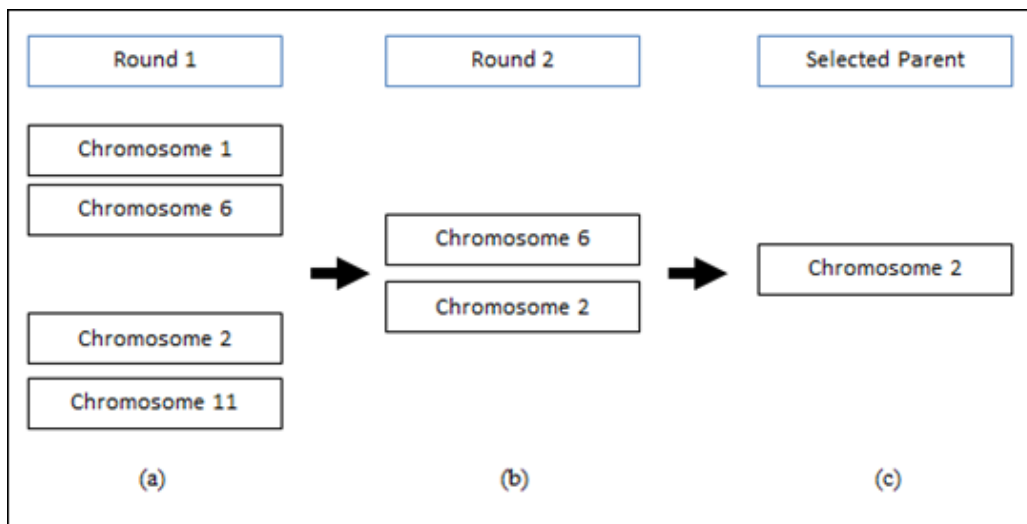


Figure 5.3: Tournament selection. (a) Four chromosomes are selected at random and assigned as paired opponents. Fitness scores are observed between opponents and the winner progresses to the next round. (b) The winners of round 1 pitted against one another in round 2 by observing their fitness score. (c) The winner of round 2 is selected as the parent chromosome for the next genetic operation. This process is repeated to select the other parent chromosome as genetic operation requires two parents to proceed

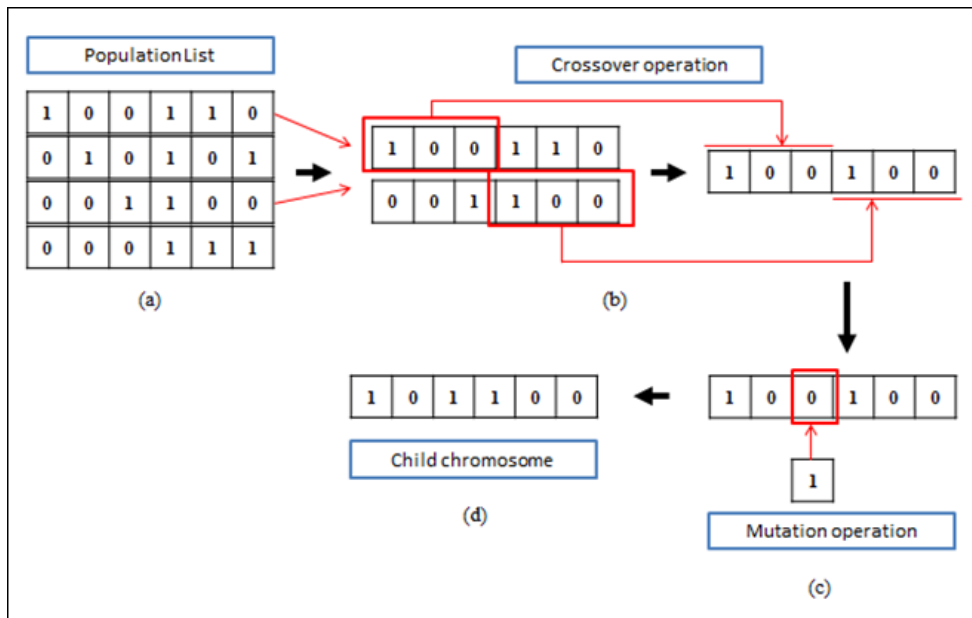


Figure 5.4: Genetic operations in the genetic algorithm. (a) A population list consisting of chromosomes represented as bit-strings. (b) Two parent chromosomes selected from the population list to perform a crossover operation, which takes a portion of each chromosome's genes and recombines them into a single, new chromosome. (c) Mutation operation flips a random bit of the chromosome. (d) Child chromosome inserted back into the population list

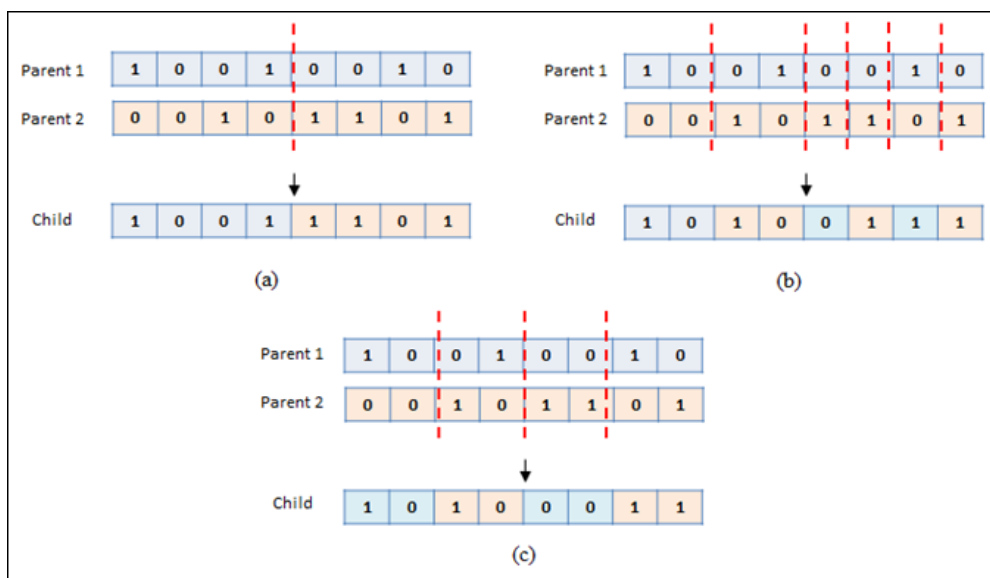


Figure 5.5: Crossover methods in the GA. (a) One point crossover method; (b) Two points crossover, and (c) Uniform crossover

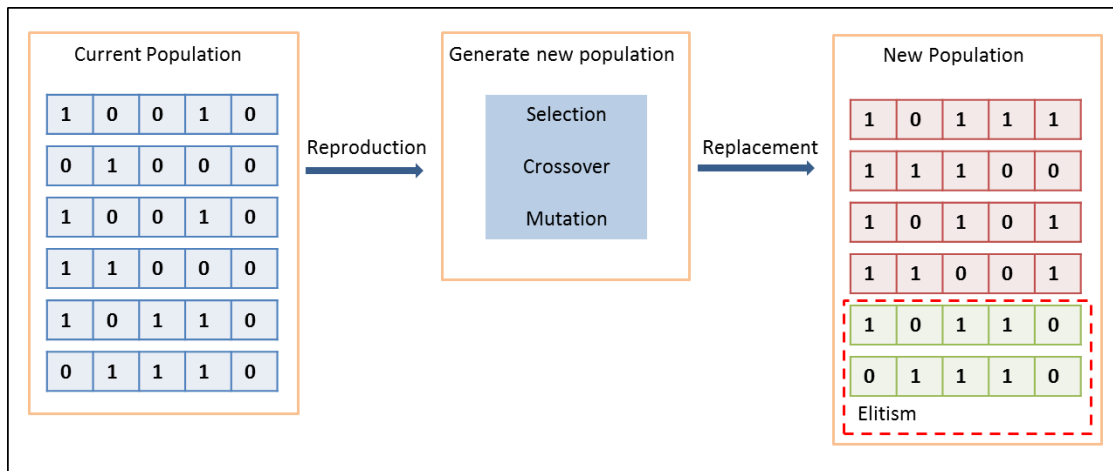


Figure 5.6: Simple-state elitism model with six chromosomes created at initialisation. After going through reproduction process, each generation concludes with modification to the population. In this case, an elitism of two parents ensures that all the chromosomes are replaced through genetic operations except for the two best parent chromosomes

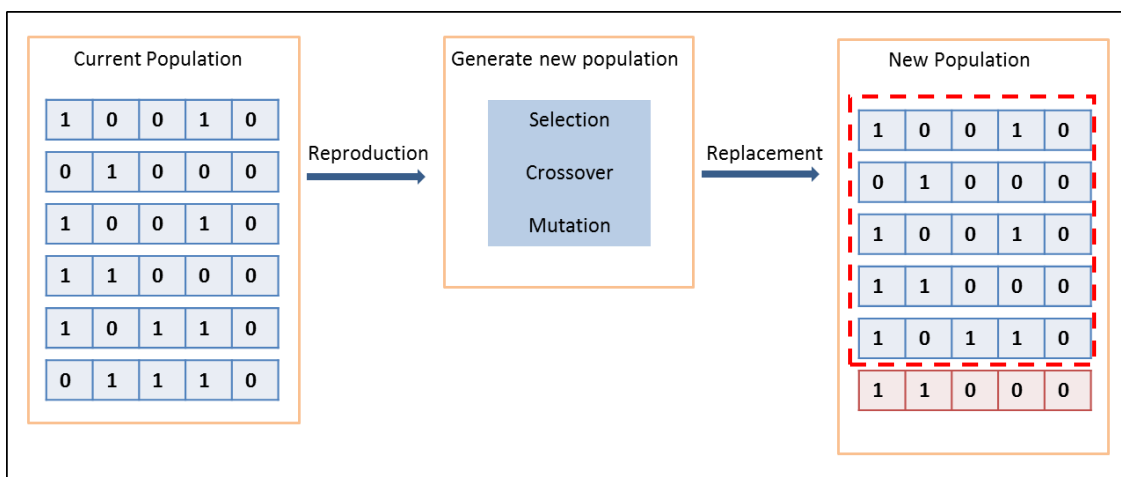


Figure 5.7: Steady-state elitism model with six chromosomes created at initialisation. During the replacement process only one chromosome, being the worst performing one, is replaced with a reproduced, offspring chromosome. The remaining chromosomes are maintained, which is also known as the overlapping population method

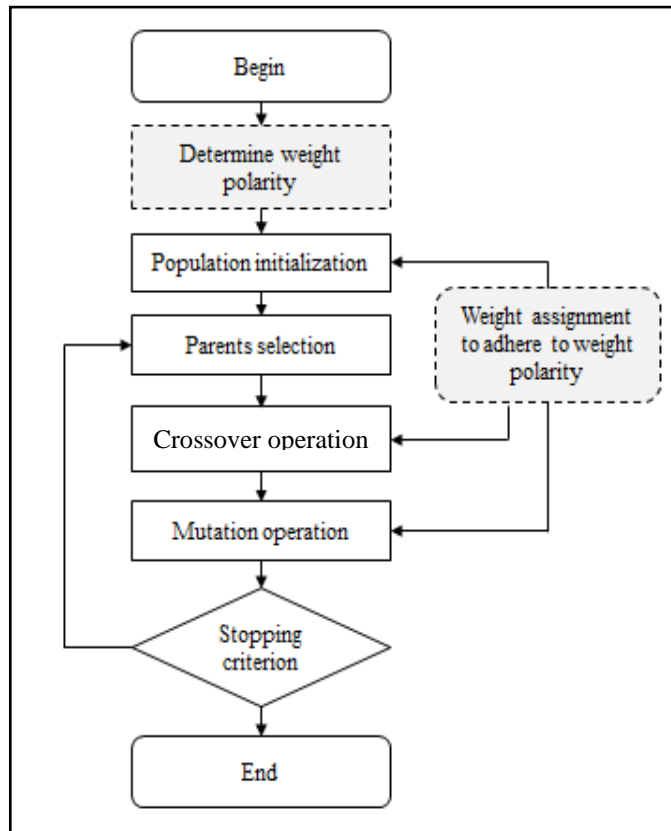


Figure 5.8: The genetic algorithm flowchart with inclusion of weight polarity constraining operations. The GA assumes normal operation except that the weight polarity needs to be identified first, and both the population initialisation and subsequent genetic operations include conditional weight assignment based on the polarity criterion

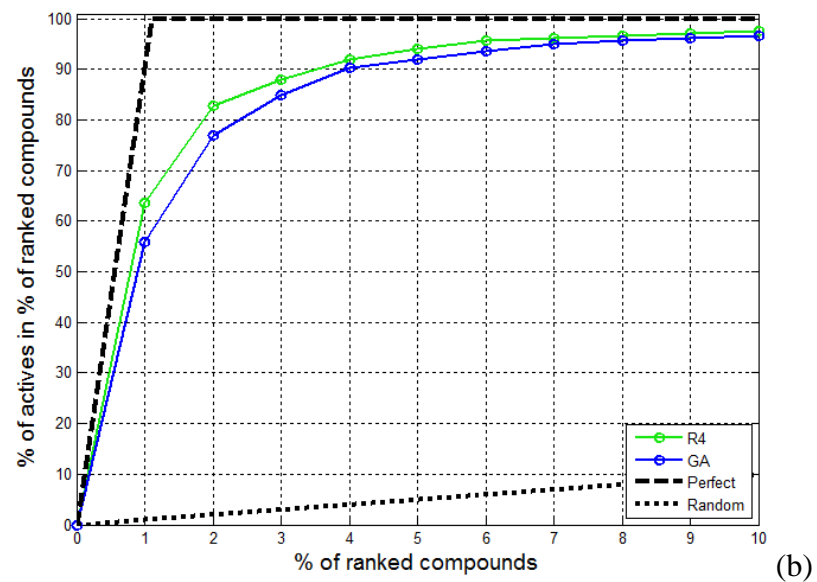
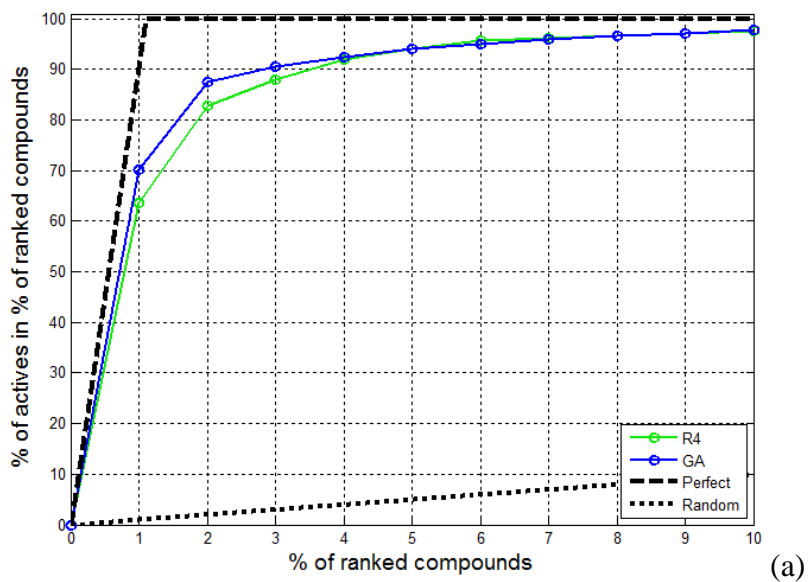


Figure 5.9: Cumulative recall plots of the GA-based SSA against SSA R4 for the RNN activity class from the MDDR dataset based on the different fitness function (a) In the top 1%; and (b) In top 10% of ranked compounds

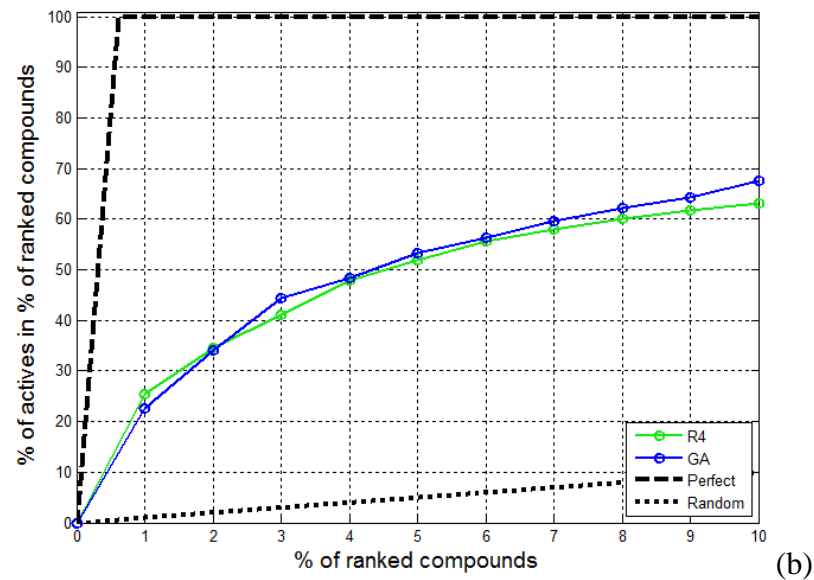
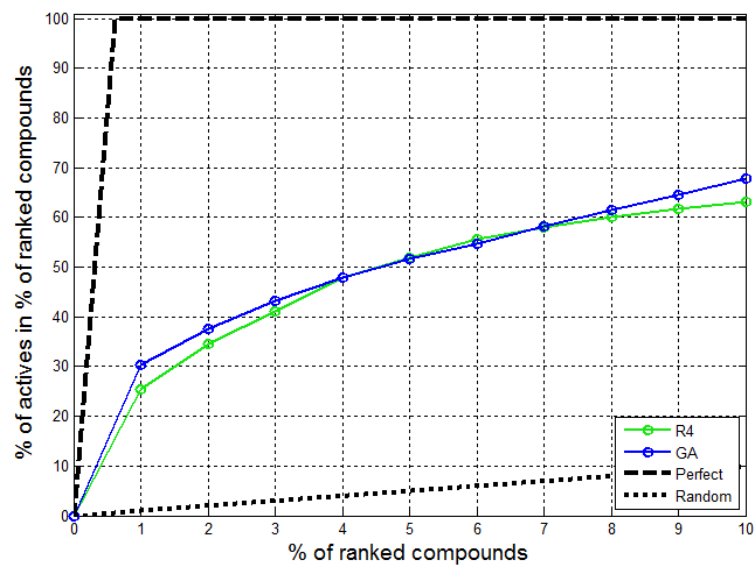
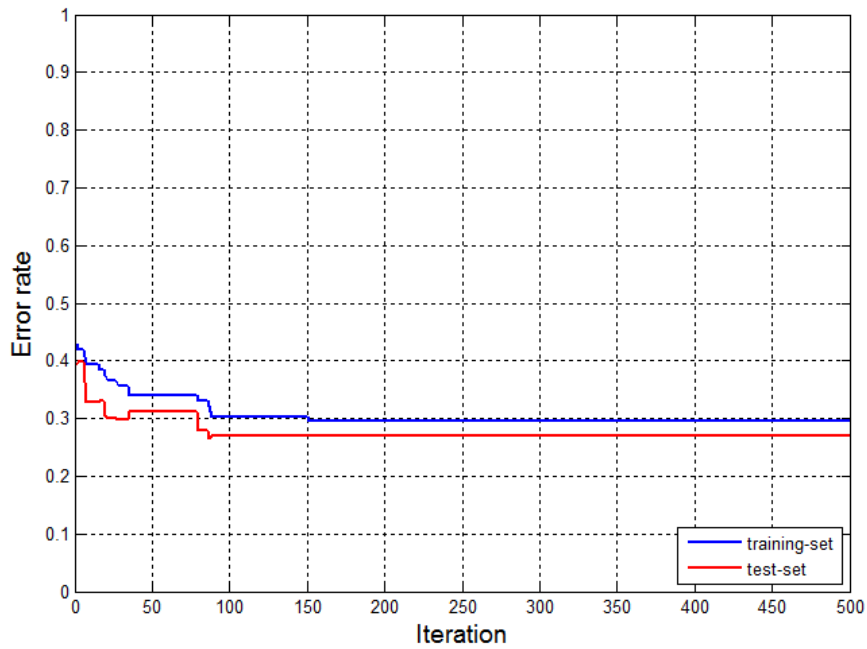
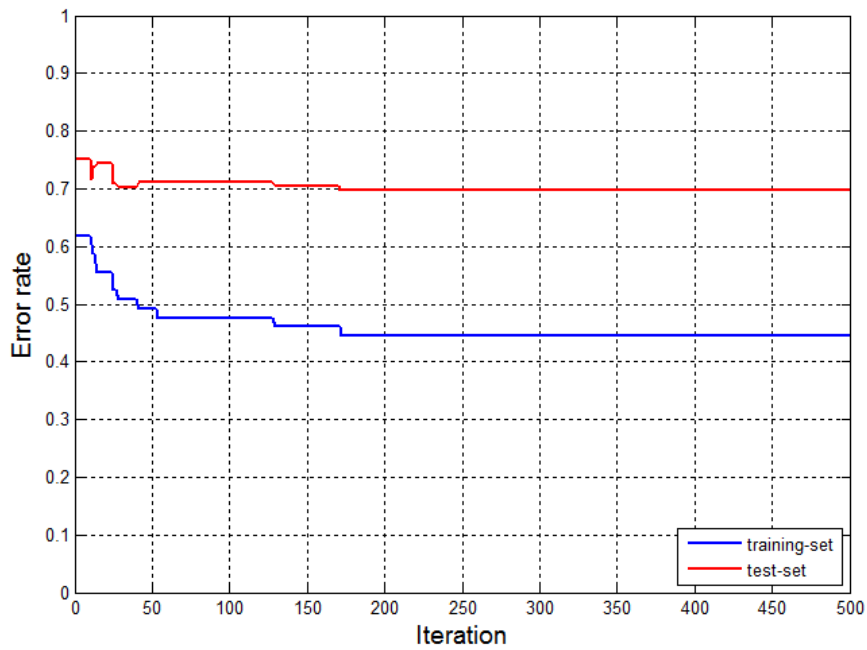


Figure 5.10: Cumulative recall plots of the GA-based SSA against SSA R4 for the COX activity class from the MDDR dataset based on the different fitness function (a) In the top 1%; and (b) In top 10% of ranked compounds

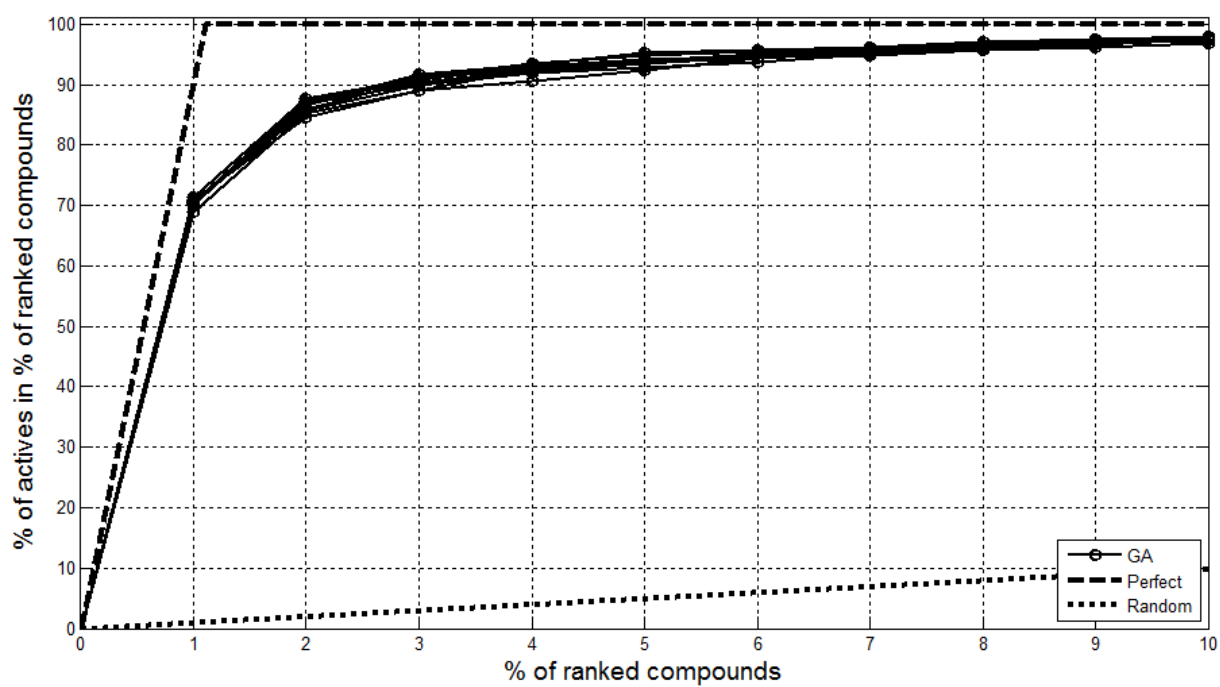


(a)

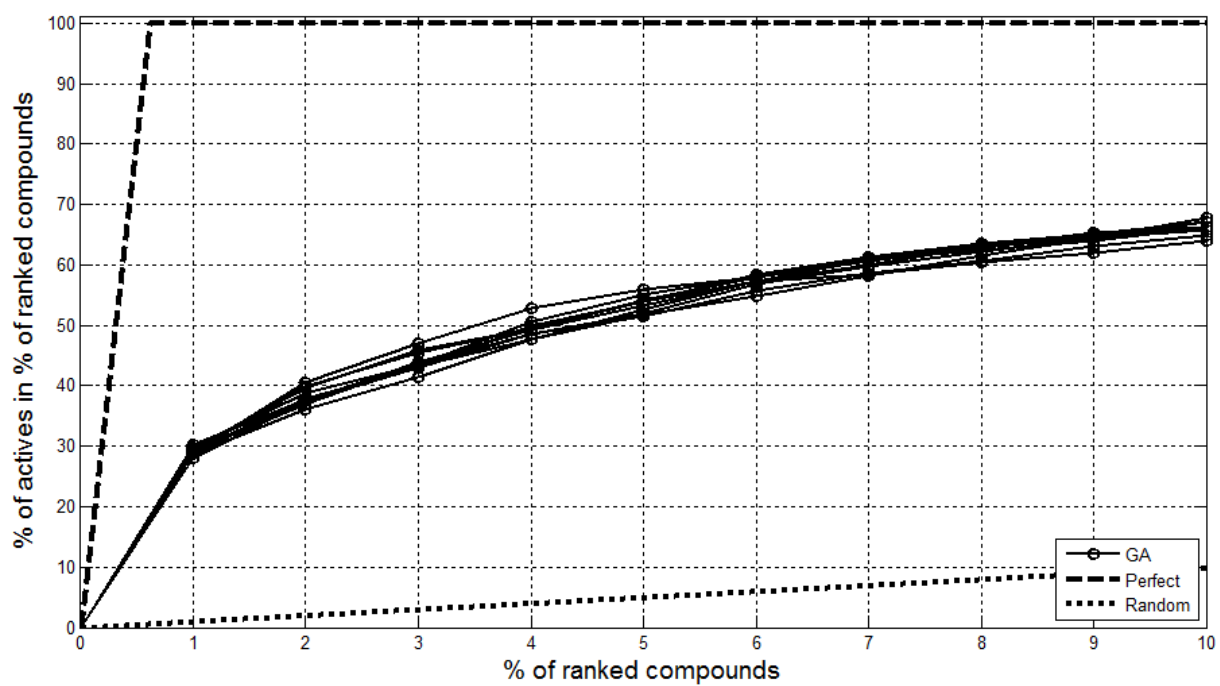


(b)

Figure 5.11: Error plot of training set versus predicted test set of the GA-based SSA following GA iterations for MDDR (a) RNN and (b) COX activity classes. Both GA instances were executed based on a chromosome population of 200 and maximum iteration of 500 to signify (i) Overfitting case, and (ii) Presence of improved recall rates in large iterations



(a)



(b)

Figure 5.12: The cumulative recall of active compounds plotted against the entire compound over 10 runs of the GA program: (a) GA instances for MDDR-based RNN activity class; (b) GA instances for MDDR-based COX activity class

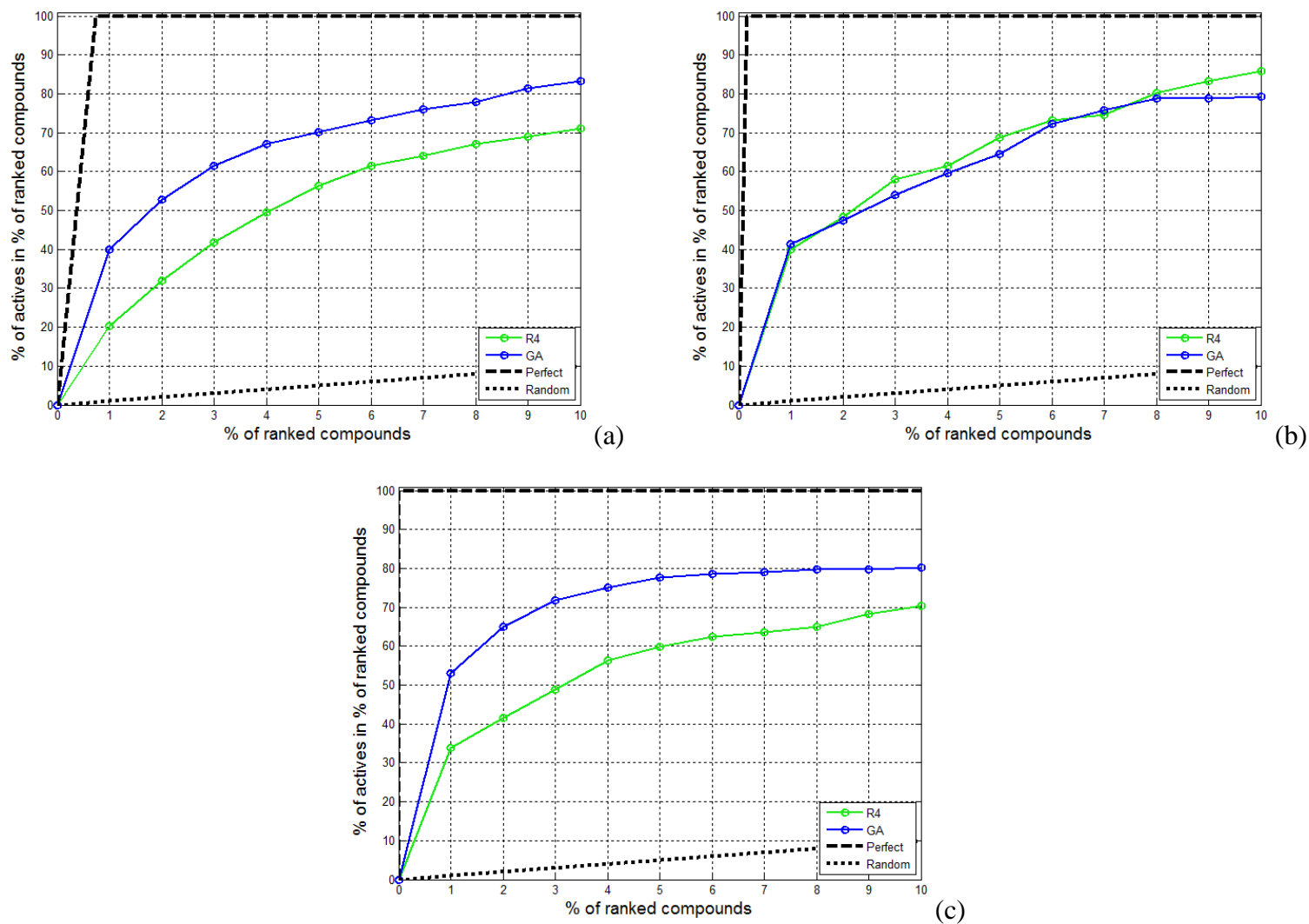
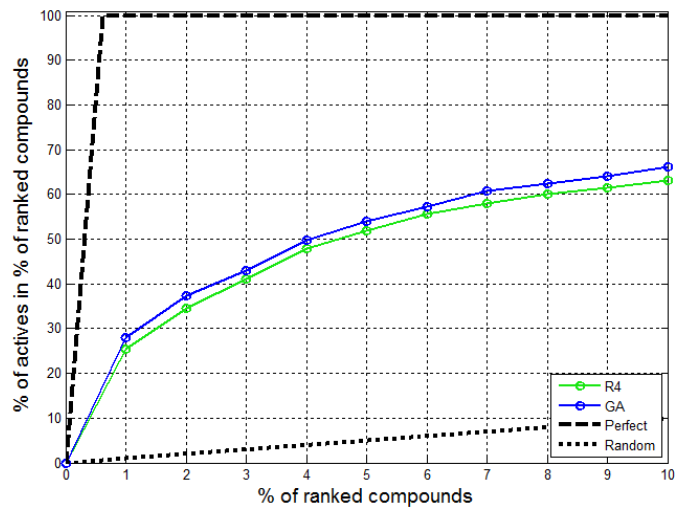
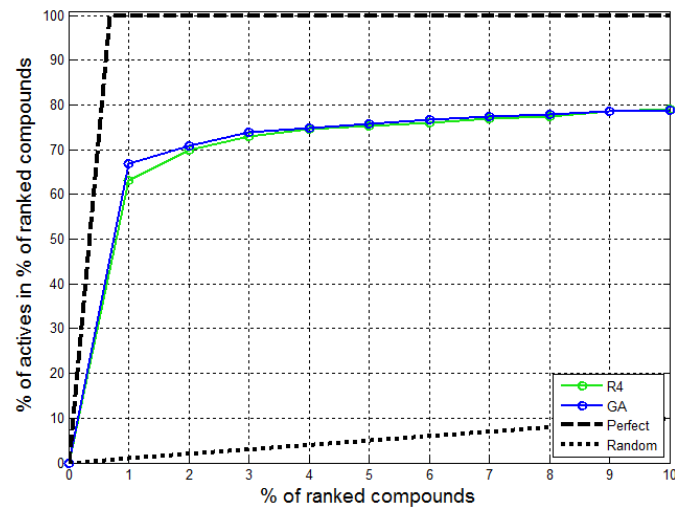


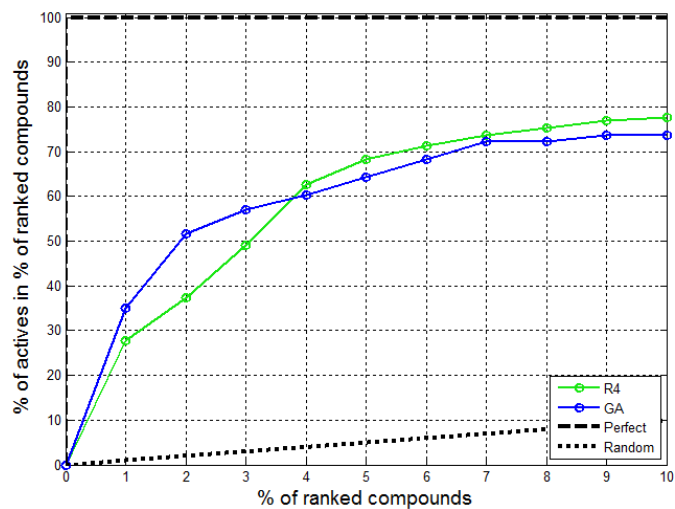
Figure 5.13: Cumulative recall plots of the GA-based SSA against SSA R4 for the 5HT3 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

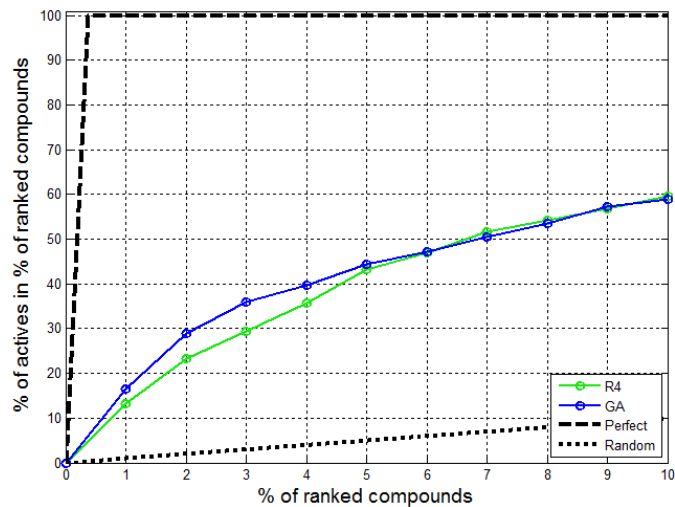


(b)

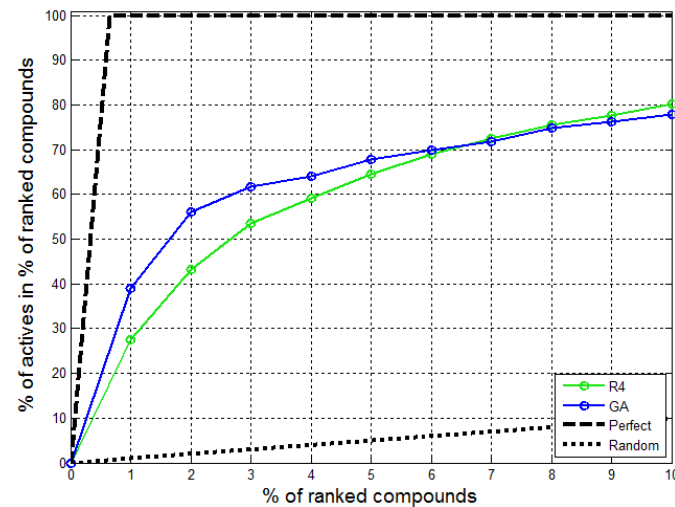


(c)

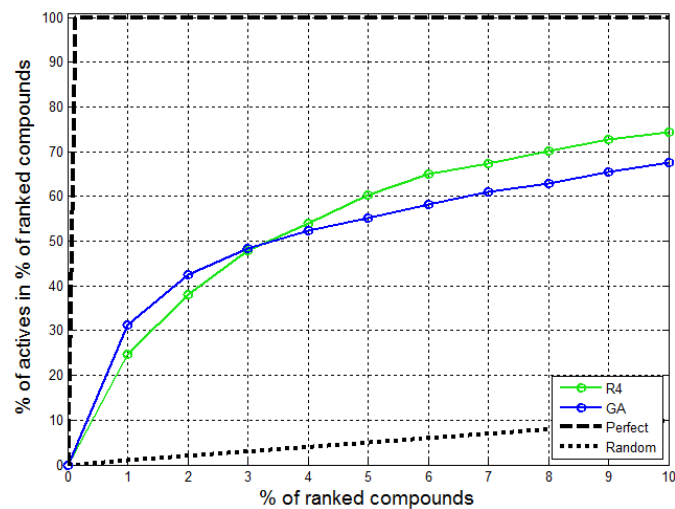
Figure 5.14: Cumulative recall plots of the GA-based SSA against SSA R4 for the COX activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

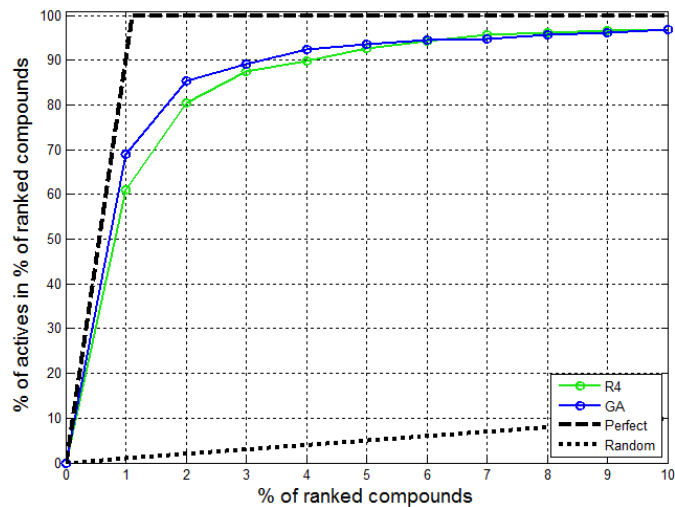


(b)

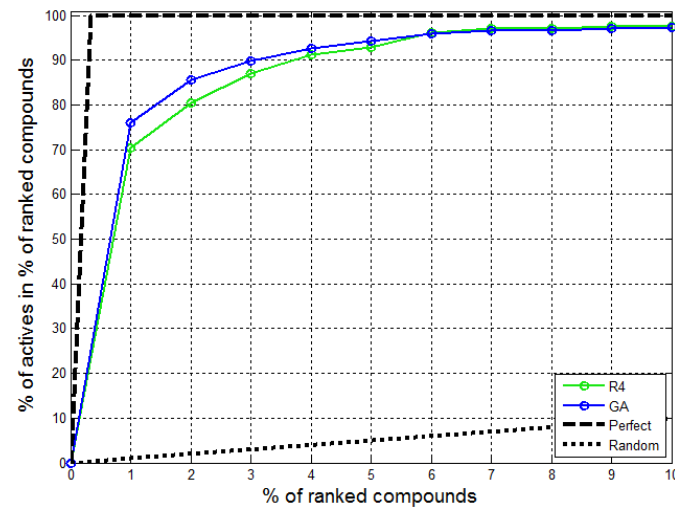


(c)

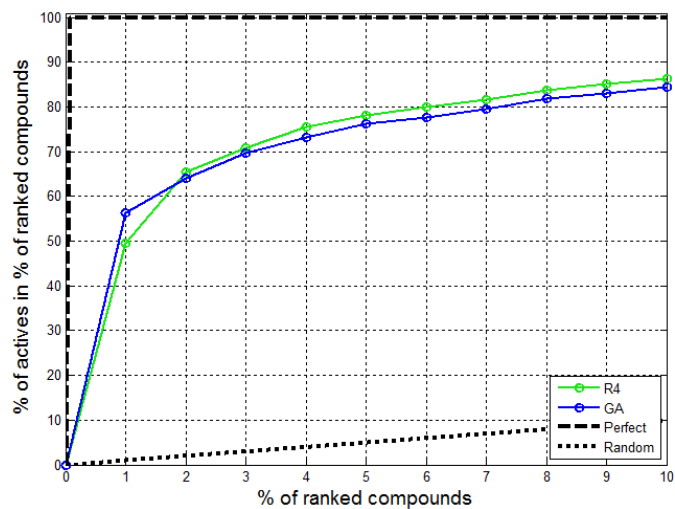
Figure 5.15: Cumulative recall plots of the GA-based SSA against SSA R4 for the D2 activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

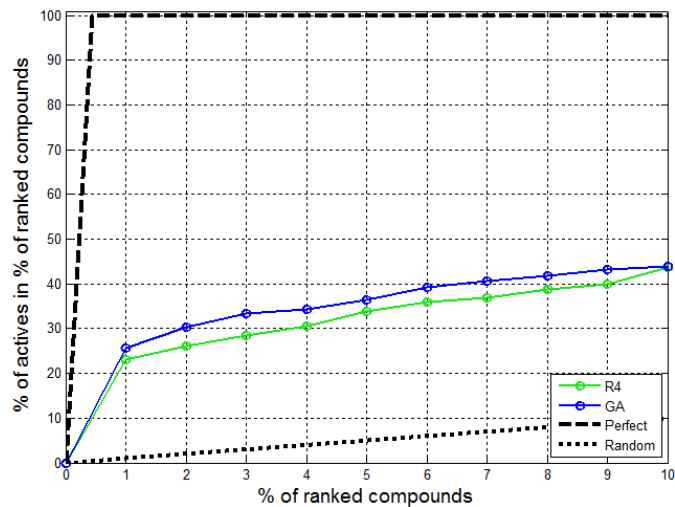


(b)

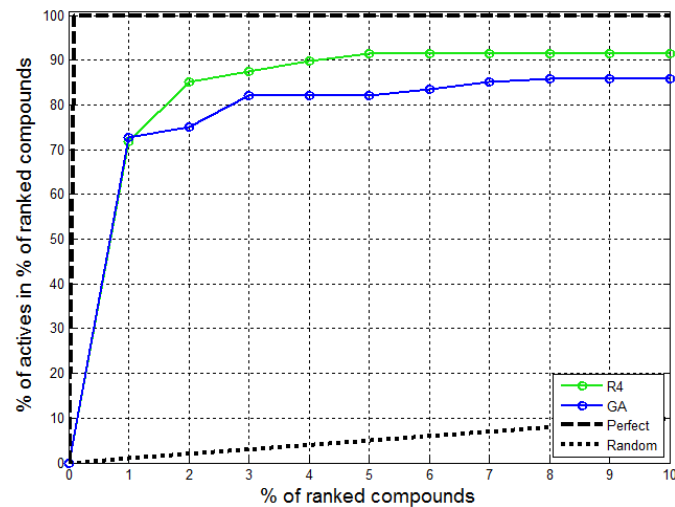


(c)

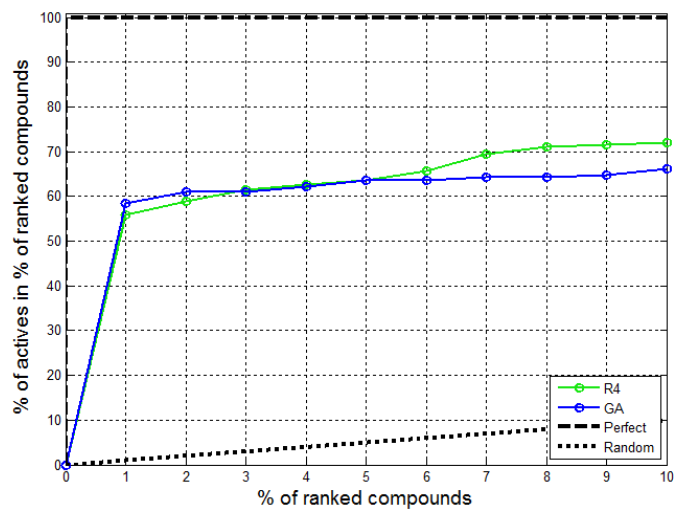
Figure 5.16: Cumulative recall plots of the GA-based SSA against SSA R4 for the RNN activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

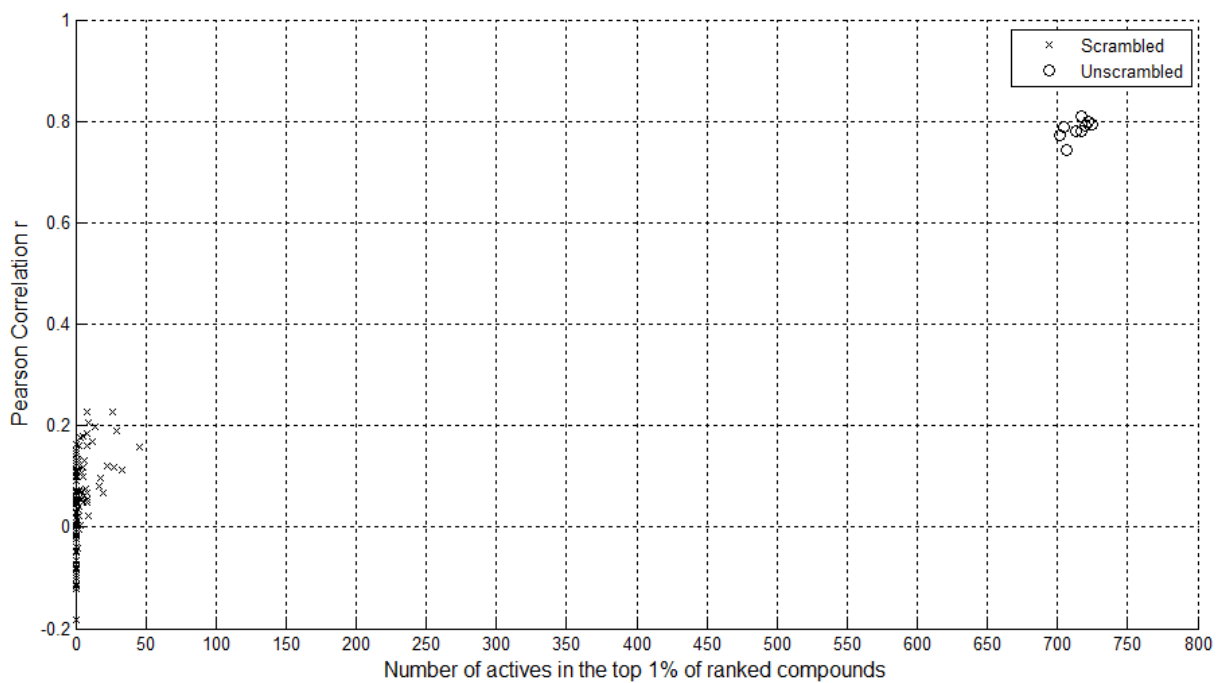


(b)

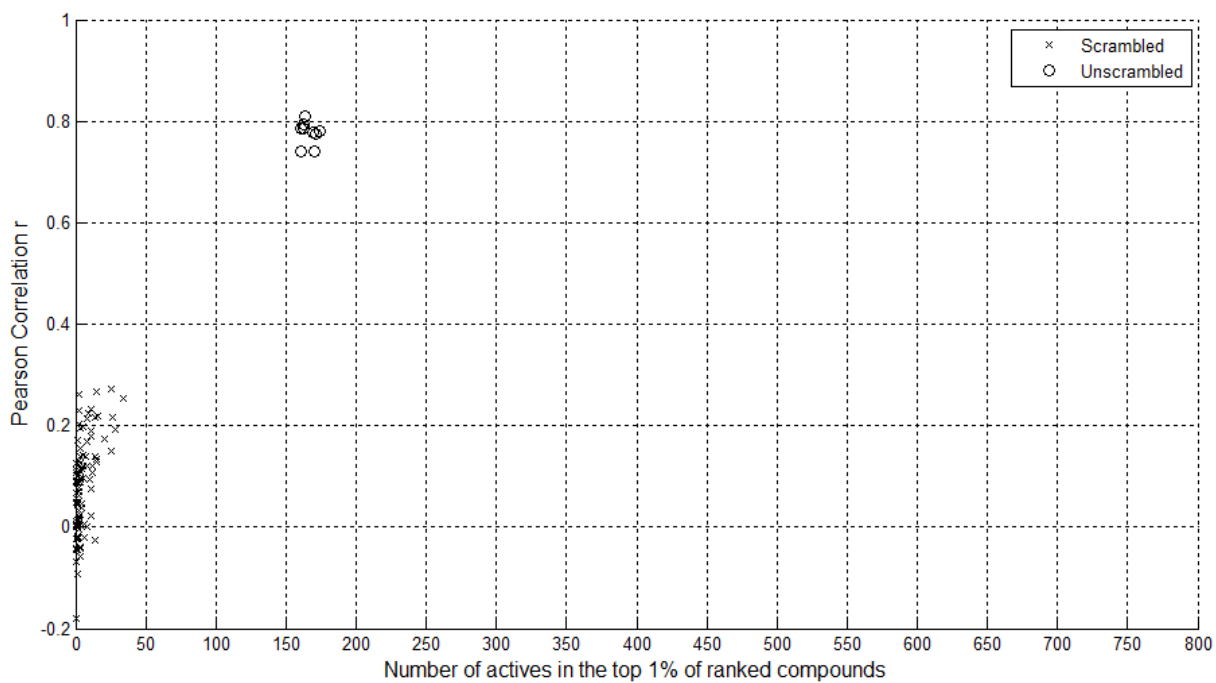


(c)

Figure 5.17: Cumulative recall plots of the GA-based SSA against SSA R4 for the PKC activity class from the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)



(b)

Figure 5.18: Permutation plots (Y-randomisation) of the MDDR-based (a) RNN and (b) COX classes, with weights calculated and applied to non-permuted test sets

Table 5.1: Example of a GA operation based on a population containing five molecules, with six chromosomes created at initialisation. (a) Fingerprints for five molecules M1-5 encoding five different substructural fragments F1-5; (b) The molecule activity state, 1 referring to an active molecule, while 0 represents inactive ones. (c) Six chromosomes C_{1-6} encoding the weights W_{1-5} for F_{1-5} ; and (d) Sums-of-weights using each chromosome C_{1-6} for each molecule M_{1-5}

Molecule	F ₁	F ₂	F ₃	F ₄	F ₅
M ₁	0	1	0	1	1
M ₂	1	0	1	1	0
M ₃	1	0	0	1	1
M ₄	1	0	1	0	0
M ₅	0	1	1	0	1

(a)

Molecule	Activity State (1-active; 0-inactive)
M ₁	1
M ₂	0
M ₃	1
M ₄	0
M ₅	1

(b)

Chromosome	W ₁	W ₂	W ₃	W ₄	W ₅
C ₁	6	2	5	0	1
C ₂	4	3	1	8	5
C ₃	9	9	3	6	7
C ₄	1	7	5	1	3
C ₅	8	4	8	2	8
C ₆	5	8	4	7	2

(c)

Chromosome	M ₁	M ₂	M ₃	M ₄	M ₅
C ₁	3	11	7	11	8
C ₂	16	13	17	5	9
C ₃	22	18	22	12	19
C ₄	11	7	5	6	15
C ₅	14	18	18	16	20
C ₆	17	16	14	9	14

(d)

Table 5.2: Weight polarity determination using the SSA R4 weighting scheme, following the example molecule and activity dataset in Table 5.1. (a) Summary of a five-fragment dictionary based on the common properties in SSA weighting schemes. (b) Weight polarity for fragments is determined on the basis of greater value between the rate-of-actives (*ROA*) against the rate-of-inactives (*ROI*). (c) The equivalent weight values and its polarity calculated using the SSA R4 weighting scheme

	F₁	F₂	F₃	F₄	F₅
<i>NACT</i>	3	3	3	3	3
<i>NINACT</i>	2	2	2	2	2
<i>ACT(I)</i>	1	2	0	2	2
<i>INACT(I)</i>	2	0	2	1	1

(a)

<i>ROA</i>	0.333	0.667	0	0.667	0.667
<i>ROI</i>	1	0	1	0.5	0.5
Simple Weight Polarity (I)	negative	positive	negative	positive	positive

(b)

SSA R4	-0.48	0.48	0.70	0.2	0.2
SSA R4 Weight Polarity (I)	negative	positive	negative	positive	positive

(c)

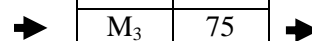
Table 5.3: Fitness score calculation using chromosome C_3 with its weights restricted by the weight polarity. (a) Chromosome C_1 weights combination and the corresponding weight polarity. (b) Assignment of chromosome weight to each fragment in the molecule set. (c) Sum of fragments' score of each molecule. (d) Ranking of chromosome based on molecule score in descending order from largest to smallest. All the active molecules are seen to benefit from SSA R4 weight polarity assignment based on their rankings at the top

	W_1	W_2	W_3	W_4	W_5
SSA R4 Weight Polarity	negative	positive	negative	positive	positive
C_3	-14	27	-88	23	66

(a)

M	FW₁	FW₂	FW₃	FW₄	FW₅
M_1	0	27	0	23	66
M_2	-14	0	-88	23	0
M_3	-14	0	0	23	66
M_4	-14	0	-88	0	66
M_5	0	27	0	0	66

(b)



M	SCR
M_1	116
M_2	-79
M_3	75
M_4	-36
M_5	93

(c)



M	SCR	ACT(I)
M_1	116	1
M_5	93	1
M_3	75	1
M_4	-36	0
M_2	-79	0

(d)

Table 5.4: Top 1% active retrieval rates for the fitness function parameter test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters.

Fitness function	Enrichment factor of actives in the top 1%	
	Test Set	
Fitness function active rate	RNN	COX
Active rate of compounds in the top 1%	70.10	30.37
Active rate of compounds in the top 10%	55.95	22.69

Table 5.5: Top 1% active retrieval rates for the GA weight range parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters.

Chromosome initialisation	Enrichment factor of actives in the top 1%	
	Test Set	
GA Weight value range	RNN	COX
0 to 10	21.10	0.00
0 to 100	29.50	0.50
-10 to 10	67.80	28.30
-100 to 100	71.20	30.14
-150 to 150	71.10	29.90
-200 to 200	70.18	30.10

Table 5.6: Top 1% active retrieval rates for the GA Population and Generation parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes.

GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters

Population and Generation	Enrichment factor of actives in the top 1%	
	Test Set	
Population Size	RNN	COX
100	69.60	30.90
200	70.88	30.12
300	70.67	30.11
400	70.78	30.12
500	70.88	30.12
Iteration	RNN	COX
100	68.93	28.15
200	71.04	30.27
300	71.04	30.27
500	71.04	30.27

Table 5.7: Top 1% active retrieval rates for the elitism model parameter. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters

Elitism model	Enrichment factor of actives in the top 1%	
	Test Set	
Elite chromosome preservation	RNN	COX
0	61.40	29.80
1	70.67	30.11
2	70.88	30.12
5	70.12	30.04
10	69.70	30.10
50	68.93	28.15
100	57.30	22.30
150	55.40	24.70
199	29.50	00.50
199 (using 1000 GA maximum iterations)	29.10	0.70

Table 5.8: Top 1% active retrieval rates for the evolution control parameter group. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. The worst enrichment values of 3 GA runs are listed below. Each parameter was executed three times and the worst result is selected to represent the individual parameters

Evolution control	Enrichment factor of actives in the top 1%	
	Test Set	
Parent selection method	RNN	COX
Roulette wheel	70.10	30.30
Tournament	69.70	30.10
Random	65.80	23.50
Crossover rate	RNN	COX
0.60	67.90	24.80
0.65	68.80	29.80
0.70	67.00	28.80
0.75	66.20	28.10
0.80	67.90	28.30
0.85	67.50	25.10
0.90	67.60	30.40
0.95	70.50	31.70
Mutation rate	RNN	COX
0.005	63.79	28.50
0.003	65.80	27.10
0.050	64.70	21.60
0.020	65.10	26.50
0.010	70.10	31.80
0.100	70.00	30.67
Crossover Method	RNN	COX
One-point	70.60	29.80
Two-point	65.50	22.10
Uniform	69.80	26.00

Table 5.9: The top-ranked molecules in ten GA runs based on test set applied data, showing the occurrences of ranked compounds based on GA run-1 that fall outside the top 1% in the other nine remaining GA runs using the (a) RNN and (b) COX activity classes in the MDDR dataset. The numbers in brackets show the number of actives actually retrieved in the top 1% for that particular GA-run

(a)										
Rank	GA Run									
	1	2	3	4	5	6	7	8	9	10
Top 1% (922)		50	48	37	38	61	49	63	47	51
	(701)	(713)	(720)	(725)	(717)	(706)	(705)	(702)	(722)	(717)

(b)										
Rank	GA Run									
	1	2	3	4	5	6	7	8	9	10
Top 1% (922)		13	15	12	21	11	22	20	15	15
	(167)	(163)	(161)	(174)	(171)	(170)	(164)	(161)	(163)	(169)

Table 5.10: Enrichment curve of actives count in the top 1% for ten GA runs of (a) Eleven activity classes in MDDR dataset; (b) Fourteen activity classes in WOMBAT dataset and; (b) Fifteen activity classes in ChEMBL dataset. Included are the mean and standard deviation for the Pearson correlation coefficients between the sets of 166 weights computed for each distinct pair of runs

Activity Class	GA Runs												Weight Pearson's r	
	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	σ	σ	Mean
	(a)													
5HT3	42.25	40.77	42.39	42.10	40.47	40.47	40.92	40.03	40.32	42.10	41.18	0.92	0.02	0.79
5HT1A	21.10	18.82	20.16	18.55	18.68	19.49	18.55	18.95	19.09	18.82	19.22	0.82	0.01	0.75
5HT	18.27	17.34	17.34	18.89	16.72	16.72	18.58	18.27	17.65	19.20	17.89	0.87	0.01	0.76
D2	17.98	16.85	17.98	17.13	16.57	19.66	17.13	16.57	16.85	17.13	17.39	0.94	0.02	0.77
RNN	71.29	68.93	70.80	70.11	70.40	70.50	70.50	70.50	70.99	70.21	70.42	0.63	0.02	0.75
AT1	48.65	47.59	47.94	47.70	47.47	47.23	47.70	48.53	48.29	48.53	47.96	0.50	0.02	0.77
THRM	49.24	47.58	50.21	47.30	48.41	48.27	48.82	47.30	47.30	48.82	48.33	0.98	0.02	0.80
SUBP	28.99	30.60	30.42	27.56	29.62	29.62	28.99	29.88	28.37	27.56	29.16	1.07	0.02	0.79
HIVP	48.89	48.30	49.48	42.07	48.30	47.56	48.44	44.89	47.11	45.33	47.04	2.28	0.02	0.79
COX	28.15	30.42	29.20	29.55	29.90	29.72	28.67	28.85	29.37	29.02	29.28	0.66	0.02	0.79
PKC	30.15	31.62	30.64	28.43	27.94	26.96	30.39	29.66	25.74	28.68	29.02	1.81	0.02	0.78

Activity Class	GA Runs											Weight Pearson's r		
	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	σ	σ	Mean
	(b)													
5HT1A	56.85	57.22	53.47	55.35	54.03	58.16	53.28	56.29	55.53	56.29	55.65	1.64	0.01	0.78
5HT3	43.43	43.43	41.41	42.42	49.49	42.93	41.41	42.93	42.93	44.44	43.48	2.30	0.02	0.76
ACHE	50.99	51.66	50.77	50.11	50.77	52.98	49.23	50.77	50.99	52.98	51.13	1.16	0.02	0.77
AT1	80.67	80.98	80.67	79.91	83.28	79.45	79.91	79.6	81.44	80.98	80.69	1.12	0.02	0.78
COX	67.66	68.01	67.09	66.86	66.86	67.55	66.97	67.55	68.01	67.66	67.42	0.45	0.02	0.77
D2	41.76	42.74	40.42	40.9	38.95	41.27	41.39	40.29	39.32	40.42	40.75	1.13	0.02	0.78
FXA	49.74	46.7	43.93	43.93	44.46	46.97	44.33	44.59	46.97	45.12	45.67	1.88	0.02	0.78
HIVP	55.37	57.34	49.56	56.45	51.43	52.02	54.78	51.53	52.61	54.48	53.56	2.50	0.02	0.81
MMP	63.84	62.56	64.00	62.72	63.52	63.36	62.4	63.68	64.00	64.00	63.41	0.63	0.02	0.77
PDE	48.69	50.75	48.32	48.69	48.32	48.88	48.88	51.68	48.51	50.00	49.27	1.15	0.02	0.77
PKC	74.22	77.34	73.44	75.00	73.44	72.66	72.66	73.44	73.44	74.22	73.99	1.38	0.02	0.75
RNN	80.80	77.52	82.67	80.33	78.69	77.28	79.86	77.52	76.11	76.81	78.76	2.09	0.02	0.78
SUBP	52.59	48.41	52.19	47.21	49.80	49.80	47.81	48.01	48.21	48.21	49.22	1.86	0.02	0.79
THRM	58.84	58.58	55.94	54.88	54.35	57.78	54.88	55.67	54.88	55.41	56.12	1.66	0.02	0.77
	(c)													
5HT1A	39.78	40.37	39.10	40.37	39.33	39.78	39.33	39.10	39.33	40.75	39.72	0.59	0.02	0.80
5HT3	55.73	57.81	53.65	53.13	57.81	53.65	54.17	54.17	53.13	57.29	55.05	1.93	0.02	0.78
5HT	34.88	34.33	34.33	33.33	35.51	35.15	32.11	35.51	34.88	35.15	34.52	1.07	0.02	0.78
ACHE	36.09	37.44	36.24	36.39	36.09	36.99	37.44	36.84	37.14	36.99	36.77	0.53	0.01	0.79
AT1	80.00	84.21	83.16	83.16	83.16	77.89	77.89	81.05	82.11	81.05	81.37	2.22	0.02	0.75
COX	41.60	41.60	40.00	41.60	36.00	35.20	36.00	40.00	35.20	36.00	38.32	2.86	0.03	0.78
D2	31.46	31.70	31.34	33.37	31.22	31.46	31.70	31.22	33.37	31.34	31.82	0.84	0.03	0.83
FXA	46.89	48.37	47.34	47.12	47.26	47.26	47.34	48.37	46.89	47.12	47.40	0.54	0.03	0.83
HIVP	63.83	64.97	63.83	68.21	64.97	64.40	64.76	64.76	64.76	64.04	64.85	1.26	0.04	0.86
MMP	71.35	67.42	65.73	67.42	67.42	71.35	67.13	71.35	67.42	71.35	68.79	2.26	0.04	0.86
PDE	38.86	37.55	38.86	38.86	42.79	42.79	43.67	37.12	38.86	42.79	40.22	2.49	0.04	0.79
PKC	59.47	58.42	57.37	58.95	58.95	58.42	59.47	59.47	57.37	58.42	58.63	0.79	0.04	0.74
RNN	56.56	57.47	57.47	56.45	56.45	56.45	56.45	56.45	56.56	56.11	56.64	0.45	0.04	0.80
SUBP	68.50	70.08	70.08	71.65	70.08	71.65	68.50	70.08	70.08	70.08	70.08	1.05	0.04	0.81
THRM	48.01	47.88	44.96	44.96	45.76	47.88	46.29	47.88	48.01	46.02	46.76	1.30	0.04	0.80

Table 5.11: Screening results using the GA-based SSA and its comparison to the SSA R4 weighting scheme for the (a) MDDR; (b) WOMBAT and (c) ChEMBL datasets. The number of actives retrieved at the top 1% based on the worst performing GA runs is recorded for the calculation of Tanimoto coefficient and the *BemisMurckoAssemblies* based diversity analysis

Activity class	Actives test set	NBC		Worst run		Murcko scaffolds		Diversity rate	
		SSA R4	GA	SSA R4	GA	SSA R4	GA		
(a)									
5HT3	677	138	271	75	83	0.38	0.39		
5HT1A	744	103	138	52	63	0.36	0.36		
5HT	323	34	54	17	20	0.32	0.38		
D2	356	47	59	24	30	0.29	0.31		
RNN	1017	620	701	192	205	0.28	0.29		
AT1	849	372	401	131	139	0.31	0.33		
THRM	723	226	342	99	151	0.25	0.38		
SUBP	1121	262	309	120	129	0.34	0.36		
HIVP	675	226	284	106	115	0.33	0.40		
COX	572	146	161	36	37	0.46	0.46		
PKC	408	94	105	34	38	0.39	0.40		

Activity class	Actives test set	NBC		Worst run		Murcko scaffolds		Diversity rate	
		SSA R4	GA	SSA R4	GA	SSA R4	GA		
(b)									
5HT1A	533	249	284	58	64	0.34	0.35		
5HT3	198	79	82	23	25	0.33	0.40		
ACHE	453	218	223	67	69	0.30	0.31		
AT1	652	514	518	105	106	0.37	0.37		
COX	869	549	581	32	39	0.36	0.36		
D2	819	225	319	61	65	0.32	0.34		
FXA	758	288	333	75	86	0.31	0.45		
HIVP	1015	398	503	127	143	0.35	0.37		
MMP	625	362	390	86	97	0.23	0.24		
PDE	536	239	259	84	86	0.48	0.58		
PKC	128	92	93	15	15	0.40	0.46		
RNN	427	301	325	73	76	0.32	0.34		
SUBP	502	217	237	52	54	0.31	0.38		
THRM	379	200	206	73	77	0.41	0.43		
(c)									
5HT1A	1335	383	522	114	141	0.36	0.38		
5HT3	192	65	102	14	25	0.34	0.38		
5HT	2202	540	707	107	117	0.51	0.52		
ACHE	665	168	240	83	99	0.40	0.42		
AT1	95	40	74	8	22	0.28	0.28		
COX	125	35	44	9	12	0.73	0.75		
D2	1672	413	522	120	154	0.31	0.31		
FXA	1352	467	634	141	163	0.37	0.37		
HIVP	1941	903	1239	264	324	0.42	0.51		
MMP	356	208	234	57	62	0.44	0.45		
PDE	229	63	85	19	25	0.54	0.60		
PKC	190	106	109	23	26	0.25	0.25		
RNN	884	438	496	68	69	0.36	0.38		
SUBP	762	476	522	104	105	0.37	0.43		
THRM	754	192	339	90	120	0.32	0.39		

Table 5.12: GA run-time benchmark at different iterations using the 10% training set of the RNN activity class, based on the (a) MDDR, (b) WOMBAT and (c) ChEMBL databases. Parameterisation of the GA is based on the final chosen ones as in Section 5.7.5, among them the population of 200 chromosomes, and a maximum iteration 200 evolutions

(a)

Machine	Time (seconds) at GA iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	2.00	217.25	468.51	2.10
WKST_01	2.80	288.42	601.74	2.99
WKST_02	3.00	331.16	660.38	3.30

Based on 10,254 compounds in training set

(b)

Machine	Time (seconds) at GA iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	2.60	284.50	551.23	2.77
WKST_01	3.50	381.20	759.22	3.77
WKST_02	4.00	458.20	911.95	4.10

Based on 13,812 compounds in training set

(c)

Machine	Time (seconds) at GA iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	34.20	3560.40	7011.20	35.05
WKST_01	<i>Insufficient memory issue, not executable</i>			
WKST_02	40.10	4215.20	8450.00	42.25

Based on 135,267 compounds in training set

Chapter 6

Genetic Programming Approach to Substructural Analysis

6.1 Introduction

Following the investigation into the use of GA for SSA in chapter 5 and its encouraging results with regard to an improvement in the retrieval of actives over SSA weighting schemes, this chapter investigates the applicability of another class of evolutionary algorithm, specifically the Genetic Programming (GP), tailored for SSA. Similar to the GA, the GP-based SSA required two experiments. The first was to identify and understand various parameterisation options existing in the GP, and finalise a suitable parameter set. The second experiment analysed GP-based SSA and quantified its performance in comparison to the reviewed SSA weighting schemes and GA-based SSA.

Genetic programming (GP) is a type of machine-learning based, problem-solving technique, derived from the domain of evolutionary algorithms which also include genetic algorithms (GA). Both GP and GA are considered to be stochastic and heuristic in their approach to problem solving. GP, however, differs in that it uses blocks of computer programs for its population, instead of the single representations of a bit string in GA. The GP's population is frequently presented in the shape of a tree-structured plan. Similar to GA, GP is capable of exploring the algorithmic search space and evolving computer programs to perform a defined task. Although GP can be traced back to the 1950s, it was not until the 1990s that John R Koza pioneered GP for the optimisation of real-world and complex problems (Koza, 1990).

6.2 Fundamental components of GP for SSA

Both GP and GA belong a class of evolutionary algorithms where the aim of the optimisation is to evolve a population of candidate solutions that, when evaluated, produce individuals as close as possible to the desired goal. The main difference between GP and GA, however, is in their representation of the candidate population. In the case of GP, population members are represented as computer programs or a complex algorithm made up of functions (primitive by default) and variables usually connected in the form of a tree. As described in the earlier Chapter 5, Section 5.2.1, GA defines its population as comprising of fixed-length binary

string or some other data structure values. In both cases, however, the population is still manipulated by the computer program to perform a series of evolutions mimicking the genetic lifecycle.

Figure 6.1 describes the basic functionality of GP. First, a population of random programs is generated. Each of the candidate programs runs through the evolutionary cycle, essentially via genetic operations such as crossover and mutation (much like the GA) and their solution capability, or fitness, is quantified. The repeated cycle should lead to the continuous breeding of fitter programs, which will ideally lead to the preferable solution. A distinct feature of GP is that it does not function to simply optimise parameters such as GA (and perhaps other machine learning methods). Instead, GP works by searching for an ideal sequence of functions and variables to form a computer program. In general terms, this should be understood as automated programming. GA strives only to solve a specific set of problems where the termination criteria are known (or expected). However, it will not go beyond that instance of the problem. GP has been claimed to be an effective general machine learning paradigm, which was proven based on successful application of GP to solve a wide range of problems (Koza, 1992).

Koza (1992) stated that the GP has five major preparatory steps that the user is required to specify. As chromosomes are based on blocks of computer programs, the user first needs to identify suitable variables and operands categorised as individual nodes based on the following:

- (i) Terminal sets consisting of individual, independent variables.
- (ii) Operator functions that essentially represent simple operation(s) which connect the terminal recursively.
- (iii) Fitness measurement of the candidate solution's accuracy.
- (iv) Choice of evolution and generational methods.
- (v) Termination criterion, which signals to the GP to terminate if a criterion is satisfied.

The last three steps are similar to the requirements of the GA preparation. Table 6.1 summarises the main differences between GA and GP implementation as discussed above and based on the following criteria below.

6.2.1 Encoding of Chromosomes

Based on the terminal and function pools, the chromosomes in GP are set up as a connection of nodes resembling a tree with a singular parent node at the top. This follows the branches of child nodes known as sub-trees which are usually replaced, modified or mutated during the lifecycle of the GP in order to drive the evolution process. Figure 6.2 describes a simple tree model representation of a GP program candidate. The tree consists of nodes made up of individual variables or functions, which are connected in a chain from top to bottom. The tree starts out with one defined terminal at the top and is expanded downwards by various randomly generated function sets and child terminal sets. The tree in Figure 6.2(a) can be read mathematically as shown in Equation 6.1:

$$a / (b + (\cos(c))) \quad (\text{Equation 6.1})$$

This in itself is a valid equation that follows mathematical logic. There are, however, instances when tree modelling can become particularly complex and care should be taken to only generate valid trees. Figure 6.2(b) shows an example of an invalid tree, read as shown in Equation 6.2:

$$a / (b + (\times c)) \quad (\text{Equation 6.2})$$

The equation above can be translated back and forth into a GP tree. It is, however, impossible for such an equation to be valid simply because an operand is missing within the multiplication term at the end of the equation. This can be modified to include another operand i.e. a child node underneath the multiplication node. This requirement is also known as the arity criterion. Recalling the multiplication function, the arity value is thus defined as 2, while a primitive function like the cosine function in Figure 6.2(a) has an arity of 1, meaning that it requires only 1 node to be its child. A variable node does not yield a child, giving it an arity of 0 and this is where a sub-tree terminates for that branch. A sensible GP program should ensure that the arity criterion is enforced; this can be done by applying exception controls. For example, a program can check for errors when generating sub-nodes, such as forcing a parent mathematical node to produce two children when the arity of that node is equal to 2.

6.2.2 Population growth

Koza (1992) describes several ways of initialising a tree-based chromosome population. There are fundamentally two ways to construct the trees: the *grow* and the *full* methods. The *grow* method allows for the tree to grow its nodes from both primitive sets of terminals and functions, until it reaches the maximum allowable tree depth as defined by the user. The term “depth” here is defined as the depth from the root node (otherwise known as depth 1) to the lowest point of the traversed node, as shown in Figure 6.3. An example of the *grow* method is shown in Figure 6.3. A maximum tree depth is set during initialisation; in this case it is set at 3. The first iteration must have a function primitive selected as the parent node so that the tree can grow larger. Subsequent iterations, however, allow for primitives to be selected from both the terminals and functions pool. For the above figure, iteration 2 selected variable *a*, thus terminating any possibility of growth on the left side of the tree branch, even though the maximum tree depth has not yet been reached on this side. Continuing from iteration 3 to 5, the tree continues to grow on the right side until it reaches the maximum tree depth at 4. Note that the *grow* method also allows for the tree to be smaller than the maximum tree depth. Following again the example from Figure 6.3, if the tree were to choose another variable on the right side in iteration 3, this would mean that the tree would not grow any larger. It would be finalised with only 3 nodes, which is acceptable for the *grow* method.

In the *full* method, the tree is forced to grow until it reaches the maximum defined tree depth. This is illustrated in Figure 6.4. In this method, the first node of the tree can only be selected from function primitives. Subsequent child nodes can only select primitive functions (as selecting terminals will end the growth), until it reaches the maximum tree depth. When it reaches the maximum tree depth, it will then end the growth by selecting variables. Referring to the example from the figure, iteration 2 selects a function node. However, since the maximum depth is reached in iteration 4 at the left side of the tree and with a function of 2-arity, it is forced to complete this branch with 2 variables. Similarly, iteration 5 chooses a function, while iteration 6 completes the former with a variable.

Another popular variant of the population initialisation method combines both the *full* and *grow* methods. This method was also introduced by Koza and is known as the *ramped half-and-half* method. In this method, half of the population uses the *grow* method with a depth randomly defined from the maximum depth allowed, while the other half of the population is defined using the *full* method. This method ensures that the population of the GP is filled

with varying tree size to ensure a higher diversity of potential solution terms in the chromosomes.

6.2.3 Evolutionary operators

Genetic operations for the GP remain largely similar to those for GA, being mainly the crossover and mutation operations. The difference is largely in the implementation of the operations themselves. Figure 6.5 describes an example of a crossover operation involving two chromosome trees. From the figure, the two parent chromosomes are selected and the node depth identified (usually randomly) as a starting point for the crossover to take place. Nodes from that depth down to the lowest level of the parent chromosomes are removed and exchanged with each other. Koza (1992) also describes two types of mutation that are possible in a tree-like representation, as illustrated in Figure 6.6. The first type of mutation is also known as the single terminal mutation (Figure 6.6a). It replaces only a random single terminal from the tree with another valid terminal. For example, a mathematical function that undergoes this type of mutation can be replaced by any other function terminal that has the same criterion as the original terminal, such as its arity requirement, or some other prerequisite. Likewise, a variable terminal can choose any other variable, although it is not allowed to mutate into a function as this would change the whole tree branch structure below the mutate terminal. The second type of mutation is known as sub-tree mutation (Figure 6.6b). This allows the entire sub-tree of the randomly chosen terminal to be replaced by another randomly generated sub-tree.

6.3 Previous works in GP

There have been extensive reports on the success of applications of GP in various fields. McPhee, Poli and Langdon (2008) have compiled many GP-based works across different domains; this includes applications in financial trading (Chen & Liao, 2005; Chen, & Yeh, 2002; Samanidou, Zschischang, Stauffer & Lux, 2007), medicine, biology and bioinformatics (Handley, 1993; Koza & Andre, 1996).

In relation to chemoinformatics application, Nachbar (1998) demonstrated the use of GA and GP to manipulate molecule topology for chemical structure optimisation, using representations of tree-based data structures and sets of algorithms. Birchall (2005) studied high-throughput screening using reduced graph approaches, by utilising machine learning methods, which in his case was done via GP. It was argued that the predictive power of GP

may be able to assist with the enormous search space as represented by HTS data. Birchall implemented both single-objective and multi-objective GPs, which only accounted for small improvements in the predictive performance of the reduced graph, plus some other less notable benefits (Birchall, 2011).

Nicolotti et al. (2002) reported on the investigation of multi-objective method in quantitative structure-activity relationships using genetic programming. The objectives comprise of model fitting, the total number of terms and the occurrence of non-linear terms. The study reported that the multi-objective model was found to be at least as good as the model obtained from existing statistical methods. The model can also be used by chemists in interpreting the statistical robustness of chemicals.

6.4 Experimental details

The main objective of this experiment was to utilise GP to identify effective equations that can generate a set of suitable fragment weights for use in the ranking of SSA-based compounds. It was noted that the GA-based SSA application was found to be superior to traditional SSA methods. The main goal of the experiment was to explore any possibility of improving the SSA method when using the GP approach. The results gained using GP are evaluated against the GA-based SSA scheme and existing SSA weighting schemes (i.e. Robertson and Sparck-Jones's R4) for performance evaluation.

6.4.1 Dataset

The datasets used for the GP experiments were discussed in detail in Chapter 3, Section 3.2. They comprise eleven, fourteen and fifteen activity classes from the MDDR, WOMBAT and ChEMBL databases respectively (Table 3.1). Similarly, this study used the predictive sets of each activity class to represent the training set. They contained 10% active and 10% inactive molecules as the input dataset to be used in the GP. The remaining 90% of the data were classed as the test set, and used to evaluate the predictive performance of the training set. The GP was run on the training set in order to calculate the fitness of an individual during the search and learning process.

6.4.2 Hardware

Similar to the GA, similar hardware was used as listed in Table 3.2. The performance of the hardware when performing the GP was observed and is discussed in this chapter.

6.4.3 Algorithm Implementation

MATLAB remains the software of choice to program the main algorithms and the required functions for the GP, as most of the main genetic functions such as chromosome generation, population control, fitness initialisation and measurement and others were already in place. The main GP algorithm is similar to the GA algorithm (Algorithm 5.1). The exception is that the GP program handles a different chromosome population consisting of variable-length program blocks (further divided into terminal and function sets), instead of the fixed binary or string arrays as employed in the GA. Hence, the genetic reproduction operations of the GP differ slightly compared with the GA in its reproduction and mating procedures. Appendix B lists the full pseudo code of the GP-based SSA program written in MATLAB.

6.4.3.1 Chromosomes population and generation

Chromosomes in the GP program are represented as program blocks made up of terminal (or variables) and function sets. The combination of both terminals and functions are what allows an equation to be formed, in which the most suitable will be determined through the course of genetic evolution. Here, the implementation of the GP in the case of biological activity prediction via the SSA method is essentially based on the assignment of SSA-based variables and simple mathematical operations to yield a set of fragment weights.

A series of variables and primitive functions were investigated, as summarised in Table 6.2. The terminal sets were identified based on the following variables used in most SSA equations (listed as VARIABLES_A combination set in the table): (i) N or the total number of compounds, (ii) $NACT$ or the number of active compounds, (iii) $NINACT$ or the number of inactive compounds, (iv) $TOT(I)$ or the total number of compounds containing fragment I, (v) $ACT(I)$ or the total number of active compounds containing fragment I, (vi) $INACT(I)$ or the total number of inactive compounds containing fragment I. All of these variables were discussed in detail in Chapter 2, Section 2.9.3. In conjunction with this, a number of additions were also proposed to the basic terminal set, which consisted of smaller terms used in various SSA weighting schemes. These are listed in Table 6.2, defined as the VARIABLES_B terminals set. For example, the SSA R1 equation is defined as “ $(ACT(I) / NACT) / (TOT(I) / N)$ ”. Hence it is possible to divide the equation into two smaller terms: (i) “ $ACT(I) / NACT$ ”, and (ii) “ $TOT(I) / N$ ”. The argument for using these smaller terms as terminals is that the equation terms presented in the various SSA equations are equation components that are more appropriate. It is predicted that the equations may be able to converge faster when a set

of predefined terminals consisting of logical terms is available from the start of the GP program. The objective was also to reduce the generation of extreme terms in an equation which may pose the likelihood of overfitting problems at a later stage. The effects of the definition of both terminal sets is investigated and discussed in the GP parameterisation experiment.

For the chromosome function set, four fundamental mathematical expressions were used by default to be experimented. These consisted of (i) the *plus* operation, (ii) the *minus* operation, (iii) the *multiplication* operation, and (iv) the *division* operation. Several other complex operations were investigated. These consisted of (v) the *logarithmic* function or log, (vi) the *exponent 2* function, and (vii) the *square root* operation. These functions are listed in Table 6.2(b) and grouped as the FUNCTIONS_A function set. The modulus operation was excluded as it is a rather complex operator to be used in designing an SSA-based mathematical equation. The limitation of the power function to only the power of 2 was to minimise the weight calculation complexity. Having a weight calculated based on a variable with a power of n, however, would almost certainly cause the fragment values to be larger than necessary. Likewise, for the function set combinations, a special function known as the FUNCTIONS_B function set was included. This set consisted of a log operation enforced to each GP equation. It is argued that the equation generated by the GP may sometimes unnecessarily enforce several conditions to be excessively enlarged. To illustrate this, the Figure 6.7 is referred. An equation is given and translated to its equivalent fragment weights. Note that for fragments 137, 138, 146, 148 and 150, the weighting values were determined to be excessively large values due to the presence of two large variables ($TOT(I)$) being multiplied by one another unchecked. There may be another possibility of such an occurrence; for example, a variable consisting of large values further encapsulated by multitudes of exponent 2 operations. To solve this problem, it was necessary to enforce mandatory implementation of the log function at the end of every generated GP equation, as can be seen in Figure 6.8. Here, the same equation was encapsulated with a log function. The said fragments containing the excessively large values were then stabilised to sensible numbers in a small range. This effect is also demonstrated in terms of the performance analysis in the GP experimentation section.

6.4.3.2 Suitable fitness function design for GP-based SSA

Fitness determination for the GP is similar to that used in the GA-based SSA. The fitness of one chromosome is measured as the active retrieval rate resulting from the fragment weights generated by the chromosome. Specifically, a chromosome yields an equation, which, when computed, produces a specific set of fragment weights. The fragment weights are applied to each compound and the sum of weights act as the compound's score. The compounds are then categorised based on the score.

Figure 6.9 illustrates a simplified example of a GP lifecycle from population definition to fitness evaluation. The GP program starts by identifying the parameters to be translated into chromosomes for manipulation by GP. In the case of GP-based SSA, the parameters are derived from variables; these describe the characteristics of the 2D fingerprint dataset, as shown in Figure 6.9(a). For the GP-based SSA, the main objective is to generate an equation that is able to maximise the active retrieval rate. Thus, a chromosome can be defined from a pool of variables containing the SSA variables. A function set therefore consists of mathematical operations to connect the variables. Figure 6.9(b) illustrates several examples of a series of randomly generated chromosome-based equations. Taking an example of such an equation as shown in Figure 6.9(c), the fitness value of this equation can be determined by firstly assigning the equation to be translated to fragment weights. These are then applied to the 2D fingerprint datasets. Subsequently, a compound's score is calculated as the sum of all of its fragments' scores. The compounds are then ranked based on their scores, in descending order, as shown in Figure 6.9(d). Finally, a fitness value is computed as the number of active compounds found in the top 1% list of ranked compounds. These steps are then repeated for all of the other chromosomes to generate a fitness table, as shown in Figure 6.9(e).

6.5 Experimental procedure

The experiments were divided into two parts. The first experiment was conducted to identify the best set of parameters from a varying number of parameter options. It was necessary to first identify the possible sets of values for each specific parameter under several different parameter groups. The following parameter groups were identified: (i) Terminal and function set for chromosome generation, (ii) Population size and maximum evolution, (iii) Elitism mode, (iv) Bloat control and (v) Evolution control. The parameters were tested by changing the values one at a time while retaining the other parameters at the determined default value to observe the impact of changing one parameter. The second part of the experiment focused

on analysing the actual performance of the GP-based SSA compared to the traditional SSA methods and the GA-based SSA. For benchmarking purposes, the best SSA method was used: the Robertson / Sparck-Jones R4 method. This was employed to represent the SSA based approach to be compared against the GP-based SSA program. All of the GP experiments above required an initial run on a training set, followed by its predictive verification on the remaining test sets. The process was repeated three times for each parameter value in the parameterisation experiments, while the benchmarking experiment of each activity class was repeated ten times.

6.6 Experiment setup: Parameterisation of the GP-based SSA

The first part of the experiment was performed to identify suitable parameters of the GP to be applied for all runs of the different activity classes from both databases. The majority of the parameter groups were similar to the GA experiment, and thus they were tested here as well as other parameters unique to the GP, which are discussed below.

Similar to the GA experiment performed in the previous chapter, each parameter was tested individually to observe potential changes in that particular parameter. The same procedure was used as in the previous SSA and GA experiments, whereby the least and most heterogeneous activity classes from the MDDR database, COX and RNN respectively, were tested. A training dataset of 10% active and inactive compounds was used to test each parameter. The subsequent GP-produced equation was used to generate a set of fragment weights calculated from the test dataset. The weights were then applied to the predicted test set and its retrieval performance was observed. For the parameterisation experiments, each GP run of a parameter value was run and repeated three times and their active retrieval rates were observed. It is argued that the run with the worst retrieval rate of the three instances should suitably represent the parameter being tested as it seeks to present a stable score from the three runs. Once all of the parameter values had been assigned their parameter score, it was then necessary to choose the parameter value with the highest score from all of the available parameter options being investigated. The best performing option is shaded in the parameter tables accompanying the tests below.

To ensure that results obtained from the parameterisation test are consistent and noise-free, each GP parameter test is repeated three times to note any large occurring discrepancies. The worst performing result out of the three was selected in order to effectively represent the

results of each parameter tested. The predictive sets of the two activity classes from the MDDR database, namely the RNN and COX classes, were chosen as input datasets for the parameterisation experiments for the same reason as in the GA experiments. Based on the GP program developed, and following various literatures on GP parameterisations, the experimenter identified a key number of parameters required by the GP thus: (a) fitness function at a selected percentile; (b) chromosome population size; (c) maximum generation / evolution limits; (d) elite chromosomes, (e) crossover rate; (f) mutation rate; and (g) parent selection method for offspring generation.

Each parameter is changed one at a time to systematically record the performance variation of parameters. This ensures that the effect of individual parameter variation is quantified as accurately as possible, and noise results are not mistakenly recorded. A set of initial, default parameter values were first defined. They consisted of a fitness function score based on ranked active compounds, the top one percent, a 100 chromosome population size, weights determined by formula derived from FUNCTIONS_A and VARIABLES_A set, 300 maximum iterations, a roulette wheel parent selection, elitism of 1 chromosome preservation between evolutions (mimicking simple-state model), a one-point crossover method of 0.95 probability rate and a mutation rate of 0.05, as the default GA parameters. Individual parameters being investigated were changed while the other parameter set mentioned above remained intact. The parameters were performed and the results obtained, are discussed below. To distinguish the most effective parameter, the highest values shown in Table 6.3 to 6.8 are highlighted.

6.6.1 Fitness function

Similar to the fitness function parameter test conducted in Section 5.6.1, the fitness function definition is experimented for the GP. Two sets of parameter options were tested: The fitness function score based on the top 1% ranking, and another fitness function based on the top 10% ranking.

Figures 6.10 and 6.11 show cumulative recall plots of the different fitness functions used in the GP for both the MDDR RNN and COX activity class respectively. For the MDDR RNN class, the fitness function score based on the top 1% (Figure 6.10a) recorded improvement in actives retrieval rate over the SSA R4. It is also stable in other ranked percentile, except for a drop particularly at the top 3% ranking. On the fitness function based on the top 10% ranking

(Figure 6.10b) however, the actives retrieval rate is seen to severely struggle against the SSA R4. It only achieved improved actives retrieval rate over the SSA specifically at the top 10% ranked percentile. For the MDDR COX class, the fitness function score based on the top 1% ranking (Figure 6.11a) managed improvement in actives rates over the SSA R4. This is true especially in the top 1% of ranked compounds, and in a majority of other percentiles up to the top 10% ranked percentile. For the fitness function score based on the top 10% ranking (Figure 6.11b), the GP is able to achieve improved rate of actives in the top 1% of ranked compounds and in other ranked percentile as well. A closer comparison however shows that the fitness function based on the top 1% achieved a slight improvement over the other fitness based on the top 10% ranking. Table 6.2 further shows the actual actives recall rate in the top 1% for the different fitness functions experimented. The fitness function score based on the top 1% ranking is shown to be superior to the one based on the top 10% ranking. This is true for both cases of MDDR RNN and COX classes. From this, a fitness function score based on the top 1% ranking is chosen as the preferred fitness function definition for the GP-based SSA.

6.6.2 Terminal and function sets for chromosome generation

Two different combinations of terminal sets, as well as two combinations of function sets as shown in Table 6.3 were tested. The objective was to determine the most efficient sets of variables and functions that allowed for the best active retrieval performance. It was important to identify some of the conditions and problems that may arise as discussed earlier in section 6.4.3.1. For the terminal set parameterisation, a combination of VARIABLES_A only set, and another GP implementation using both VARIABLES_A and VARIABLES_B sets were tested.

Based on the enrichment table in Table 6.4, it can be seen that the GP instance using both variable sets is able to outperform the VARIABLES_A only set in terms of active retrieval rate for both the MDDR RNN and COX activity classes. The effect of the different terminal set combinations were analysed for both the MDDR RNN and COX activity classes using the error plots shown in Figure 6.12 and Figure 6.13 respectively. The error plots compare the error rate of the training set with its application in the predicted test set based on each of the GP's iteration. This is to show any evidence of overfitting in the solution generated by the training set, when applied on the predicted test set. In the case of GP-based SSA, the error

rate is calculated as the inverse of the rate of active compounds retrieved in the top 1% of the ranked molecules.

For the MDDR RNN activity class, it was found that a GP run using only the VARIABLES_A set showed large overfitting error from around GP iteration 50 onwards (Figure 6.12a). The training error here recorded reduction in the error rate, but its equivalent test error marks a large increase in the test set error. In contrast, the alternative GP implementation using terminals consisting of both the VARIABLES_A and VARIABLES_B sets records a consistent performance of the GP for the predicted test set instances, as seen in Figure 6.12b. It was noted that both types of GP implementation recorded a similar training-based retrieval performance, as witnessed by their training error rate of around 0.41, but their equivalent predicted-test set performance for both varies. A slightly different trend is witnessed for the MDDR COX activity class, where the GP implementation using only terminals of the VARIABLES_A set (Figure 6.13a) causes a large gap in performance between the training set and the predicted test set instances, particularly in the early iterations of 20 onwards. It is only in the later iterations of 160 onwards that the training and test set performance is stabilised. This contrasts with the GP implementation using both the VARIABLES_A and VARIABLES_B terminal sets (Figure 6.13b), which maintains a consistent performance between the training and test set instances. With regard to the functions, the combination of FUNCTIONS_A sets with a log function wrapping the equation introduced by FUNCTIONS_B shows a slight increase in the enrichment values for both activity classes when compared to only FUNCTIONS_A set.

6.6.3 Chromosome structure

Model simplicity is the primary goal of the law of parsimony, otherwise known as Occam's razor, in which a solution should not be excessively complex if it can be solved by another simpler solution. In GP, a critical issue may arise when the model becomes too complicated, but does not yield improved solution suitability with each added complexity. This is often known as the "bloating problem". Bloating occurs when within each generation of chromosome evolution, the chromosome continues to grow larger in size but without having any similar (positive) effect on fitness suitability. Several studies have suggested that restrictions be imposed on chromosomes in order to minimise the bloating effect (Langdon, 1999; Poli et al., 2007). Here, two parameters were tested: (i) a chromosome tree depth

consisting of the following tested values: 4, 6, 8 and 10; and (ii) the maximum number of nodes with the following values: 20, 25, 30, 35, 40 and 50.

Several observations can be made from the enrichment table in Table 6.5. First, it is stressed that while the maximum tree depth parameter is set to a certain value, the actual depth of the generated chromosome is often much smaller than the depth limit defined. For example, consider the case of a tree depth parameter of 10 from the table, in which the generated chromosomes for both the MDDR RNN and COX activity classes yield equations with a smaller tree depth of 6. Similar behaviour is observed with regard to the maximum nodes parameter, where in the case of a maximum node size of 30, the actual node size generated for both classes are 25 and 18 respectively.

In terms of the actual performance of the various parameters in this group, a maximum tree depth of 6 and a maximum node of 30 were observed to achieve the highest retrieval rates when compared with the other parameter options presented. The larger tree depth and node size limits, however, did not yield any significant benefit in terms of the active retrieval rate.

6.6.4 Population size, generation and tree properties

It is stated that programs within GP populations tend to increase rapidly in size as the population evolves and, if unchecked, they might consume excessive machine resources (Langdon 2000). A traditional way to overcome this is by enforcing a size or depth limit on the programs, in which the effects of size and depth limits are presented in Section 6.6.3 above. Under this parameter group, three parameters were tested which define the chromosome population. These can assist in the management of population and solution search space: (i) the population size, (ii) the maximum GP evolution iteration, and (iii) the construction method of the chromosome tree.

With regard to the chromosome population size parameter, several different values were tested beginning with the smallest value of 100, 200, 300, and 500. On the maximum evolution iteration, the following values were also tested: 50, 100, 200, 300 and 500. One of the main objectives was to identify the maximum limit of population and iteration count that had a direct impact on the run time efficiency. For example, if one GP run with a large population size and an equally large maximum iteration remains stagnant after a designated iteration. It is necessary to consider smaller values of both parameters, as no benefits are to be

gained with the larger ones. For the population size parameter test, except for the population size parameter of 100, there was a somewhat similar level of performance in terms of retrieval rate for both activity classes between the population sizes of 200, 300 and 500, even though the size parameter of 200 is slightly higher than the others. It is argued that the population size of 100 may not have reached the convergence state due to it being smaller. Having a larger population size, in this case a value of 500, does not dictate the performance rate, especially when compared to the population size of 200.

For the iteration parameter, the test was setup in such a way that the GP application was executed with a maximum iteration of 500, reflecting the largest iteration value to be tested, as listed in Table 6.6. It was necessary to record the performance of active retrievals in the predicted test set at iterations of 50, 100, 200, 300 and finally 500. From the table, it was observed that (somewhere) between iteration 100 and 200, both activity classes recorded their peak retrieval performance, which remained stagnant up to the 500th iteration.

Three chromosome population construction methods were tested: (i) the grow method, (ii) the full method and (iii) the ramped half-and-half method. Based on the retrieval performance for each parameter, it was found that the grow method generally yielded higher retrievals compared to the other two methods. It was therefore chosen as the population tree construction method of choice.

6.6.5 Elitism model

The elitism model serves to preserve high fitness valued chromosomes during each evolution iteration, which might otherwise become mutated into lesser chromosomes. Elitism ensures that the chromosome is untouched and brought forward to the next generation of genetic evolution. For this parameter, the following elite chromosome preservations were tested: 0 or no preservation, 1, 2, 3, 5 and 10. Similar to the GA parameterisation test, also tested were the steady state model for verification, in which 199 chromosomes are preserved for each evolution out of the 200 total chromosomes. Table 6.7 shows an elitism of 0 was not able to achieve high retrieval rates compared to the different values tested. This is strikingly similar to the behaviour in the GA program, in which zero elitism causes all chromosomes to be mated and mutated completely at random. This subsequently affects the chance of self-preservation of the promising candidates. An elitism of 2 shows a higher degree of retrieval rate for both activity classes than the other parameter values. Likewise, the steady state model

of 199 elitism recorded very low enrichment rates much like its counterpart in the GA program.

6.6.6 Evolution control

As in the GA experiment, the following parameter sets grouped under the evolution control category were identified: (i) the Parent selection method, (ii) the Crossover rate, and (iii) the Mutation rate. From these parameters, a number of different parameterisation combinations were tested, with the options and their corresponding results listed in Table 6.8. The parent selection method deals with the method of choosing two parents for crossover and mutation purposes. Three methods are already established in the literature. These are: (i) the roulette wheel method, (ii) tournament selection and (iii) the random selection method. From the test results, it was observed that in both activity classes, the roulette wheel method performed slightly better than the other two methods tested. The crossover and mutation rate selected, however, were somewhat different from the values obtained in the GA experiment, with the best crossover rate recorded at 90 percent and a mutation rate of 20 percent for both cases of RNN and COX activity classes.

6.6.7 Final parameterisation selections

Extensive parameterisation tests of the GP were carried out in order to maximise GP search potential. From the tests, several parameters were found to be critical to GP's performance while other parameters were less sensitive.

The GP's choice of terminal variables for its chromosomes definition was found to be the most critical. A combination of both the six main SSA variables and sub-terms extracted from various SSA equations was found to maximise GP's search capability, as well as ensuring that the GP does not introduce a high probability of overfitting. Tests have also showed that the relevancy of mandatory log operations can stabilise weight values generated by the equations. Similar to the GA's case, the elitism parameter was also found to require values in the range of 1 to 5 to optimise GP-based results, as having either zero or extreme elitism values negatively affects the GP's performance. Tests showed that the chromosome tree depth and tree node size parameters retrieved comparable performance between the values in each parameter. The exception was the use of a small value for the maximum tree depth which affected the GP's results. For the chromosome population size and generation limit, the parameters were sensitive when small population and iteration values were used. By contrast,

its performance was not significantly affected by larger population and longer iterations. Finally, for the evolution control parameters, which include the parent selection method, crossover and mutation rate, much like the GA-based SSA, only fine tuning of the parameters were required to maximise active retrieval performance.

Based on the parameterisation results shown in this section, it was decided to apply the following GP parameters for the next experiment to benchmark the GP performance against GA and SSA, given all of the activity classes from both databases. The finalised parameters are as follows: (1) *Fitness function* of active rates in the top 1% of ranked compounds; (2) GP terminal set formed of the VARIABLES_A and VARIABLES_B sets and GP primitive function set formed of FUNCTIONS_A and a mandatory log operation at the end of the equation based on the FUNCTIONS_B set (both terminal and function sets as listed in Table 6.4); (3) Chromosome *maximum population* of 200; (4) GP *maximum evolution* at 200 iterations; (5) the grow population construction method; (6) An *elitism* of one chromosomes; (7) *maximum tree depth* of 6; (8) 30 *maximum nodes* in a tree; (9) The *roulette wheel* parents chromosome selection method; (10) *Crossover rate* of 0.85; and (11) *Mutation rate* of 0.20.

6.7 Experiment result: Analysis of the performance of GP-based SSA

Having finalised the suitable parameterisation of the GP, it was possible to run the GP-based SSA in order to quantify its performance relative to both the GA-based SSA and the R4 weighting scheme. Several performance analysis criteria are discussed below.

6.7.1 GP robustness

To quantify the randomness factor in the solutions obtained from the GP method, a robustness test was performed by analysing the results of the 10 GP runs. The main focus was on the two activity classes RNN and COX from the MDDR database as inputs into the GP robustness test. The retrieval rates of the two classes above are plotted as enrichment curves, shown in Figure 6.14 (a-b). The deviations in the retrieval rates at different ranked percentiles were small for both classes, and in comparison, were much smaller than its GA-based counterpart (Figure 5.12). It is concluded that the GP-based SSA is capable of producing effective and consistent results, confirming the robustness of the GP results.

6.7.2 GP weights correlation and consistency of compounds

Similar to the GA, the relationship of the fragment weights generated by the 10 GP runs was measured for both the MDDR RNN and COX classes (Section 5.7.1) using the Pearson's *r* correlation coefficient. The Pearson's *r* correlation coefficient is an indication of how well the generated GP weights relate to one another between the multiple run instances.

For the RNN class, high correlation values were obtained with a minimum of 0.85 between the individual GP run instances (Figure 6.16a), and similarly for the COX class in which a minimum correlation value of 0.99 was recorded (Figure 6.16b). The mean and standard deviation for the Pearson's *r* averaged over the 45 pairs of runs for each activity class are 0.91 and 0.015 (RNN) and 0.99 and 0.001 (COX). The correlation levels are indeed higher when compared to the 10 runs of the GA-based SSA analysed in Chapter 4.

The consistency level of the compounds retrieved for the ten GP-based SSA runs was further analysed. The results are shown in Table 6.9, where the first GP run (dubbed GP_run1) is assigned as the reference run. Its active molecules in the top 1% are identified and compared to those of the corresponding nine remaining runs. The upper value for each run describes the number of different active compounds in the top 1% compound ranking which are not present in the equivalent top 1% of the GP_run1. The lower values shown in brackets describe the actual number of active molecules retrieved in the top 1% of the test set of the ten GP runs. There were marked smaller differences in the active molecules for the other nine GP runs with reference to the GP_run1 for both activity classes, especially when compared with the GA-based SSA results. In particular, for GP_run2 of the MDDR RNN class, the active compounds retrieved were identical to those obtained in the reference GP_run1. Other GP runs recorded a small level of variation with a maximum of 25 different active compounds retrieved in GP_run9 (Table 6.9a). The trend is similar for the MDDR COX activity class (Table 6.9b), where the different active compounds retrieved from multiple runs ranged from 0 to 20. From the total of 173 active compounds retrieved in the top 1% of the COX GP_run1, this accounts for roughly a 0-10% change for the other runs. These results show a strong indication of consistency in performance obtained from multiple GP runs, thus outperforming the GA-based SSA in this respect.

6.7.3 Analysis of GP runs on all activity classes

Table 6.10 presents the enrichment factor of actives retrieved in the top 1% of the ten GP runs for each activity class from all three databases. Each row of the table corresponds to a single activity class which lists the total number of actives retrieved in the test set using the GP weighting scheme. The final two columns on the right for both Table 6.10 contain the mean and standard deviation of the number of actives retrieved in the top 1% of the ranked molecules. From the mean and standard deviation values, it can be seen that there is a high degree of clustering of active compounds in the top 1% of the ranked data. The variation in the actives retrieved in the top 1% is less dispersed between the multiple runs. Table 6.10 also provide the mean Pearson correlation coefficient and its standard deviation between the sets of 166 weights for the 10 runs on each class. The mean correlation values for all activity classes across the three databases are small with low standard deviations. For the MDDR database, the majority of the classes achieved a mean correlation score of 0.80 and above, except for the AT1 class which scored a mean r value of 0.75. Three classes in SUBP, COX and PKC all scored a perfect correlation score of 1, meaning that for the 10 GP runs executed, the resulting equations generated were similar for each class. Similarly, for the WOMBAT database, 5 out of 14 activity classes scored a complete r value of 1. These are the 5HT3, ACHE, AT1, PDE and THRM classes. The least correlated class is the MPP activity class with a mean R of 0.79. Finally, for the ChEMBL database, 7 activity classes achieved a complete r value of 1. These are the 5HT3, AT1, COX, D2, FXA, MMP and PDE classes, while the lowest mean correlation is much higher than the other database, with a minimum score of 0.92.

Three observations can be made regarding the results from Table 6.11 of this study. First, it was observed that the GP-based SSA method performs better than the SSA R4 in all cases when comparing both the GP runs (i.e. the worst, the best and the mean of the ten runs) in each activity class. Second, however, when comparing the GP and GA in each activity class, the GP-based SSA was found to be slightly less effective than the GA searches from all databases. The GP-based SSA (i.e. the worst run) was better for 11 of the 40 activity classes (from the MDDR database, the classes are 5HT, D2, AT1 and COX; from WOMBAT the classes are ACHE, PKC and THRM) and from ChEMBL the classes are AT1, MMP, PKC and SUBP), while the remaining 29 activity classes achieved improvements using the GA method. Moreover, upon comparing the number of actives using the evolutionary computation method (i.e. the best run of GP and GA), the GP run showed less actives

compared to the GA method, only 6 out of 40 activity classes (from the MDDR database, the classes are 5HT, D2 and COX; from WOMBAT the class is PKC and from ChEMBL the classes are MMP and PKC). Furthermore, the mean number of actives for the ten GP and GA runs was also determined to evaluate the effectiveness of the GP application. Tables 6.12 shows that the mean values derived from the use of GA are consistently superior to the GP in most cases (except for 4, 1 and 2 classes of MDDR, WOMBAT and ChEMBL respectively).

Third, the most interesting finding was that although the GP technique was found to slightly increase the number of actives, the GP method showed a high degree of clustering of actives in the top 1% between the ten runs. In most cases across all activity classes (except for COX class from MDDR and THRM activity class from ChEMBL) the GP showed low standard deviation values between the ten GP runs. This indicates that the GP method has a fair degree of repeatability compared to the GA-based SSA scheme. The GP technique was also found to increase the active retrieval rate when compared to the SSA methods.

6.7.3.1 Enrichment curve analysis

Similar to the SSA and GA experiments, cumulative recall plots of the five activity classes (5HT3, COX, D2, RNN, and PKC) were extracted from the three databases. These included a comparison to the best performing SSA equation (the R4 scheme) and the worst run instance of the GA-based SSA (Figures 6.16 to 6.20). Several observations can be made and these are discussed below.

For the 5 classes displayed, three trends can be observed from the curve plots, and they are described below. The first trend shows a number of classes which reveal total superiority of enrichment value obtained by the GA when compared to the investigated GP-based SSA. Examples are from the ChEMBL 5HT3, WOMBAT and ChEMBL COX (Figure 6.16b-c), WOMBAT and ChEMBL D2 (Figure 6.17b-c), and RNN classes from all three databases (Figure 6.18a-c). In some cases, like the MDDR 5HT3 (Figure 6.15a), the uplift of the GA results is as much as 15% compared to the investigated GP-based SSA in the top 1% ranked compounds.

Several classes showed that the GP is at least similar, or superior to the GA counterpart. WOMBAT 5HT3 class (Figure 6.15b) can be seen to obtain a slightly higher active retrieval rate than the GA in the top 1%. This is followed by other classes such as the MDDR COX

(Figure 6.16a) and WOMBAT PKC (Figure 6.19b). From all the plots, two classes show distinct superiority of the GP-based SSA over the GA and SSA R4 from the top 1% to the top 10% of ranked compounds. These classes are the MDDR D2 (Figure 6.17a), and the ChEMBL PKC (Figure 6.19c).

From the plots, it can be seen that the GP enrichment results are more stable in the top 10% of recall rates when compared to the GA. The GP results are seen to maintain steady improvements over the SSA R4 results, except for a few instances where the SSA R4 exhibit extreme superiority of retrieval rates in the later ranked percentiles (WOMBAT PKC in Figure 6.19b).

6.7.3.2 Analysis of diversity

A diversity analysis was conducted on the GP results to quantify the structural diversity of the ranked compounds. The results are compared with those of the GA-based SSA and the R4 scheme. The number of distinct Murko scaffolds in the top 1% of ranked actives was calculated using Pipeline Pilot software. The mean pairwise similarity values were also observed by calculating the similarities between each compound in the activity classes using the Tanimoto coefficient. The average of the similarity values was considered to show the diversity level of the number of actives retrieved for each activity class using the GP method. The diversity of the actives achieved from the worst run using the GA and the GP-based weighting schemes. The actives obtained from SSA R4 are compared in Table 6.11. From the table, it is clear that the GA results possess the highest diversity of compounds, followed by the GP method, and finally the SSA R4 method. The method with the highest diversity for each class is also shaded in the table.

6.7.4 Kendall's W analysis

For further analysis, Kendall's W was used to quantify the degree of association between eleven, fourteen and fifteen sets of rankings from the MDDR, WOMBAT and ChEMBL datasets, respectively, from the 3 weighting schemes (i.e. GP-based, GA-based and SSA R4). The results of the Kendall's W analysis are summarised in Table 6.12. It can be seen that the performance of the SSA weighting schemes is listed in decreasing order based on the mean ranking of the actives in the top 1% for each activity class when summed across all of the rankings. The results obtained from the Kendall's W analysis for the MDDR searches are shown in Table 6.12(a). The computed value of W is 0.77. References to the critical values

table reveal that the value of W is significant at the $p < 0.01$ level. A similar ranking trend was obtained when using the WOMBAT and ChEMBL databases, with a 1% cut-off value as shown in Tables 6.12(b) and 6.12(c). The value obtained for W was computed as 0.75 and 0.88, respectively, at the $p < 0.01$ level of statistical significance. Overall, the Kendall's W test for the ranked compounds in each class of the MDDR, WOMBAT and ChEMBL databases revealed that there is a strong agreement between the three weighting schemes. The analysis indicated the following rankings:

$$GA\text{-based SSA} > GP\text{-based SSA} > SSA R4$$

Taken together, these results indicate that the GP-based method performs less well than the GA-based method, while the SSA R4 performs worst in all the MDDR, WOMBAT and ChEMBL datasets, respectively.

6.7.5 Wilcoxon signed rank test

It was observed that the GA outperforms the GP for all databases as revealed by the Kendall's W test. The research therefore proceeded to use the Wilcoxon signed rank test to determine the direction of superiority of a measure between the performance of the GP-based and GA-based results for each class. To conduct the test, the enrichment factor of actives of the top 1% of the worst and best runs using the GP and GA methods (as reported in Chapter 5) was observed. Further information on the statistical tests can also be found in Chapter 3, Section 3.5.3. Overall, the results that compared the worst run obtained from the GP and GA-based methods show that the observed W for both databases is larger than W_{critical} (i.e. $W \geq W_{\text{critical}}$) at $p \leq 0.01$, as referred to in the table of critical values of the Wilcoxon signed rank test. No significant differences were found between the the worst run of the GP and GA for all three database cases. It can be concluded that the performance of the GP-based method was similar to that of the GA-based run. However, when comparing the best GP and GA runs, the Wilcoxon signed rank revealed that there was a significant difference between the two runs except for MDDR.

Furthermore, to identify whether the GP-based method is practical for SSA application, the mean of the ten GP and GA runs were compared. The Wilcoxon signed rank test was also used to quantify the magnitude of the superiority of a measure against the two methods. The results show that these differences were statistically significant except for the MDDR dataset.

In summary, these results indicate that there were significant differences between the means of the ten GP and the GA runs; hence, the performance of the GP-based method can be said to be less than that of the GA runs.

6.7.6 GP-generated equations

Equations produced by the GP-based SSA were observed. Table 6.13 presents both the worst and the best solutions obtained on all the three and for each activity class. Note that the equations listed in Table 6.13 have been simplified from their original form: an example is provided in Figure 6.21. The freely available Wolfram Alpha Expression Simplifier tool (n.d., Retrieved from <http://www.wolframalpha.com/>) was used. From Table 6.13, several observations were made.

First, it was found that there was no occurrence of repeated, identical equations generated by the GP in at least two different activity classes from the three databases. Several activity classes, however, were observed to obtain exactly identical equations produced during its 10 GP runs, such as those seen in the MDDR SUBP; WOMBAT ACHE, PDE and THRM; and ChEMBL AT1, COX and PDE classes.

In terms of variable population, several equations were found to be formed using all six main variables (*N*, *NACT*, *NINACT*, *TOT*, *ACT*, *INACT*). These can be seen in the best equations of the MDDR 5HT and D2; and WOMBAT 5HT3 classes; or in the worst equation of the ChEMBL THRM class. In contrast, the equation with the least amount of variables used are found in the worst equation of WOMBAT PKC class, or the best equation of ChEMBL FXA class. All of them employed only three types of variables. It can be seen that the most frequent variable used to form the equation is the *ACT* (the number of particular fragments present in the active compounds), followed by *INACT*, *TOT*, *N* and *NACT*. The least used variable in the equations was determined as *NINACT* (the number of inactive compounds in the database), which appeared only 13 times in the GP equations across all databases. The variable *ACT* was observed to appear in all of the equations generated for each activity class.

6.7.7 Model validation with Y-randomisation

Similar to the GA experimentation, a model validation of the GP-based SSA using the Y-randomisation technique was performed to investigate presence of chance correlation. The technique utilises scrambling the training dataset and is used to predict a model. For our

scrambling test, the previously generated 100 scrambled training set comprising 10% of active and 10% inactive compounds from the MDDR-based RNN and COX activity classes were used previously employed in Chapter 5. Note that the training sets above contain scrambled compounds activity state while leaving the fingerprints and compounds class labels intact. The resultant weights from GP-based SSA using the scrambled sets above were then applied to the test set.

Recall the ten repeated GP runs, as discussed in Section 6.7.1 (GP robustness test). For the purpose of model validation, the 10 GP instances are referred to as the unscrambled GP runs. Subsequently, it was possible to identify two meaningful variables to observe: (i) Pearson's r coefficient, based on the weight distribution when compared to the first run of the unscrambled GP result, hereby dubbed unscrambled_GP_run1; and (ii), the active molecule retrieval rate in the top 1% of the compound ranking. The scrambled GP runs should not match or correlate well with unscrambled GP runs, since in doing so, it means that there is the presence of chance correlation using deformed data. To calculate Pearson's r , it was necessary to correlate the sets of 166 weights of the unscrambled_GP_run1 with the sets of 166 weights obtained from the 100 scrambled GP runs. The remaining nine unscrambled GP runs were also correlated with the unscrambled_GP_run1. Similarly, the percentage of actives retrieved in the top 1% of active compounds was observed for both the scrambled and unscrambled GP results.

Figure 6.20 (a-b) shows the Y -randomisation plot for the two MDDR-based activity classes; the RNN and COX. The X -axis of the plot represents the retrieval rate in the top 1% ranking, while the Y -axis denotes the Pearson's r values versus the unscrambled GP_run1. From both figures, the ten unscrambled GP results performed positively, clustering in the top right side. This signals a similarly high active retrieval rate, while maintaining a good correlation with the unscrambled GP_run1. This is the opposite of the scrambled GP runs, which recorded very low correlation values of Pearson's r (none higher than 0.35). All of the scrambled results also failed to retrieve comparable actives in the top 1% of the ranked compounds for both classes, especially when compared to the unscrambled ones. The mean and standard deviation of the number of actives retrieved in the top 1% of the scrambled GP cases were 8.64 and 15.68 (RNN). Two standard deviations above and below the mean creates a range from -22.72 to 40.01. Hence, of the 100 runs, 95 runs fall into this range, which is 95% of the runs. The mean and standard deviation of the numbers of actives retrieved in the top 1% of

the scrambled GP cases for the COX were 5.05 and 7.22. Two standard deviations from the mean create a range from -9.40 to 19.50, where 91 runs fall into the range. Based on these results, it can be seen that the GP-based SSA is unable to arrive at chance correlations with deformed data.

6.7.8 Run-time benchmarks of GP-based SSA

Similar to the GA, the execution of the GP-based SSA was fairly intensive on a single computer, depending on the data size and choice of parameters. A run-time analysis to understand hardware suitability in different data scenarios is discussed here. The hardware used follows the ones listed in Table 3.2.

The RNN activity class was used as the benchmarked class for all three database instances, with the same parameterisation set as discussed in Section 6.6. Table 6.14 lists the breakdown of the run-times of the GP for the individual iterations and subsequently the total run-time for a complete GA program. These are based on 200 maximum GP iterations. For the MDDR-based RNN class, the 10% training set is made up of 10,254 compounds, while for the WOMBAT-based class, the 10% training set is equivalent to a total of 13,812 compounds. The ChEMBL-based RNN activity class is significantly larger for the case of a 10% training set, consisting of 135,267 compounds.

For the MDDR-based RNN activity class, as shown in Table 6.14(a), a GP run using 10% training set runs on average between 2.70 and 3.50 seconds per GP iteration, with a total run-time for the GA program averaging between 532.5 seconds and 721.4 seconds, equivalent to 8.8 and 12 minutes respectively. Table 6.14(b) shows the run-time breakdown for the WOMBAT-based RNN activity class via a 10% training set of GP run. The average run time for a single iteration is between 2.80 and 4.10 seconds, with the total run-time of all 200 iterations averaging between 578.4 and 933.1 seconds (9.64 and 15.5 minutes), respectively. For the ChEMBL case shown in Table 6.14(c), similar to the GA case reported in Chapter 4, it was noted that the machine WKST_01 was not able to execute the GP-based SSA due to the problem of its small memory limitation; it only has 4GB DDR of physical RAM. Between the SERVER and WKST_01 machines, the run-time of a single iteration was timed at 37.20 and 48.10 seconds respectively. The total run-time of a complete GP run in the ChEMBL case was 7621.50 seconds for the SERVER machine and 9680.5 seconds for the WKST_01 machine; this translates to roughly between 127 and 161 minutes respectively. From the

above, it may be summarised that in terms of run-time performance, a fairly mild influence of the machine's processor architecture on increasing the run-time efficiencies was observed. This should not however, be a critical factor in the consideration of hardware choice for running GP-based SSA. This is a similar behaviour to the GA-based SSA.

6.8 Discussion

Various analyses were performed to gauge the performance level of the GP-based SSA. The primary method of analysis was based on the active retrieval rates for each activity class in the three databases. These were compared to equivalent implementations of the SSA R4 and GA-based SSA. The screening results were analysed by determining the worst, mean and best results of the 10 runs executed for both the GP and GA, and also from the SSA R4 obtained results. Based on this analysis, it was observed that the GP method outperformed the SSA R4 in all cases of activity classes. Compared to the GA method, however, it was generally found that the GP is less effective than the GA in its comparison of the worst, mean and best runs of the two methods. The exceptions were a few cases where the GP obtained superior retrieval of active rates to that of the GA-based SSA, particularly the MDDR 5HT,D2,COX and AT1; WOMBAT PKC, ACHE and THRM; and ChEMBL MMP, PKC and AT1 classes. On the other hand, a positive outcome of the GP is seen in its extremely high correlation between runs, based on an investigation into the GP's robustness and consistency of active compound recall in the top 1% of ranked molecules. The diversity rates of actives in the top 1% and Murcko scaffolds show the GP to be less effective when compared to the GA. Several class instances, however, demonstrate the GP's capability of improvement in the two values above. Permutation tests for model validation proved that the GP would not be able to generate a successful solution if trained from randomly generated datasets.

Cumulative recall plots show that the GP-based SSA, similar to the GA, maintains a stable trend of improved retrieval rates in the top 10% of ranked molecules over the SSA R4. Several classes across all three databases showed that the GP outperformed both the SSA and GA methods in up to the top 10% of ranked molecules, such as the ChEMBL PKC class.

A Wilcoxon signed rank test was performed to measure the significance of the difference between the performances of the GP and GA. This was followed by a Kendall's W statistical test to measure the significance of the agreement in the results obtained by the GP and GA methods. The Wilcoxon test indicated that there is no significant difference at the $p < 0.01$

level in the performance between the worst GP and GA results. With regard to the mean and best performance comparison of the GP and GA, a significant difference at the $p < 0.01$ level in the performance of the mean, was found. It was the best run of the GP and GA results particularly for the WOMBAT and ChEMBL databases. The performance of the GP-based method was statistically proven to be less than that of the GA runs.

The Kendall's W test for the ranked compounds in each class of the MDDR, WOMBAT and ChEMBL databases revealed that there is a strong statistical agreement between the performance of the three weighting schemes in all three databases at the $p < 0.01$ level. The GA-based SSA was the best weighting scheme, followed by the GP-based SSA and finally the SSA R4. Based on the Kendall's W statistical test results, retrieval rate performance of the GP was seen to be less effective than the GA. The performance of GA and GP, however, was observed as comparable if Kendall's assessments were made based on the mean of the GA and GP's 10 runs. Therefore, the observed differences between the GP and GA methods in this study were not significant for any of the activity classes in the three databases.

An assessment was made of the suitability of the GP-based SSA in real world application from two related tests. The parameterisation experiments have established the GP-based SSA's performance sensitivity to be critical of several parameters as follows: (i) Terminal variables set for chromosome definition requires the combination of both variables and sub-terms from SSA equations to maximise retrieval performance and overcome overfitting, while (ii) a compulsory log operation performed on the GP equation helps in stabilising weight values. Finally, (iii) an acceptable elitism model was also proven to affect GP's performance. Similar to the case of GA in Chapter 5, other parameters, however, did not significantly affect GP's performance as those listed above, other than the requirements of parameter fine-tuning. In terms of run-time performance, hardware, benchmark tests recorded similar results to those obtained by the GA-based SSA in Chapter 5. The results above demonstrate good practicality of GP-based SSA in real-world pharmaceutical application.

6.9 Conclusion

This study investigated the application of a GP-based SSA method. The research performed rigorous evaluation of the effectiveness of the GP-based weighting schemes in a chemoinformatics application pertaining to 2D fingerprint datasets. In this chapter, it was concluded that the GP-based SSA method is inferior to the GA-based SSA method in most

cases. It can be argued that the main reason of GP's lesser performance is due to its strict requirement to evolve equations without any direct access to individual weights, as opposed to the GA, which explicitly seeks to obtain weight that will optimise the ranking of the test set. However, both the GP and GA methods successfully provide uplifts in the upper-bound of the active retrieval performance especially in the top 1% of ranked molecules. As a consequence to inconsistencies in results obtained by the GP for the three databases, the GP method is considered to be less suitable as an alternative to the R4 weighting scheme.

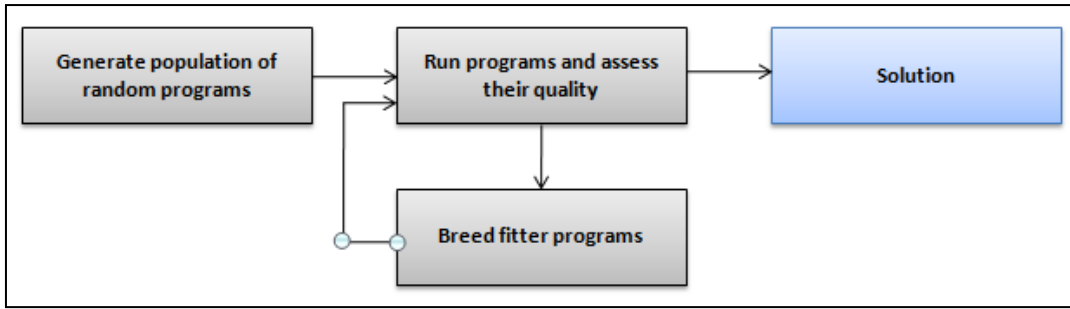


Figure 6.1: Genetic programming basic flow, (after Poli, Langdon, McPhee & Koza, 2008)

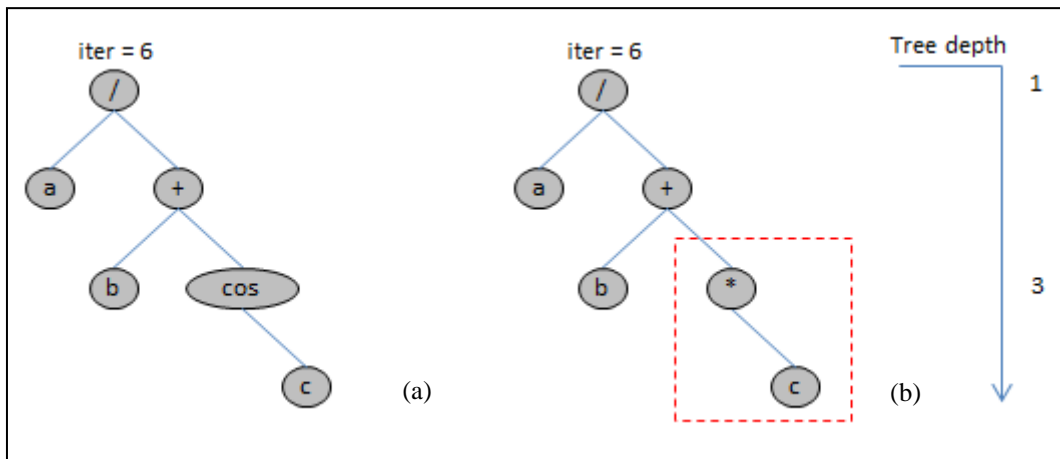


Figure 6.2: Simple structure of a tree model in GP, (a) a valid tree model compared to (b) invalid tree with incomplete equation portion in its child branch, highlighted in red

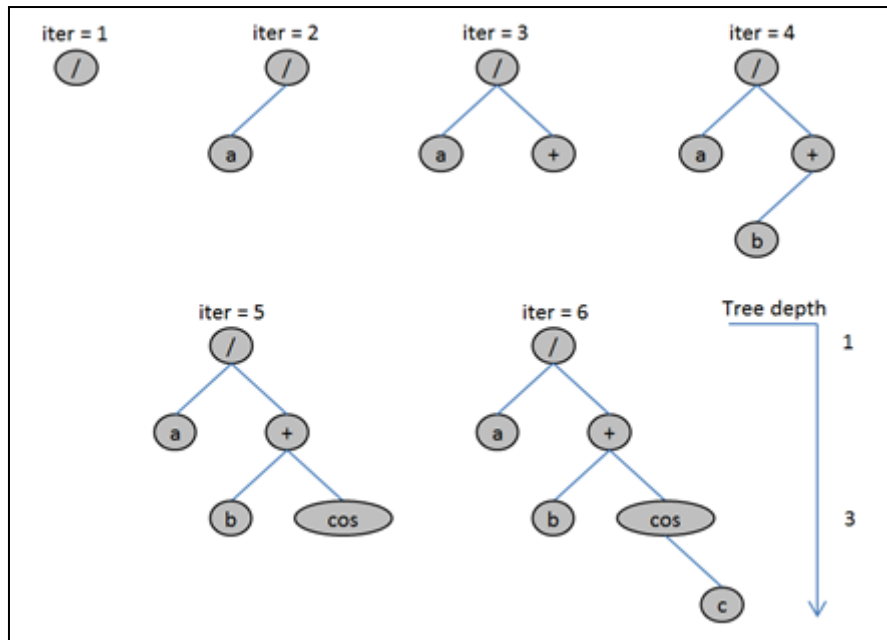


Figure 6.3: Chromosome tree creation using the grow method. Tree defined with a maximum depth of 3 levels

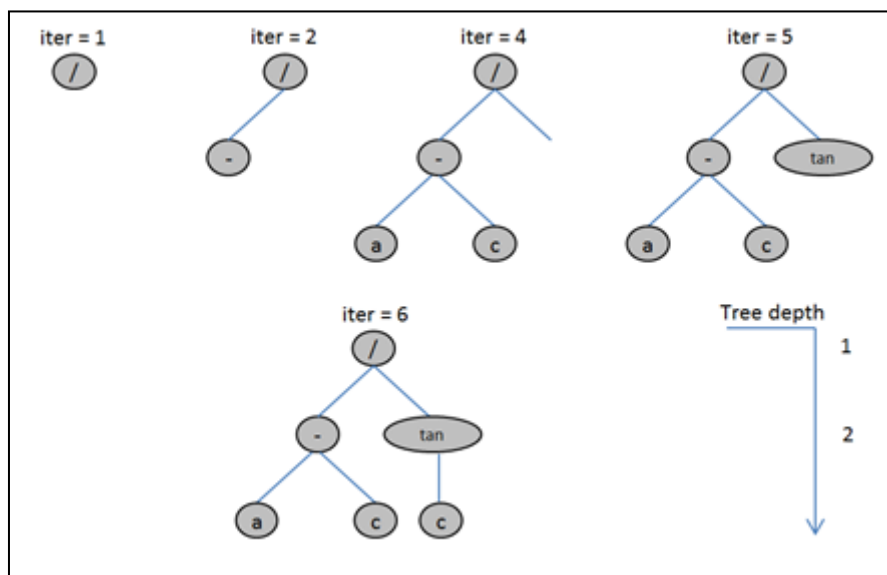


Figure 6.4: Chromosome tree creation using the full method. Tree defined with a maximum of 3 levels of tree depth

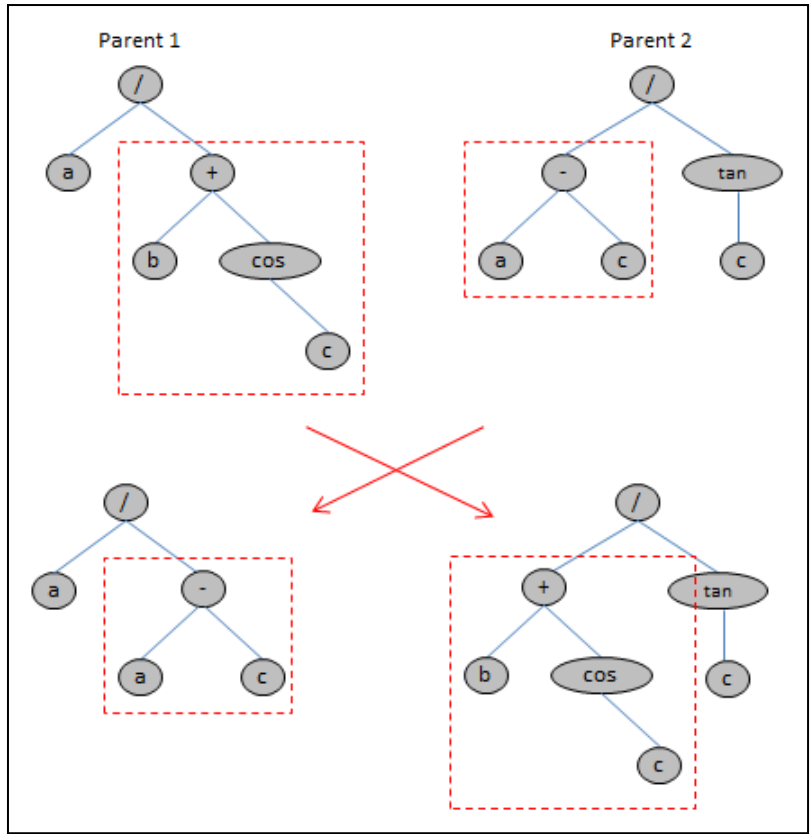


Figure 6.5: GP's crossover operation diagram

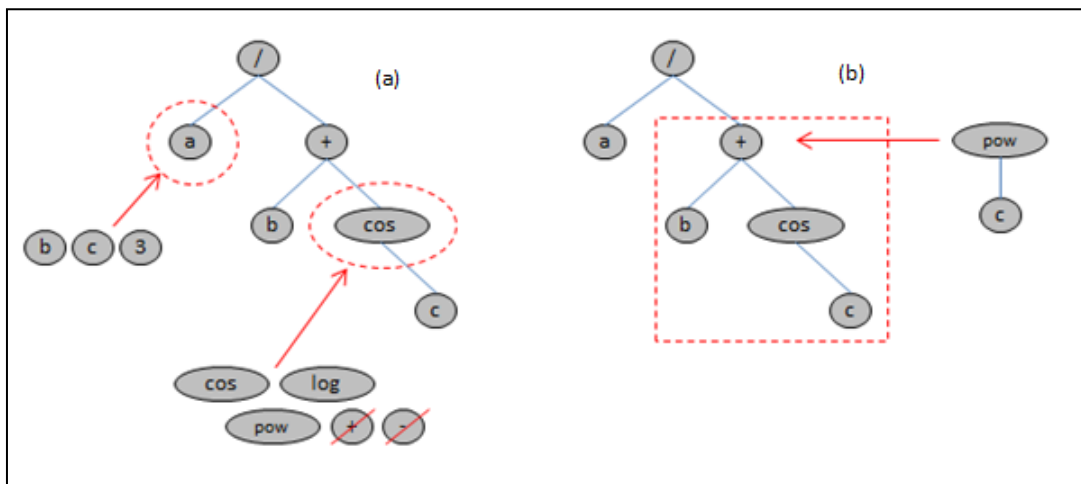


Figure 6.6: GP's mutation example showing the (a) Single terminal mutation, and (b) Sub-tree mutation

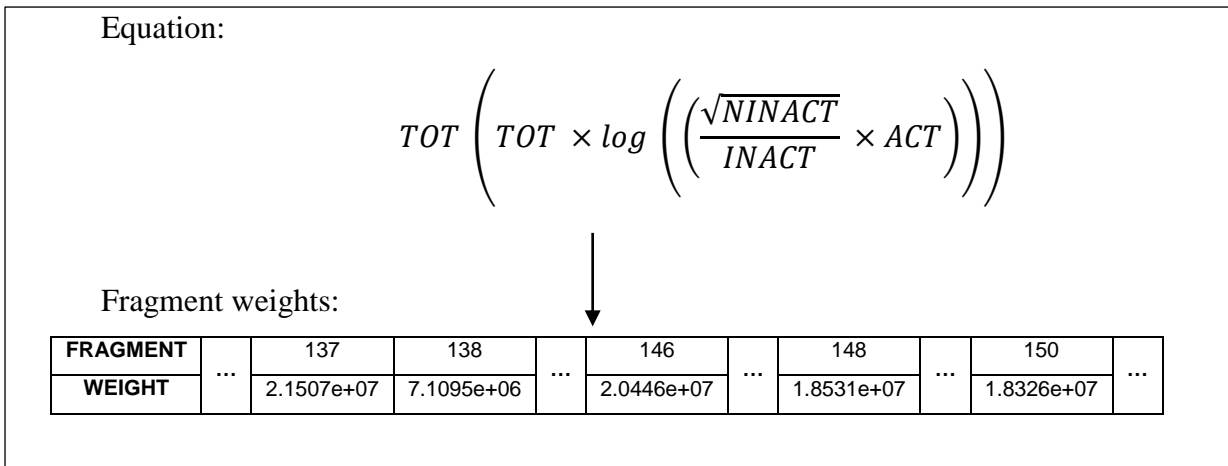


Figure 6.7: A GP equation producing chaotic fragment weights of inappropriate large values. The multiplication of a large variable *TOT* with itself while enhanced by the accompanying exponential term causes a number of weights to be significantly larger in value

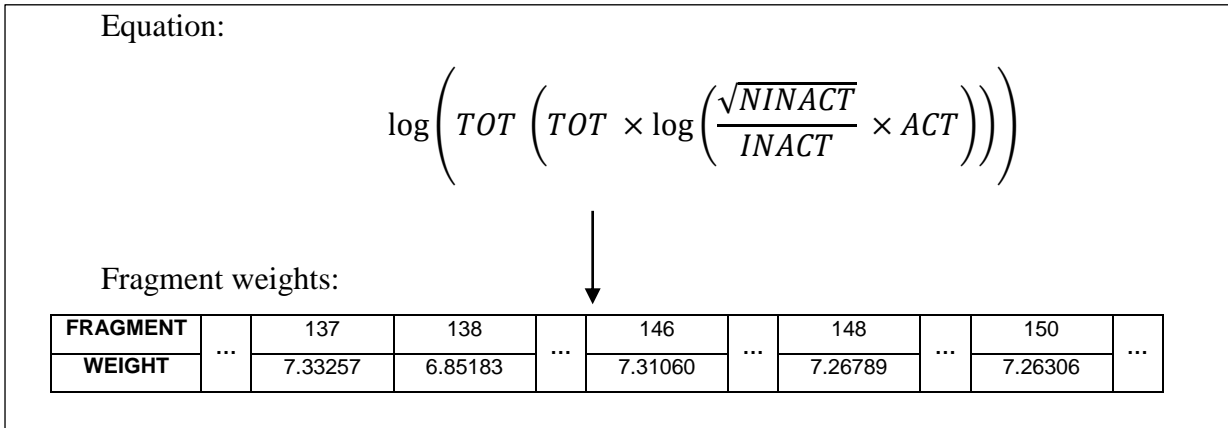


Figure 6.8: An equation from Figure 6.7 now wrapped by a mandatory log function generates much smaller weight values

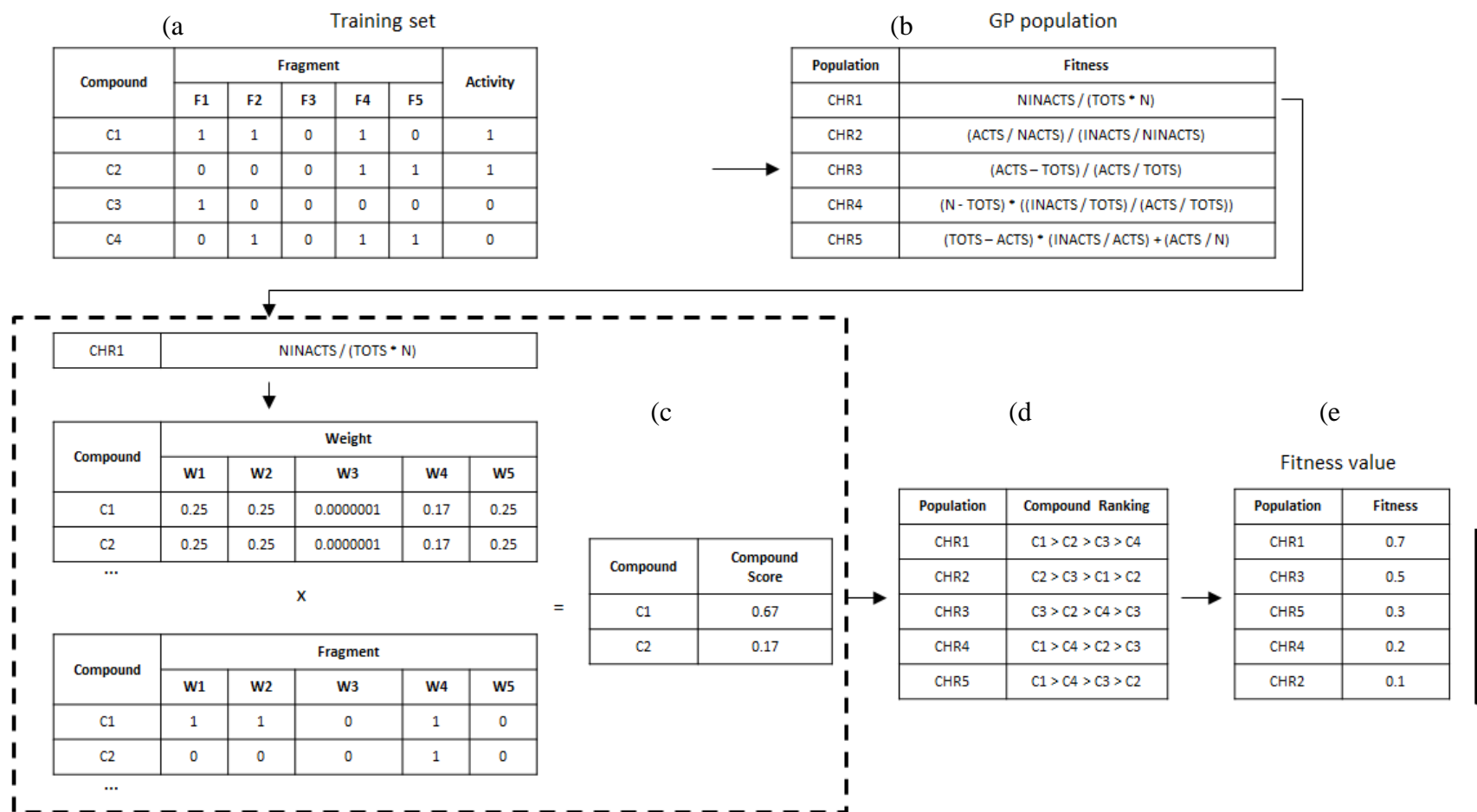
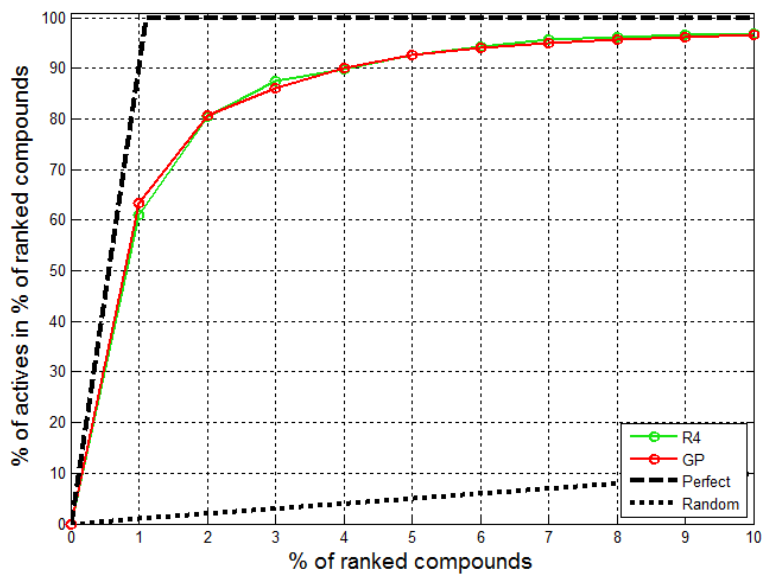
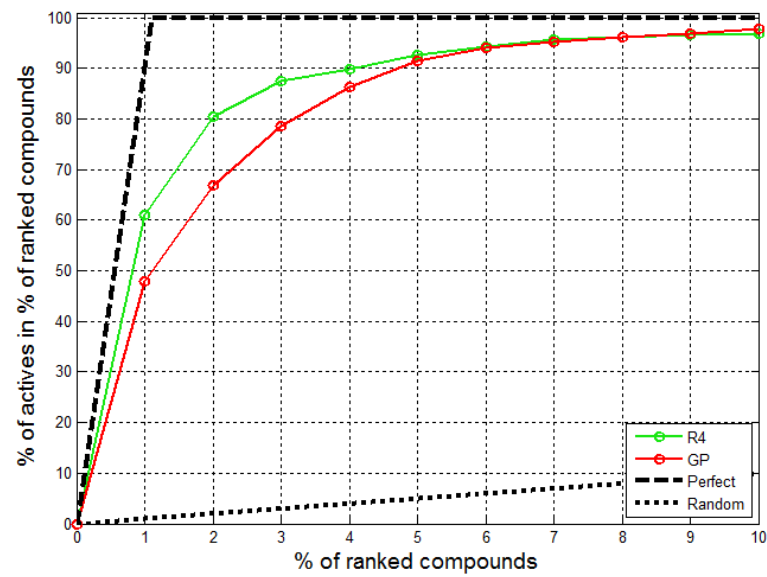


Figure 6.9: GP representation of chromosomes towards fitness determination. (a) A training set made up of compounds via 2D fingerprints description. (b) A GP population representing chromosome equations made up of parameters to explain training set. (c) Chromosome chr1 is translated from the equation form to weight values, applied to training set to determine compound score. (d) Compounds are ranked in descending order. (e) Fitness of chromosome is calculated as the rate of the active retrieval in the top percentile of the ranked compounds set

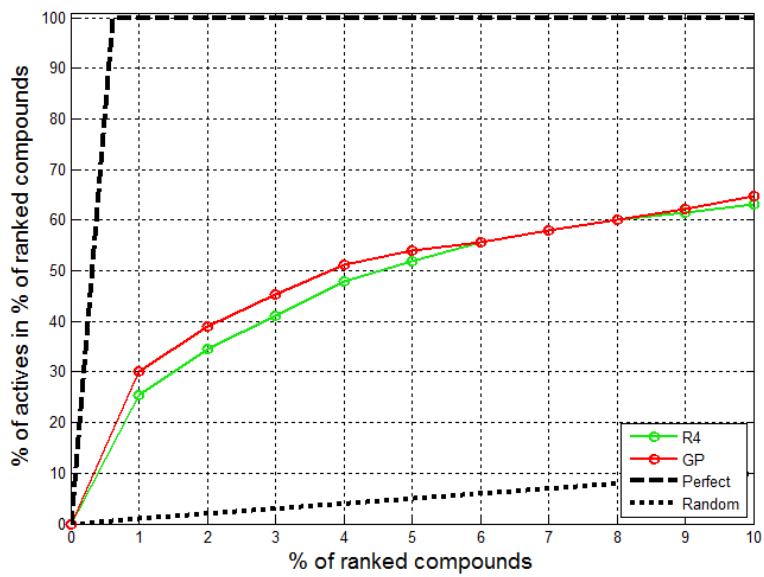


(a)

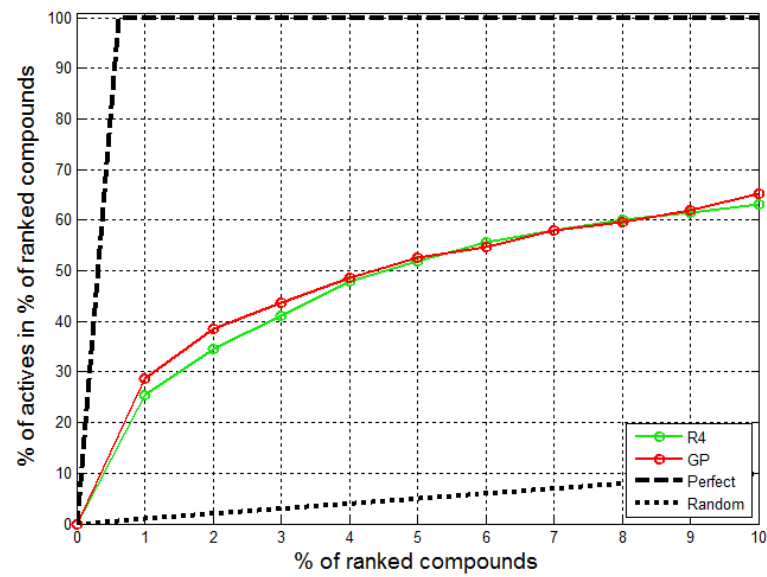


(b)

Figure 6.10: Cumulative recall plots of the GP-based SSA against SSA R4 for the RNN activity class from the MDDR dataset based on the different fitness function in (a) The top 1%; and in (b) The top 10% of ranked compounds

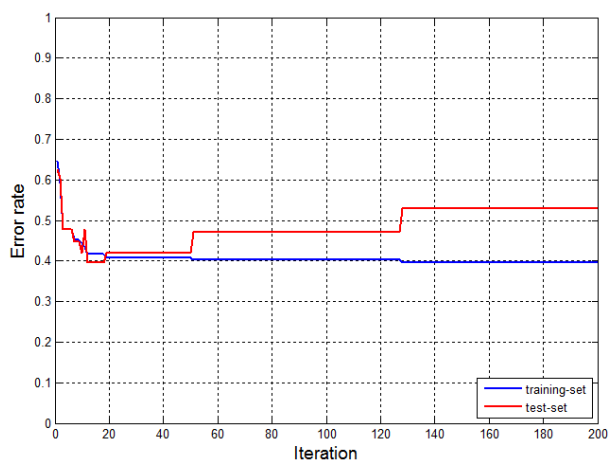


(a)

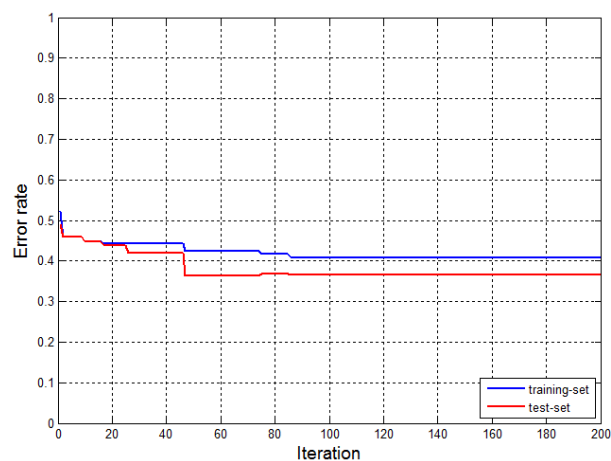


(b)

Figure 6.11: Cumulative recall plots of the GP-based SSA against SSA R4 for the COX activity class from the MDDR dataset based on the different fitness function in (a) The top 1%; and in (b) The top 10% of ranked compounds

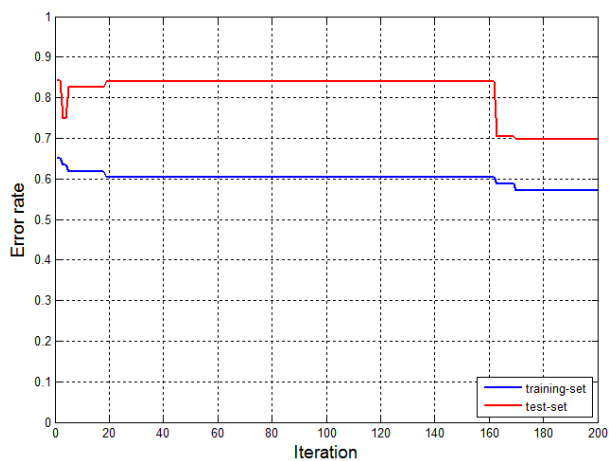


(a)

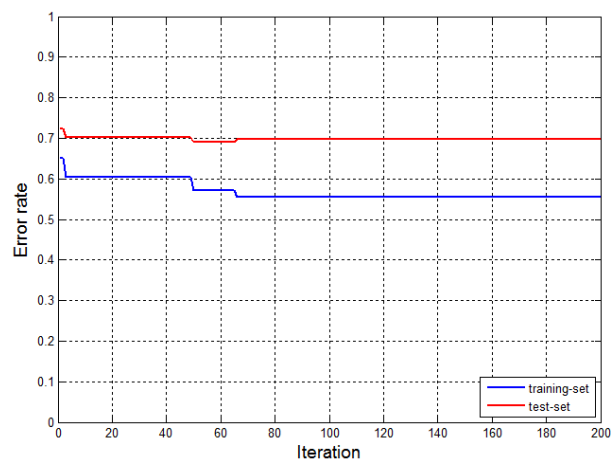


(b)

Figure 6.12: Error plot of training set versus predicted test set of the GP-based SSA for MDDR RNN activity class, based on (a) VARIABLES_A set only, and (b) VARIABLES_A and VARIABLES_B combination

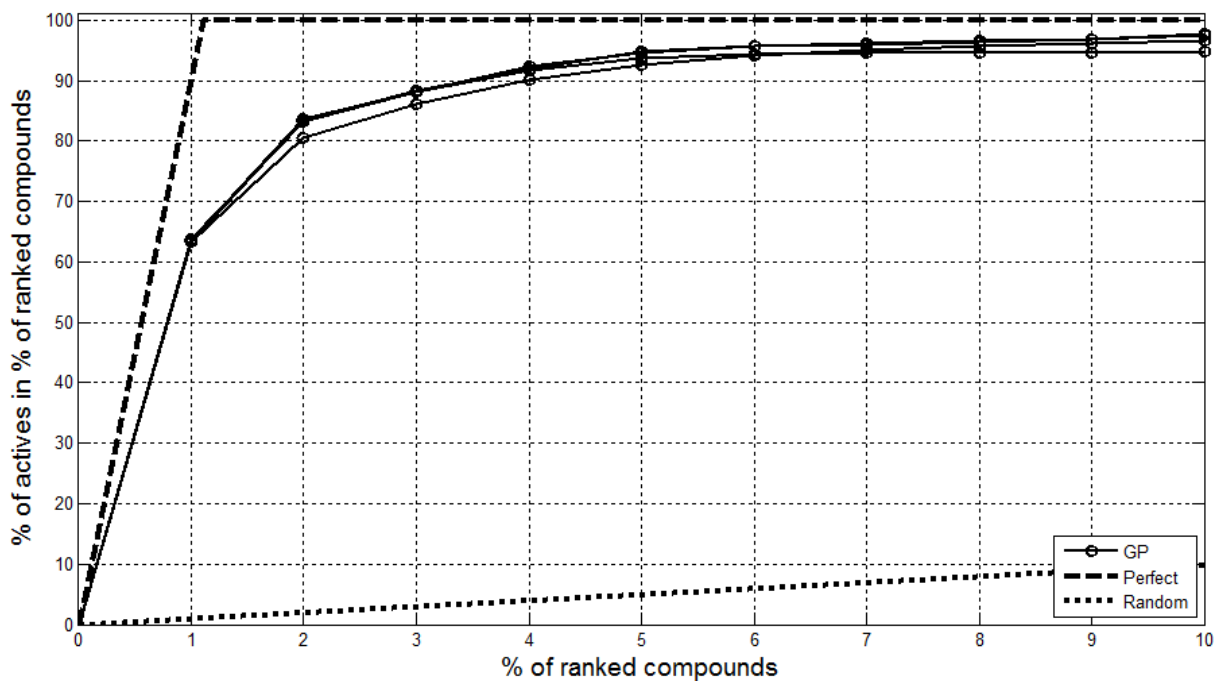


(a)

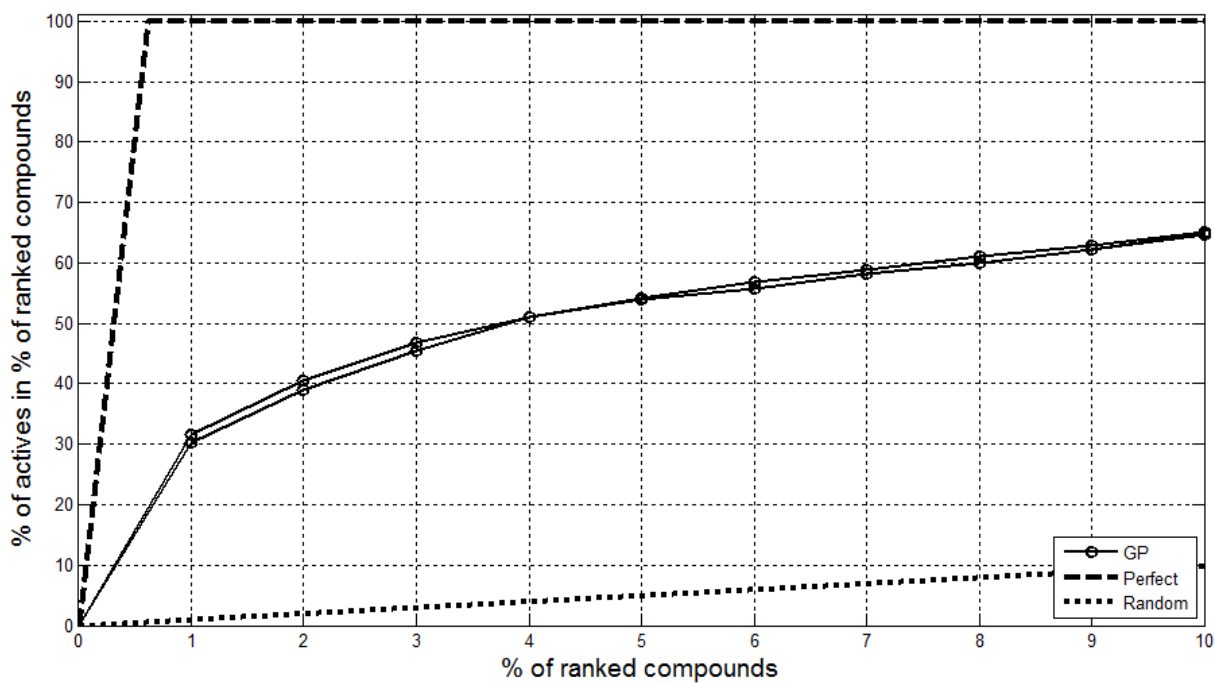


(b)

Figure 6.13: Error plot of training set versus predicted test set of the GP-based SSA for MDDR COX activity class, based on (a) VARIABLES_A set only, and (b) VARIABLES_A and VARIABLES_B combination

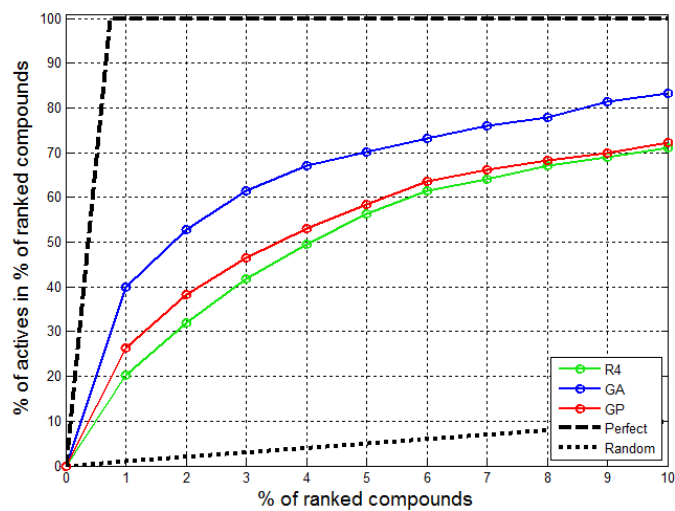


(a)

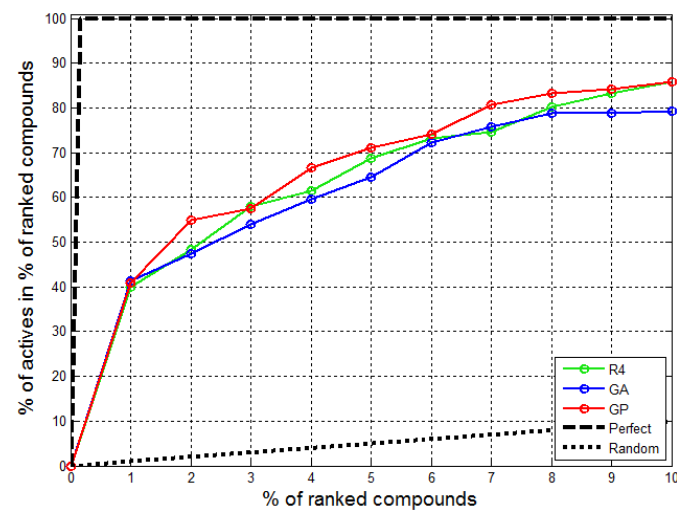


(b)

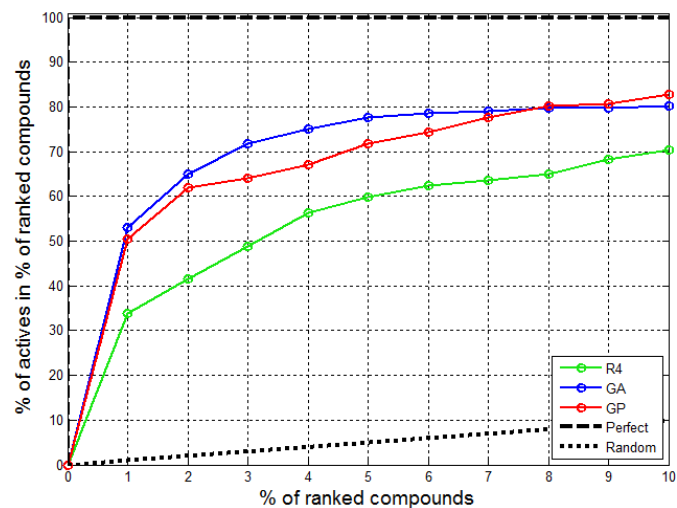
Figure 6.14: The Cumulative recall of active compounds plotted against the entire compound over 10 runs of the GP program: (a) GP instances for MDDR-based RNN activity class; (b) GP instances for MDDR-based COX activity class



(a)

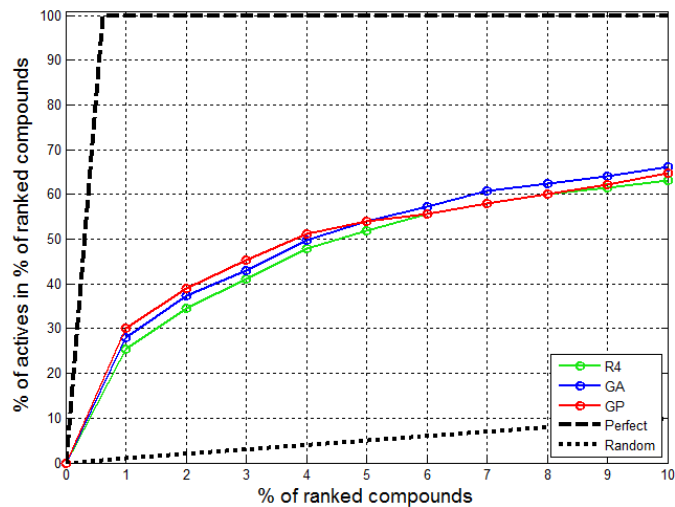


(b)

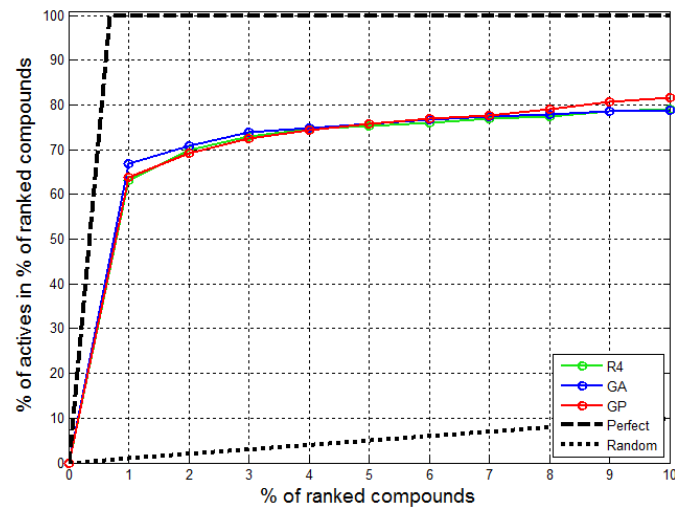


(c)

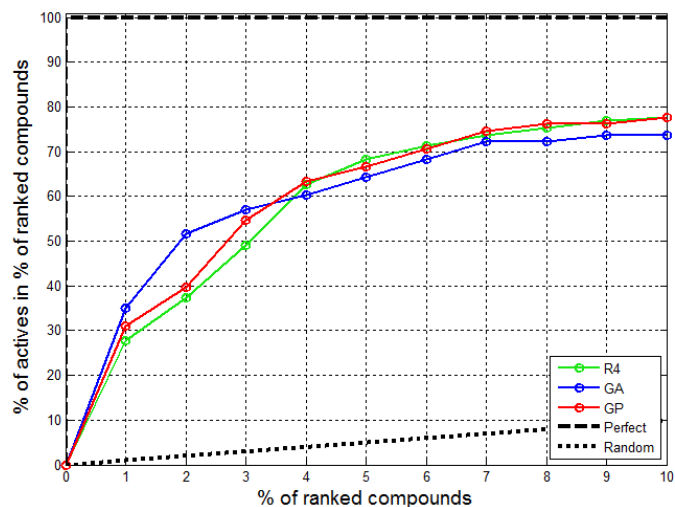
Figure 6.15: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for 5HT3 activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)



(b)



(c)

Figure 6.16: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for COX activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method

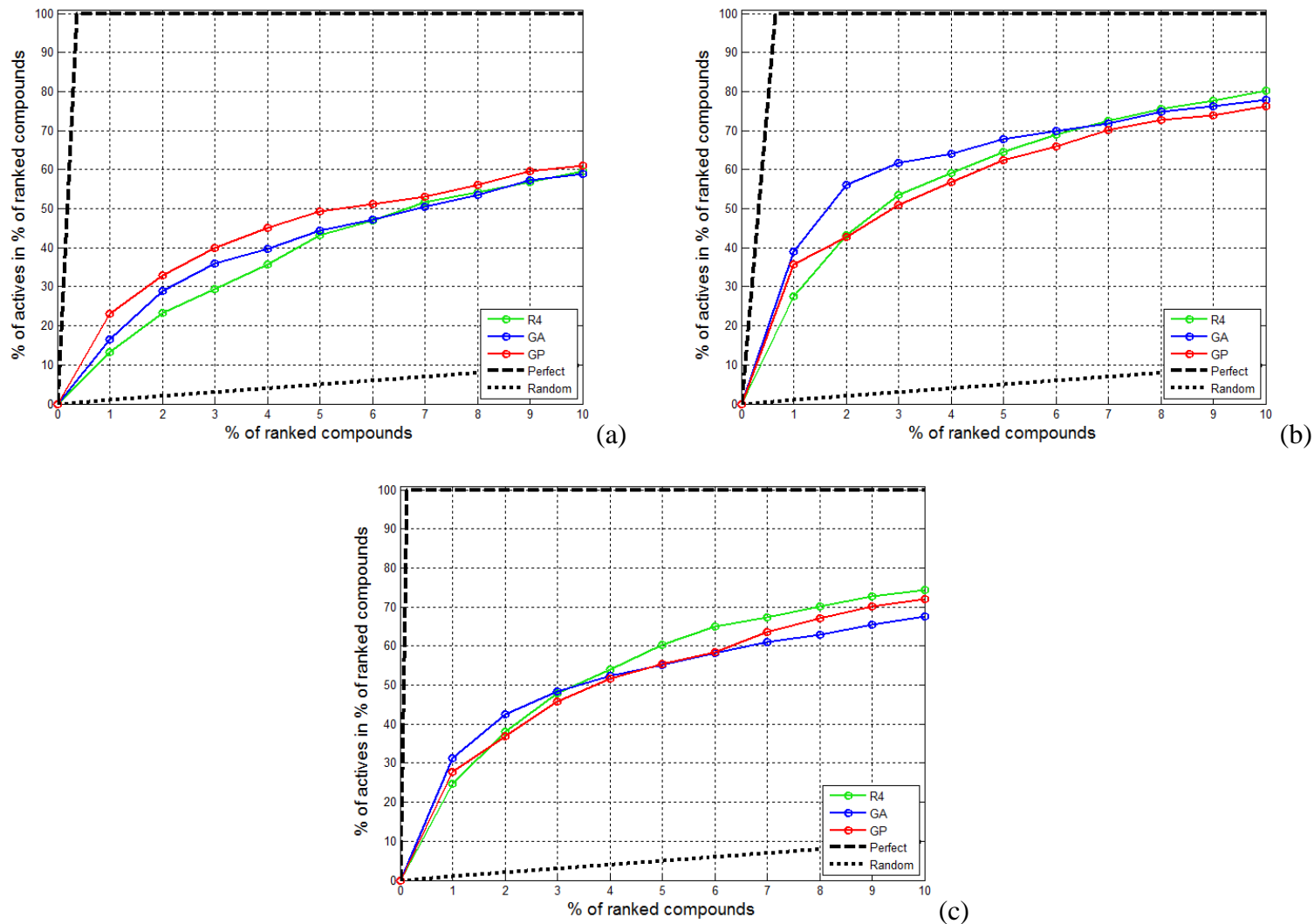
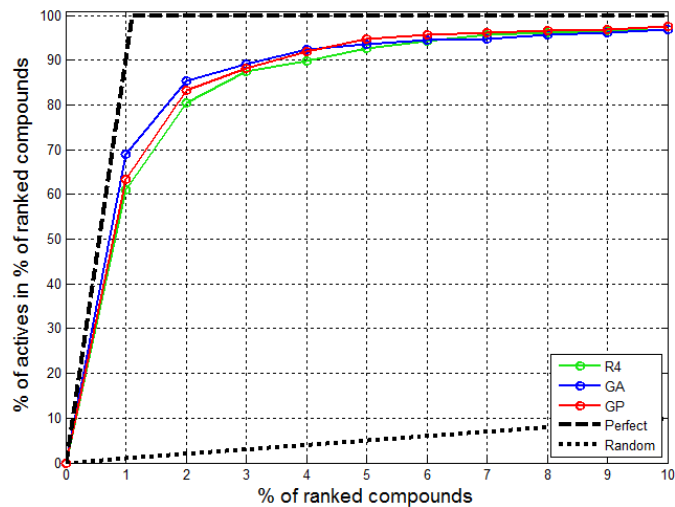
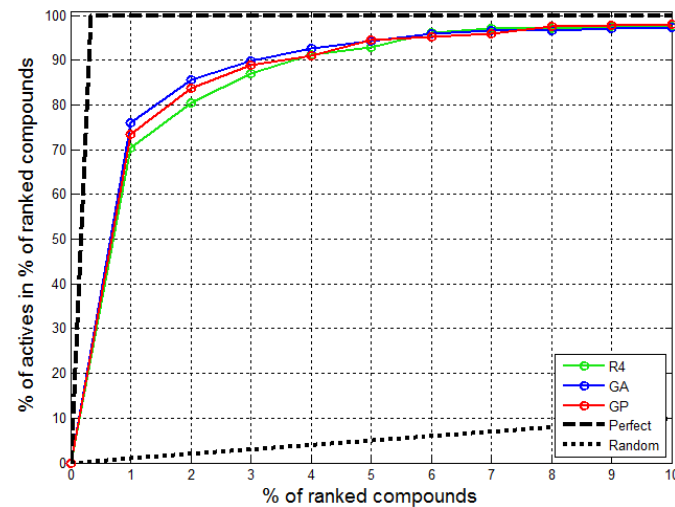


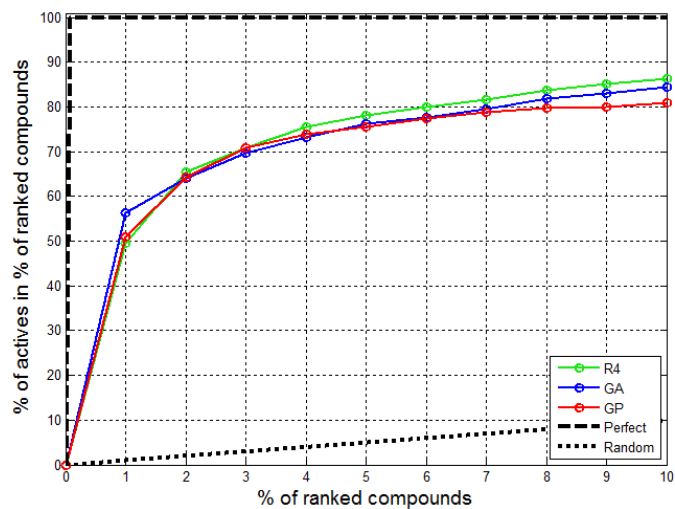
Figure 6.17: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for D2 activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

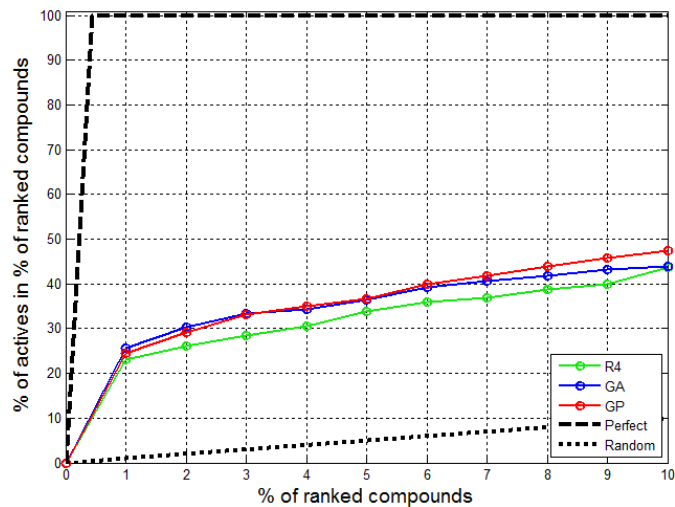


(b)

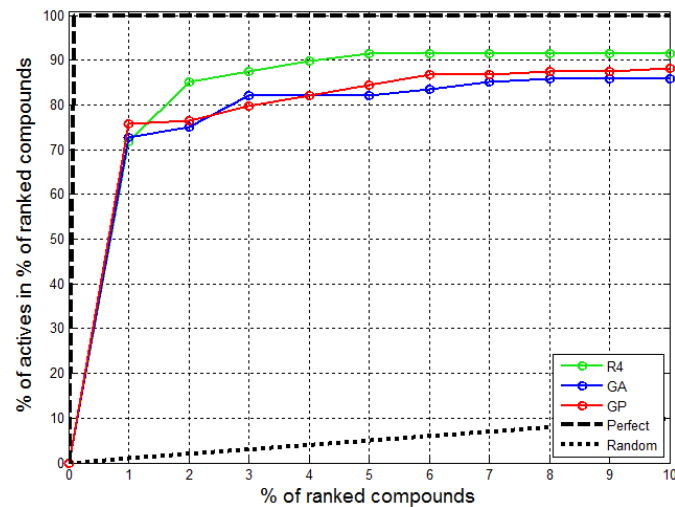


(c)

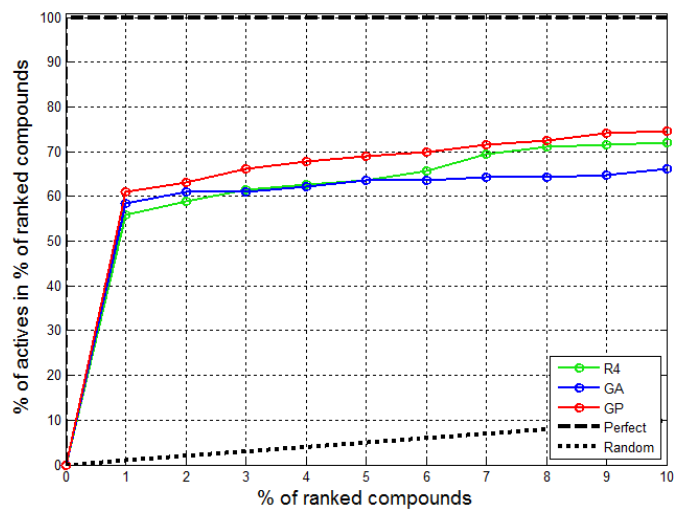
Figure 6.18: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for RNN activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)

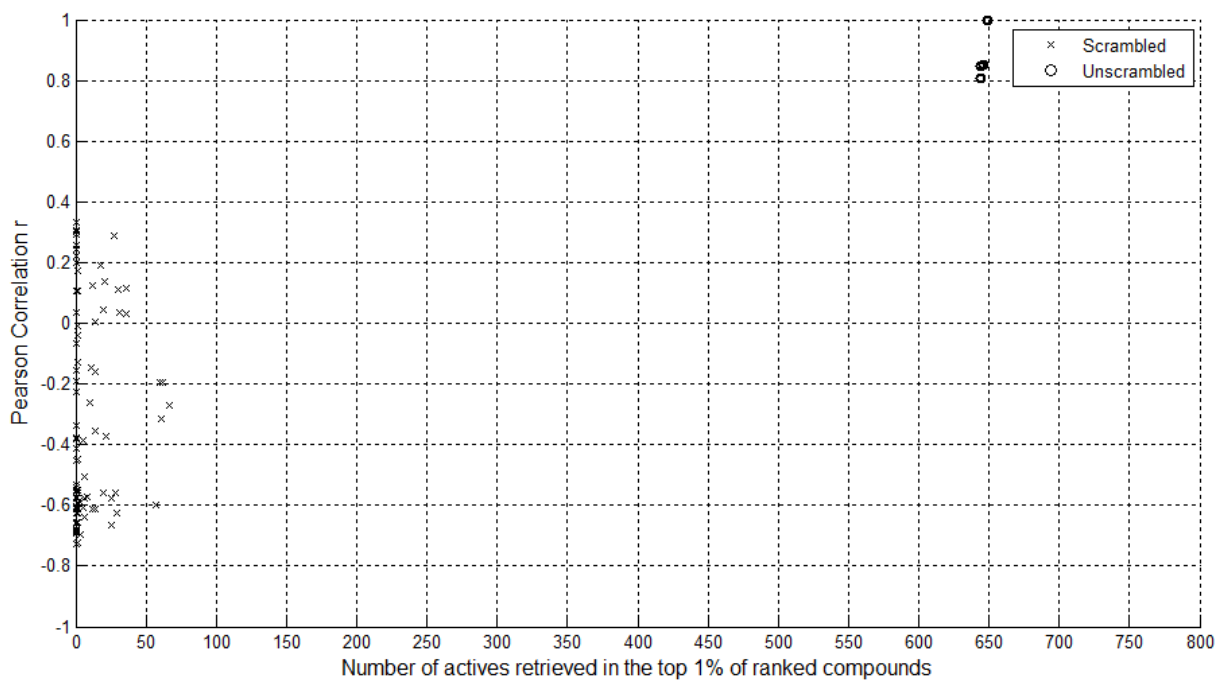


(b)

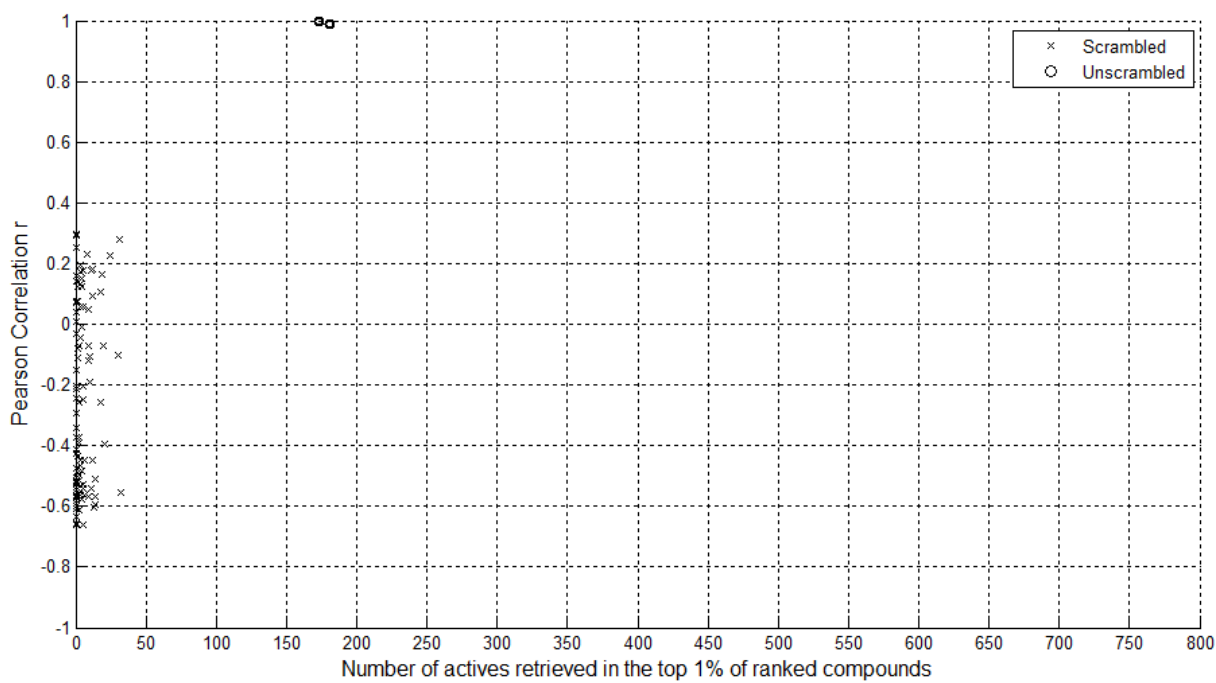


(c)

Figure 6.19: Cumulative recall plots of the GP-based SSA against SSA R4 and GA-based SSA for PKC activity class for the (a) MDDR (b) WOMBAT and (c) ChEMBL dataset. Plots represent the worst performing run of each method



(a)



(b)

Figure 6.20: Permutation plots (Y-randomisation) of the MDDR-based (a) RNN and (b) COX classes, with weights calculated and applied to non-permuted test sets

Original equation from GP:

$$\log \left(\frac{\sqrt{\sqrt{\sqrt{\frac{INACT}{N}} \times \frac{ACT}{NACT-ACT}}}}}{\frac{TOT}{N-TOT}} \right)$$

(a)



Simplified equation:

$$\log \left(\frac{ACT \sqrt[8]{\frac{INACT}{N}} \times (N - TOT)}{TOT \times (NACT - ACT)} \right)$$

(b)

Figure 6.21: Example of a GP-based SSA (a) Original equation and (b) The simplified equation using Wolfram Alpha expression simplifier online tool

Table 6.1: Summary of differences between GA and GP

	Genetic Algorithm	Genetic Programming
Inventor	John Holland	John R Koza
Typical Application domain	Combinatorial optimisation	Computer program / function design
Features	Attempt to find best solution by genetically breeding population of individual over a series of generations	Attempt to let computer solve problems without explicitly programmed, via design of complex algorithm genetically bred over a series of generations
Representation	Arrays of binary or string representation (other forms of representation possible)	Tree structure consisting of arithmetic, logical formula or primitive functions and programs, alongside terminal variables
Chromosome Size	Fixed	Tree in GP may vary in depth and width
Recombination	N-point or uniform	Exchange of sub-trees
Mutation	Bitwise bit-flipping with fixed probability	Random changes in trees
Survivor Selection	Generational placement	All children replace parents

Table 6.2: Top 1% active retrieval rates for the GP fitness function definition test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded

Fitness function	Enrichment factor of actives in the top 1%	
	Test Set	
Fitness function active rate	RNN	COX
Active rate of compounds in the top 1%	63.32	30.19
Active rate of compounds in the top 10%	47.98	28.78

Table 6.3: Terminal and function variable combinations used for chromosome initialisation.

(a) List of tested variable combinations defined for the GP chromosome terminal set; (b) List of tested operator functions for the GP chromosome function set

(a)

Combination	Terminals	
VARIABLES_A	$N, NACT, NINACT, TOT, ACT, INACT$	
VARIABLES_B	Terminal combination	Used by SSA equation
	ACT / TOT	SAF
	$TOT * NACT$	SAS
	$NACT / N$	SAS, Hodes'
	$TOT * NACT / N$	SAS, Avidon's
	$ACT - TOT * NACT / N$	SAS
	$ACT / NACT$	R1, R2
	TOT / N	R1
	$INACT / NINACT$	Mayer & Sens, R2
	$(NACT - ACT)$	Mayer & Sens, R3, R4
	$ACT / (NACT - ACT)$	R3, R4
	$N - TOT$	R3
	$TOT / (N - TOT)$	R3
	$NINACT - INACT$	R4
$INACT / (NINACT - INACT)$	R4	
$ACT / INACT$	WT1	
$(ACT - INACT) / TOT$	WT2	

(b)

Combination	Functions
FUNCTIONS_A	<i>plus, minus, multiplication, division, log, power-of-2</i>
FUNCTIONS_B	Mandatory <i>log</i> wrapping the whole GP equation

Table 6.4: Top 1% active retrieval rates for the GP terminal and function set combination test. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded

Terminal / Function set combination	Enrichment factor of actives in the top 1%	
	Test Set	
Combination set	RNN	COX
VARIABLES_A only	52.21	30.15
VARIABLES_A and VARIABLES_B	63.21	30.66
Log-function wrapping	RNN	COX
FUNCTIONS_A only	62.8	30.1
FUNCTIONS_A and FUNCTIONS_B	63.0	30.2

Table 6.5: Top 1% active retrieval rates for the GP's chromosome structure tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. Each parameter's tree depth and node size values obtained are listed as well. The GPs was performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters.

The best parameter value is shaded

Chromosome structure	Enrichment factor of actives in the top 1%		Final tree depth	
	Test Set		RNN	COX
Maximum tree depths	RNN	COX	RNN	COX
4	55.20	27.40	4	4
6	63.32	30.24	6	5
8	62.15	30.24	5	5
10	63.32	30.15	6	6

Maximum nodes	Enrichment factor of actives in the top 1%		Final tree nodes	
	RNN	COX	RNN	COX
20	61.60	30.60	15	17
25	62.80	30.80	21	17
30	63.20	31.20	25	18
35	62.40	30.80	22	21
40	63.20	30.60	18	20
50	62.10	29.70	26	18

Table 6.6: Top 1% active retrieval rates for the GP's population and generation based parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GAs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded

Population and Generation	Enrichment factor of actives in the top 1%	
	Test Set	
Population size	RNN	COX
100	60.20	29.25
200	62.87	30.94
300	62.84	30.80
500	62.77	30.80
Iteration	RNN	COX
50	61.80	29.70
100	62.53	30.13
200	62.87	30.94
300	62.87	30.94
500	62.87	30.94
Tree construction	RNN	COX
Grow	62.23	31.05
Full	61.04	30.24
Ramped half and half	61.87	30.94

Table 6.7: Top 1% active retrieval rates for the GP's elitism model parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded

Elitism model	Enrichment factor of actives in the top 1%	
	Test Set	
Population size	RNN	COX
0	60.10	28.90
1	63.32	30.24
2	63.82	31.64
3	62.87	30.94
5	62.50	30.94
10	61.70	29.80
199 (steady state, only 1 chromosome change per evolution running on 1000 maximum iterations)	45.21	15.50

Table 6.8: Top 1% active retrieval rates for the GP's evolution control parameter tests. Listed are test set enrichment values for MDDR's RNN and COX activity classes. GPs were performed on a training set of 10% active and inactive compounds, where the resultant weights are subsequently applied on the predicted test set of 90% active and inactive compounds. Each parameter was executed three times and the worst result is selected to represent the individual parameters. The best parameter value is shaded

Evolution control	Enrichment factor of actives in the top 1%	
	Test Set	
Parent selection	RNN	COX
Roulette wheel	62.80	31.40
Tournament	61.50	29.25
Random	62.10	30.80
Crossover rate	RNN	COX
0.80	61.50	30.80
0.85	62.10	30.45
0.90	63.40	31.20
1.00	62.80	31.20
Mutation rate	RNN	COX
0.00	60.40	29.90
0.01	61.70	30.85
0.05	61.70	29.80
0.10	62.50	30.94
0.20	63.20	31.20
0.50	62.80	30.80

Table 6.9: The top-ranked molecules in ten GP runs based on test set applied data, showing the occurrences of ranked active compounds based on GP run-1 that fall outside the top 1% in the other nine remaining GP runs using the (a) RNN and (b) COX activity classes in the MDDR dataset. The numbers in brackets show the number of actives actually retrieved in the top 1% for that particular GP-run

(a)										
Rank	GP Run									
	1	2	3	4	5	6	7	8	9	10
Top 1% (922)		0	18	18	18	16	0	0	25	18
	(649)	(649)	(646)	(646)	(646)	(644)	(649)	(649)	(644)	(646)

(b)										
Rank	GP Run									
	1	2	3	4	5	6	7	8	9	10
Top 1% (922)		20	0	20	20	0	0	20	20	20
	(173)	(181)	(173)	(181)	(181)	(173)	(173)	(181)	(181)	(181)

Table 6.10: Enrichment curve of actives count in the top 1% for ten GP runs of (a) Eleven activity classes in MDDR dataset; (b) Fourteen activity classes in WOMBAT dataset and; (b) Fifteen activity classes in ChEMBL dataset. Included are the mean and standard deviation for the Pearson correlation coefficients between the sets of 166 weights computed for each distinct pair of runs

Activity Class	GP Runs												Weight Pearson's r	
	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	σ	σ	Mean
	(a)													
5HT3	26.74	26.74	26.29	26.74	26.74	26.74	26.74	26.29	26.74	26.74	26.65	0.19	0.07	0.91
5HT1A	15.46	16.26	15.46	14.92	15.19	15.73	15.73	16.26	16.26	16.26	15.75	0.50	0.06	0.88
5HT	20.43	20.43	20.43	18.89	18.89	20.43	18.89	18.89	20.43	20.43	19.81	0.80	0.00	0.98
D2	21.63	23.03	23.03	23.03	23.03	23.03	21.63	21.63	21.63	23.03	22.47	0.73	0.00	0.98
RNN	63.82	63.82	63.52	63.52	63.52	63.32	63.82	63.82	63.32	63.52	63.60	0.20	0.02	0.91
AT1	48.53	48.53	48.53	48.41	48.53	48.53	48.53	48.41	48.53	48.53	48.50	0.05	0.31	0.75
THRM	43.71	44.95	44.12	43.98	44.12	44.12	44.95	44.95	44.95	44.95	44.48	0.51	0.01	0.99
SUBP	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	0.00	0.00	1.00
HIVP	40.30	40.30	39.85	39.85	40.00	40.00	40.30	40.00	39.85	40.00	40.04	0.19	0.03	0.93
COX	30.24	31.64	30.24	31.64	31.64	30.24	30.24	31.64	31.64	31.64	31.08	0.72	0.00	0.99
PKC	25.49	25.49	24.51	24.51	25.49	24.51	25.49	25.49	24.51	25.49	25.10	0.51	0.00	0.99

Activity	GP Runs												Weight Pearson's r		
	Class	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	σ	σ	Mean
							(b)								
5HT1A	48.97	48.97	48.97	48.97	48.97	48.78	48.97	48.97	48.97	48.97	48.97	48.95	0.06	0.39	0.72
5HT3	42.42	42.42	42.42	40.91	40.91	42.42	40.91	42.42	42.42	42.42	42.42	41.97	0.73	0.00	0.99
ACHE	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	0.00	0.00	1.00
AT1	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	0.00	0.00	1.00
COX	64.10	64.10	63.75	63.75	64.10	64.10	64.10	64.10	64.10	64.10	64.10	64.03	0.15	0.00	0.99
D2	36.02	35.78	35.78	36.02	35.78	35.78	36.02	35.78	36.02	36.02	36.02	35.90	0.13	0.05	0.92
FXA	40.63	40.63	39.84	39.84	39.84	40.63	40.63	39.84	40.63	39.84	39.84	40.24	0.42	0.00	0.99
HIVP	42.07	42.07	42.76	42.76	42.07	42.07	42.07	42.07	42.76	42.07	42.07	42.28	0.33	0.01	0.99
MMP	62.24	63.20	62.56	62.24	63.20	63.20	63.20	63.20	62.56	63.20	62.88	62.88	0.43	0.13	0.79
PDE	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	0.00	0.00	1.00
PKC	75.78	77.34	75.78	77.34	77.34	77.34	77.34	76.56	76.56	77.34	76.88	76.88	0.66	0.10	0.84
RNN	75.88	73.54	74.71	73.54	75.88	75.88	74.71	73.54	75.88	74.71	74.82	74.82	1.03	0.07	0.83
SUBP	47.01	47.01	47.01	47.01	45.62	47.01	47.01	47.01	45.62	47.01	46.73	46.73	0.59	0.01	0.99
THRM	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	0.00	0.00	1.00
							(c)								
5HT1A	33.11	32.43	33.11	33.11	33.11	32.43	31.91	32.43	31.91	33.11	32.67	32.67	0.50	0.00	0.92
5HT3	51.56	51.56	51.56	51.56	51.56	51.56	51.56	50.52	50.52	51.56	51.35	51.35	0.44	0.00	0.99
5HT	29.79	29.79	29.79	29.79	29.56	29.56	29.79	29.79	29.56	29.79	29.72	29.72	0.11	0.02	0.97
ACHE	29.92	28.87	29.92	29.92	28.87	29.92	29.92	29.92	28.87	28.87	29.50	29.50	0.54	0.00	0.99
AT1	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	0.00	0.00	1.00
COX	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	0.00	0.00	1.00
D2	28.23	27.81	27.81	28.23	28.23	28.23	27.81	28.23	28.23	27.81	28.06	28.06	0.22	0.00	0.99
FXA	39.13	39.13	39.28	39.28	39.28	39.28	39.28	39.28	39.28	39.28	39.25	39.25	0.06	0.00	0.99
HIVP	49.92	49.92	49.92	49.05	49.41	49.92	49.92	49.41	49.41	49.41	49.63	49.63	0.33	0.01	0.97
MMP	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	0.00	0.00	1.00
PDE	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	0.00	0.00	1.00
PKC	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	0.00	0.00	1.00
RNN	51.13	51.13	51.13	50.90	51.13	51.13	51.13	50.90	51.13	50.90	51.06	51.06	0.11	0.02	0.96
SUBP	70.08	70.08	69.16	70.08	70.08	70.08	70.08	70.08	70.08	70.08	69.99	69.99	0.29	0.01	0.99
THRM	37.93	37.93	35.15	35.15	35.15	37.93	35.15	37.93	37.93	37.93	36.82	36.82	1.44	0.02	0.90

Table 6.11: Screening results using the GP-based SSA and its comparison to the SSA R4 and GA-based SSA weighting scheme for the (a) MDDR; (b) WOMBAT and (c) ChEMBL datasets. The number of actives retrieved at the top 1% based on the worst performing GA runs is recorded for the calculation of Tanimoto coefficient and the *BemisMurckoAssemblies* based diversity analysis

Activity	Actives	NBC	Best run		Mean of ten runs		σ of ten runs		Worst run		Murcko scaffolds			Diversity rate		
			class	test set	SSA R4	GA	GP	GA	GP	GA	GP	SSA R4	GA	GP	SSA R4	GA
(a)																
5HT3	677	138	287	181	278.80	180.40	6.21	1.26	271	178	75	83	80	0.38	0.39	0.38
5HT1A	744	103	157	121	143.00	117.20	6.13	3.71	138	111	52	63	54	0.36	0.36	0.37
5HT	323	34	62	66	57.80	64.00	2.82	2.58	54	61	17	20	22	0.32	0.38	0.28
D2	356	47	70	82	61.90	80.00	3.35	2.58	59	77	24	30	43	0.29	0.31	0.31
RNN	1017	620	725	649	716.20	646.80	6.44	2.04	701	644	192	205	187	0.28	0.29	0.29
AT1	849	372	413	412	407.20	411.80	4.26	0.42	401	411	131	139	144	0.31	0.33	0.33
THRM	723	226	363	325	349.40	321.60	7.06	3.69	342	316	99	151	136	0.25	0.38	0.32
SUBP	1121	262	343	264	326.90	264.00	12.03	0.00	309	264	120	129	123	0.34	0.36	0.33
HIVP	675	226	334	272	317.50	270.30	15.42	1.25	284	269	106	115	143	0.33	0.40	0.36
COX	572	146	174	181	167.50	177.80	3.75	4.13	161	173	36	37	47	0.46	0.46	0.43
PKC	408	94	129	104	118.40	102.40	7.40	2.07	105	100	34	38	36	0.39	0.40	0.38

Activity	Actives	NBC	Best run		Mean of ten runs		σ of ten runs		Worst run		Murcko scaffolds			Diversity rate		
class	test set	SSA R4	GA	GP	GA	GP	GA	GP	GA	GP	SSA R4	GA	GP	SSA R4	GA	GP
5HT1A	533	249	310	261	296.60	260.90	8.72	0.32	284	260	58	64	61	0.34	0.35	0.33
5HT3	198	79	98	84	86.10	83.10	4.56	1.45	82	81	23	25	25	0.33	0.40	0.33
ACHE	453	218	240	224	231.60	224.00	5.27	0	223	224	67	69	70	0.30	0.31	0.30
AT1	652	514	543	515	526.10	515.00	7.34	0	518	515	105	106	105	0.37	0.37	0.37
COX	869	549	591	557	585.90	556.40	3.87	1.26	581	554	32	39	35	0.36	0.36	0.36
D2	819	225	350	295	333.70	294.00	9.21	1.05	319	293	61	65	65	0.32	0.34	0.31
FXA	758	288	377	308	346.20	305.00	14.23	3.16	333	302	75	86	91	0.31	0.45	0.44
HIVP	1015	398	582	434	543.60	429.10	25.37	3.38	503	427	127	143	139	0.35	0.37	0.32
MMP	625	362	400	395	396.3	393.00	3.92	2.67	390	389	86	97	97	0.23	0.24	0.23
PDE	536	239	277	241	264.10	241.00	6.15	0	259	241	84	86	83	0.48	0.58	0.52
PKC	128	92	99	99	94.7	98.40	1.77	0.84	93	97	15	15	15	0.40	0.46	0.30
RNN	427	301	353	324	336.30	319.50	8.91	4.38	325	314	73	76	82	0.32	0.34	0.29
SUBP	502	217	264	236	247.10	234.60	9.33	2.95	237	229	52	54	53	0.31	0.38	0.39
THRM	379	200	223	209	212.70	209.00	6.27	0	206	209	73	77	77	0.41	0.43	0.45

Activity class	Actives test set	NBC	Best run		Mean of ten runs		σ of ten runs		Worst run		Murcko scaffolds			Diversity rate			
		SSA R4	GA	GP	GA	GP	GA	GP	GA	GP	SSA R4	GA	GP	SSA R4	GA	GP	
								(c)									
5HT1A	1335	383	544	442	530.30	436.10	7.90	6.72	522	426	114	141	127	0.36	0.38	0.37	
5HT3	192	65	111	99	105.70	98.60	3.71	0.84	102	97	14	25	21	0.34	0.38	0.37	
5HT	2202	540	782	656	760.10	654.50	23.58	2.42	707	651	107	117	120	0.51	0.52	0.50	
ACHE	665	168	249	199	244.50	196.20	3.50	3.61	240	192	83	99	93	0.40	0.42	0.47	
AT1	95	40	80	76	77.30	76.00	2.11	0	74	76	8	22	24	0.28	0.28	0.29	
COX	125	35	52	39	47.90	39.00	3.57	0	44	39	9	12	11	0.73	0.75	0.73	
D2	1672	413	558	472	532.00	469.20	13.98	3.61	522	465	120	154	131	0.31	0.31	0.30	
FXA	1352	467	654	531	640.80	530.60	7.28	0.84	634	529	141	163	162	0.37	0.37	0.40	
HIVP	1941	903	1324	969	1258.80	963.30	24.45	6.36	1239	952	264	324	292	0.42	0.51	0.36	
MMP	356	208	254	260	244.90	260.00	8.03	0	234	260	57	62	76	0.44	0.45	0.44	
PDE	229	63	100	68	92.10	68.00	5.70	0	85	68	19	25	23	0.54	0.60	0.52	
PKC	190	106	113	116	111.40	116.00	1.51	0	109	116	23	26	27	0.25	0.25	0.27	
RNN	884	438	508	452	500.70	451.40	4.00	0.97	496	450	68	69	69	0.36	0.38	0.36	
SUBP	762	476	546	534	534.00	533.30	8.00	2.21	522	527	104	103	109	0.37	0.43	0.39	
THRM	754	192	362	286	352.60	277.60	9.78	10.84	339	265	90	120	109	0.32	0.39	0.37	

Table 6.12: Kendall's W analysis for the top 1% actives retrieved of the ranking for (a) Eleven activity classes in MDDR, (b) Fourteen activity classes in WOMBAT, and (c) Fifteen activity classes of the ChEMBL database

(a)

Weighting Schemes	Activity Class											Mean Rank	Rank Position
	5HT3	5HT1A	5HT	D2	RNN	AT1	THRM	SUBP	HIVP	COX	PKC		
GA	2.00	2.00	1.00	1.00	2.00	1.00	2.00	2.00	2.00	1.00	2.00	1.64	1
GP	1.00	1.00	2.00	2.00	1.00	2.00	1.00	1.00	1.00	2.00	1.00	1.36	2
SSA R4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3

(b)

Weighting Schemes	Activity Class														Mean Rank	Rank Position
	5HT1A	5HT3	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
GA	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00	2.00	1.00	1.79	1
GP	1.00	1.00	2.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	2.00	1.14	2
SSA R4	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	3

(c)

Weighting Schemes	Activity Class															Mean Rank	Rank Position
	5HT1A	5HT3	5HT	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
GA	2.00	2.00	2.00	2.00	1.00	2.00	2.00	2.00	2.00	1.00	2.00	1.00	2.00	1.00	2.00	1.73	1
GP	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00	1.00	2.00	1.00	1.27	2
SSA R4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3

Table 6.13: The worst and best performing GP equations selected from the 10 runs of each activity class, based on the (a) MDDR, (b) WOMBAT, and (c) ChEMBL18 datasets. All equations are simplified from its original form

(a)

Activity Class	Worst performing GP equation	Best performing GP equation
5HT	$\log\left(\frac{\text{ACT NINACT}}{\text{INACT}(\text{NACT} - \text{ACT})} - \frac{\text{ACT}}{\text{NACT}} + \frac{\text{NACT}}{\text{N}}\right)$	$\log\left(\sqrt[8]{\frac{\text{ACT}}{\text{NACT} - \text{ACT}} + \frac{\text{INACT}}{\text{NINACT}} + \frac{\text{N}}{\text{NACT TOT}}}\right)$
5HT1A	$\log\left(\frac{\text{N}\left(\frac{\text{ACT}}{\text{NACT}}\right)^{3/4}}{\text{TOT}}\right)$	$\log\left(\frac{\text{ACT}\left(\sqrt[4]{\frac{\text{ACT}}{\text{INACT}}} - 1\right) - \text{NACT}\sqrt[4]{\frac{\text{ACT}}{\text{INACT}}}}{(\text{ACT} - \text{NACT})\sqrt{\frac{\text{TOT}}{\text{N-TOT}}}}\right)$
5HT3	$\frac{1}{2} \log\left(\frac{\text{ACT N}\sqrt{\text{NINACT}}}{\text{NACT}^2 \text{TOT}}\right)$	$\log\left(\sqrt{\frac{\text{ACT} - \text{INACT}}{\text{TOT}}} + \frac{\text{ACT}}{\text{INACT}} + \frac{\text{NINACT}}{\text{N}}\right)$
AT1	$\log\left(\frac{\text{ACT NINACT}}{\text{INACT}(\text{NACT} - \text{ACT})} - \frac{\text{ACT}}{\text{NACT}} + \frac{\text{NACT}}{\text{N}}\right)$	$\log\left(\frac{\text{TOT}(\text{ACT} - \text{NACT})\sqrt[8]{\frac{\text{ACT} - \text{INACT}}{\text{TOT}}} + \text{ACT}(\text{TOT} - \text{N})}{\text{TOT}(\text{ACT} - \text{NACT})}\right)$

COX	$\log \left(\frac{\text{INACT NACT} \sqrt{\frac{\text{ACT}}{(\text{ACT} - \text{NACT})^2} + \text{ACT NINACT}}}{\text{INACT NACT}} \right)$	$\log \left(\frac{(\text{N} - \text{TOT}) \left(\frac{\text{ACT N}}{\text{TOT} (\text{NACT} - \text{ACT})} \right)^{3/2}}{\text{N}} \right)$
D2	$\frac{1}{2} \log \left(\frac{\text{ACT}}{\text{NACT} - \text{ACT}} \right) - \frac{1}{4} \log \left(\frac{\text{ACT TOT}}{\text{NACT} (\text{N} - \text{TOT})} \right)$	$\log \left(\frac{\text{ACT NINACT}}{\text{INACT} (\text{NACT} - \text{ACT})} - \sqrt{\frac{\text{ACT}}{\text{NACT} - \text{ACT}} - \frac{\text{TOT}}{\text{N}}} \right)$
HIVP	$\log \left(\sqrt{\frac{\text{ACT N}}{\text{INACT} (\text{NACT} - \text{ACT})}} - \sqrt[4]{\frac{\text{N}}{\text{NACT TOT}}} \right)$	$\log \left(- \frac{\text{ACT NACT NINACT}}{(\text{ACT} - \text{NACT}) \left(\text{ACT NINACT} \sqrt{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}} + \text{INACT NACT} \right)} \right)$
PKC	$\log \left(\sqrt{\sqrt{\text{ACT} - \text{INACT}} + \sqrt{\frac{\text{TOT} (\text{NINACT} - \text{INACT})}{\text{INACT} (\text{N} - \text{TOT})}}} \right)$	$\log \left(\sqrt{\sqrt{\frac{\text{ACT} - \text{INACT}}{\text{TOT}}} + \frac{\text{TOT} (\text{NINACT} - \text{INACT})}{\text{INACT} (\text{N} - \text{TOT})}} \right)$
RNN	$\log \left(\frac{\text{ACT} \sqrt[8]{\frac{\text{INACT}}{\text{N}}} (\text{N} - \text{TOT})}{\text{TOT} (\text{NACT} - \text{ACT})} \right)$	$\log \left(\frac{\text{INACT} \sqrt[8]{\frac{\text{ACT} (\text{INACT} - \text{NINACT})}{\text{INACT} (\text{ACT} - \text{NACT})}}}{\text{TOT}} \right)$

SUBP	$\log \left(\frac{N \sqrt{-\frac{\frac{ACT}{ACT-NACT} \sqrt{\frac{INACT}{INACT-NINACT}} (INACT-NINACT)}}{INACT}}}{NINACT} \right)$	
THRM	$\log \left(-\frac{ACT}{(ACT-NACT) \sqrt{\frac{(N-TOT) \sqrt{\frac{NACT}{NINACT}} + TOT}{N-TOT}}} \right)$	$\log \left(\frac{ACT}{NACT-ACT} - \sqrt{\frac{TOT}{N-TOT} + 1} \right)$

(b)

Activity Class	Worst performing GP equation	Best performing GP equation
5HT1A	$\log \left(\sqrt[4]{\frac{\text{ACT}}{\text{NACT}}} + \sqrt{\frac{\text{N}}{\text{NACT TOT}}} \right)$	$\log \left(\sqrt[4]{\frac{\text{ACT}}{\text{NACT}}} + \sqrt{\frac{\text{N}}{\text{NACT TOT}}} \right)$
5HT3	$\frac{1}{2} \log \left(\sqrt{\frac{\sqrt{\frac{\text{ACT}}{\text{NACT}-\text{ACT}}} (\text{N} - \text{TOT})}{\text{TOT}}} + \frac{\text{NACT}}{\text{NINACT}}} \right)$	$\log \left(\frac{\text{N} \sqrt{\frac{\text{ACT}}{\text{NACT}-\text{ACT}}} (\text{NINACT} - \text{INACT})}{\text{INACT TOT}} \right)$
ACHE	$\log \left(\frac{\text{NACT TOT} \sqrt{\frac{\text{ACT}}{\text{NACT}-\text{ACT}}} + \text{N} \sqrt{-\frac{\text{TOT}}{\text{TOT}-\text{N}}}}{\text{NACT TOT} \sqrt{\frac{\text{TOT}}{\text{N}-\text{TOT}}}} \right)$	
AT1	$\log \left(\frac{\text{ACT} (\text{INACT} - \text{NINACT})}{\text{INACT} (\text{ACT} - \text{NACT})} \right)$	$\log \left(\frac{\text{ACT} (\text{INACT} - \text{NINACT})}{\text{INACT} (\text{ACT} - \text{NACT})} \right)$
COX	$\log \left(\frac{\text{ACT} \sqrt{\text{NINACT}} \sqrt{-\frac{\text{ACT}}{\text{ACT}-\text{NACT}}}}{\text{TOT}} \right)$	$\log \left(-\frac{\text{ACT} (-\text{ACT} + \text{NACT} + \text{TOT})}{\text{TOT} (\text{ACT} - \text{NACT}) \sqrt{-\frac{\text{TOT}}{\text{TOT}-\text{N}}}} \right)$

D2	$\frac{1}{2} \log \left(\sqrt[4]{\frac{\text{ACT}}{\text{INACT}}} + \frac{\text{ACT N}}{\text{NACT TOT}} \right)$	$\log \left(\frac{\text{ACT N INACT}}{\text{INACT (NACT - ACT)}} - \sqrt[4]{\frac{\text{ACT}}{\text{TOT}}} \right)$
FXA	$\log \left(\frac{\text{ACT}}{\text{NACT - ACT}} + \sqrt[4]{\frac{\text{ACT}}{\text{NACT - ACT}}} - \sqrt{\frac{\text{TOT}}{\text{N}}} \right)$	$\log \left(\frac{\text{ACT}}{\text{NACT - ACT}} + \sqrt{\frac{\text{ACT}}{\text{NACT}}} - \frac{\text{INACT}}{\text{N}} \right)$
HIVP	$\log \left(\sqrt{\sqrt[4]{\frac{\text{ACT}}{\text{NACT - ACT}}} - \frac{\text{TOT}}{\text{N - TOT}}} + \frac{\text{ACT}}{\text{NACT - ACT}} \right)$	$\log \left(\frac{\text{ACT (INACT - N INACT)} \sqrt{\frac{\text{ACT}}{\text{NACT - ACT}} + \frac{\text{INACT}}{\text{TOT}}}}{\text{INACT (ACT - NACT)}} \right)$
MMP	$\frac{1}{2} \log \left(\frac{\text{ACT N} + \text{N INACT TOT}}{\text{N TOT}} \right)$	$\frac{1}{2} \left(\log \left(\frac{\text{ACT}}{\text{NACT}} - \frac{\text{TOT}}{\text{N - TOT}} \right) - \log(\text{ACT - INACT}) \right)$
PDE	$\log \left(\frac{\text{ACT (N - TOT)}}{(\text{ACT - INACT}) (\text{ACT - NACT}) + \text{ACT} \sqrt{\text{N INACT}}} \right)$	
PKC	$\log \left(\frac{\text{ACT} \sqrt[8]{\frac{\text{ACT}}{\text{TOT}}}}{\text{NACT - ACT}} \right)$	$\log \left(\frac{\sqrt{-\frac{\text{ACT NACT}}{\text{ACT - NACT}} (\text{N - TOT}) - \text{TOT}}}{\text{N - TOT}} \right)$
RNN	$\log \left(-\frac{\text{ACT}}{\text{INACT}} + \sqrt{\frac{\text{ACT}}{\text{NACT - ACT}}} + \frac{\text{N}}{\text{NACT TOT}} \right)$	$\log \left(\frac{\text{ACT}}{\text{NACT - ACT}} + \sqrt[4]{\frac{\text{N}}{\text{NACT TOT}} - \frac{\text{TOT}}{\text{N}}} \right)$

SUBP	$\log \left(\frac{\text{ACT N} \left(\text{ACT} - \sqrt{-\frac{\text{INACT}}{\text{INACT} - \text{NINACT}}} - \text{NACT} \right)}{\text{NACT TOT} (\text{ACT} - \text{NACT})} \right)$	$\log \left(\frac{\text{ACT} \left(\frac{\text{ACT}^2}{(\text{ACT} - \text{NACT})^2} + \frac{\text{NINACT}}{\text{NACT}} \right)}{\text{TOT}} \right)$
THRM	$\log \left(\left(\frac{\text{ACT}}{\text{NACT} - \text{ACT}} + \sqrt[4]{\text{ACT}} \right) \sqrt{\frac{\text{N}}{\text{NACT TOT}}} \right)$	

(c)

Activity Class	Worst performing GP equation	Best performing GP equation
5HT	$\log\left(\frac{\text{ACT}(\text{NINACT} - \text{INACT})}{\text{INACT NACT}} + \frac{\text{INACT}}{N} + \sqrt{\frac{\text{NACT}}{\text{NINACT}}}\right)$	$\log\left(\frac{\text{ACT}\left(\frac{\text{NACT}(\text{ACT} - \text{INACT} + \text{NINACT})}{\text{NINACT}} + \frac{\text{NINACT}}{\text{NACT}}\right)}{\text{INACT}}\right)$
5HT1A	$\log\left(\frac{\text{ACT}(\text{ACT NACT} + \text{NINACT})}{\text{NACT TOT}}\right)$	$\log\left(\frac{\text{ACT}}{(\text{ACT} - \text{NACT}) \sqrt{\frac{\text{INACT}\left(-\sqrt{\frac{\text{NACT}}{N}} - 1\right) + \text{NINACT}\sqrt{\frac{\text{NACT}}{N}}}{\text{INACT} - \text{NINACT}}}}\right)$
5HT3	$\log\left(\frac{(\text{ACT NINACT} - \text{INACT}) \sqrt{\frac{\text{INACT}(\text{NACT} - \text{ACT}) - \text{ACT NINACT}}{\text{NINACT}(\text{ACT} - \text{NACT})}}}{\text{NINACT}}\right)$	$\log\left(\frac{\text{ACT}(N - \text{TOT}) \sqrt{\frac{\text{ACT}(-N - \text{TOT}) + \text{NACT TOT}}{N(\text{ACT} - \text{NACT})}}}{\text{TOT}}\right)$
ACHE	$\frac{1}{2} \log\left(\frac{\text{ACT}(\text{NINACT} - \text{INACT})}{\text{INACT} \sqrt[4]{\frac{\text{ACT}}{\text{NACT}} (\text{NACT} - \text{ACT})}}\right)$	$\log\left(\frac{\text{ACT}(\sqrt{\text{INACT NACT}} + \text{NINACT})}{\text{NACT TOT}}\right)$
AT1	$\frac{1}{2} \log\left(\frac{\text{ACT}(-\text{NACT} - \text{NINACT}) + \text{NACT}^2}{\text{NINACT}(\text{ACT} - \text{NACT})}\right)$	

COX	$\log \left(\frac{(N - \text{TOT}) \sqrt{\text{ACT} \left(\frac{1}{\text{NACT} - \text{ACT}} - \frac{1}{N} \right)}}{\text{TOT}} \right)$	
D2	$\log \left(\frac{\text{ACT}}{(\text{ACT} - \text{NACT}) \sqrt{\frac{\text{TOT} \sqrt{\frac{\text{ACT}}{\text{NACT}}}}{N - \text{TOT}}}} \right)$	$\log \left(\text{ACT} \left(\frac{\frac{N^2}{\text{TOT} - 2N} + N}{\text{TOT} (\text{NACT} - \text{ACT})} + \frac{1}{\text{NACT}} \right) \right)$
FXA	$\log \left(\frac{\text{ACT} N \sqrt[4]{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}}}{\text{NACT} \text{TOT}} \right)$	$\log \left(\frac{\text{ACT} N \sqrt[4]{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}}}{\text{INACT} \text{NACT}} \right)$
HIVP	$\log \left(\sqrt{\frac{\text{ACT}}{\text{NACT}} + \frac{\text{INACT}}{\text{INACT} - \text{NINACT}}} + \sqrt[4]{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}} \right)$	$\frac{1}{2} \log \left(\sqrt{\frac{\text{ACT}}{\text{TOT}}} \left(\frac{\text{ACT}}{\text{NACT} - \text{ACT}} + \sqrt{\frac{\text{NINACT}}{\text{NACT}}} \right) \right)$
MMP	$\log \left(\frac{\text{ACT} N \sqrt{-\frac{\text{ACT}}{\text{ACT} - \text{NACT}} + \text{INACT} \text{NINACT}}}{\text{INACT} N} \right)$	$\frac{1}{2} \log \left(\frac{\text{ACT} \sqrt{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}}}{\text{TOT}} + \frac{\text{NINACT}}{N} \right)$
PDE	$\frac{1}{2} \left(\log \left(\frac{\text{TOT}}{N - \text{TOT}} \right) - \log \left(\frac{\text{INACT}}{\text{NINACT} - \text{INACT}} \right) \right)$	

PKC	$\log \left(\frac{\text{ACT} \left(N \left(\sqrt{-\frac{\text{ACT}}{\text{ACT}-\text{NACT}}} + 1 \right) + \text{TOT} \sqrt{-\frac{\text{ACT}}{\text{ACT}-\text{NACT}}} - \text{NACT TOT} \sqrt{-\frac{\text{ACT}}{\text{ACT}-\text{NACT}}} \right)}{N (\text{ACT} - \text{NACT})} \right)$	$\log \left(2 \sqrt{\frac{\text{ACT}}{\text{NACT} - \text{ACT}}} - \sqrt{\frac{\text{INACT}}{\text{NINACT}}} \right)$
RNN	$\log \left(\frac{\text{ACT} (\text{INACT} - \text{NINACT})}{\text{INACT} (\text{ACT} - \text{NACT})} - \sqrt[4]{\frac{\text{NACT}}{\text{NINACT}}} \right)$	$\log \left(\frac{\text{ACT} + \sqrt{\frac{\text{ACT}}{\text{NACT}}} (N - \text{TOT})}{\text{TOT} (\text{NACT} - \text{ACT})} \right)$
SUBP	$\log \left(\frac{\left(\frac{\text{ACT}}{\text{NACT} - \text{ACT}} \right)^{3/4} (N - \text{TOT})}{\text{TOT}} \right)$	$\log \left(\frac{(N - \text{TOT}) \sqrt{-\frac{\text{ACT} \sqrt{\frac{\text{TOT}}{N - \text{TOT}}}}{\text{ACT} - \text{NACT}}}}{\text{TOT}} \right)$
THRM	$\log \left(\frac{\text{ACT} (\text{NINACT} - \text{INACT}) \left(\frac{\text{ACT}}{\text{NACT} - \text{ACT}} + \sqrt{\frac{N}{\text{NACT TOT}}} \right)}{\text{INACT NACT}} \right)$	$\log \left(\frac{\text{ACT}^2 (N - \text{TOT})}{\text{NACT TOT} (\text{NACT} - \text{ACT})} + \sqrt{\frac{N}{\text{NACT TOT}}} \right)$

Table 6.14: GP run-time benchmark at different iterations using the 10% training set of the RNN activity class, based on the (a) MDDR, (b) WOMBAT and (c) ChEMBL databases. Parameterisation of the GP is based on the final chosen ones as in Section 6.6.1.6, among them the population of 200 chromosomes, and a maximum iteration 200 evolutions

(a)

Machine	Time (seconds) at GP iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	2.7	289.4	532.5	2.66
WKST_01	3.25	339.1	670.7	3.35
WKST_02	3.50	371.2	721.4	3.60

Based on 10,254 compounds in training set

(b)

Machine	Time (seconds) at GP iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	2.80	289.5	578.4	2.89
WKST_01	3.70	389.2	780.5	3.90
WKST_02	4.1	431.2	834.3	4.17

Based on 13,812 compounds in training set

(c)

Machine	Time (seconds) at GP iteration			Average runtime per iteration (seconds)
	1	100	200	
SERVER	37.2	3928.5	7621.5	38.10
WKST_01	<i>Insufficient memory issue, not executable</i>			
WKST_02	48.1	4902.0	9680.5	48.40

Based on 135,267 compounds in training set

Chapter 7

Investigations Into The Application of Data Fusion

7.1 Introduction

Both the GA and GP methods for SSA were extensively experimented in the previous chapters and improvements in the number of active retrieved were reported. This chapter further investigates the application of data fusion for improvements in retrieval performance utilising the multiple runs information of the GA and GP-based SSA. The chapter presents a brief background of data fusion and the experiment details in order to implement the fusion of SSA data. Results of the experiments are analysed and discussed in terms of any observed improvement in the predictive performance of the fused screening method.

7.2 Data fusion

Data fusion is a method of combining the information gained from different sensors to achieve an effective or improved decision, compared to when only a single sensor is considered (Hall & McMullen, 2004). This method can be utilised for ligand-based virtual screening. The sensors to be combined are used as functions that score molecules in a database on their likelihood of exhibiting some required biological activity. The combination of different sources of information is already practiced in most human daily activities, such as in decision-making processes. A simple example is the use of different sensors in our everyday lives that include our sense of smell, taste, feeling, hearing and seeing. In a more practical sense, for instance, a manager considering at hiring a new employee makes informed decisions based on the different traits of the candidate, such as their skills, experience and communication abilities. These traits collectively produce a decision about the candidate's eligibility to be hired. Data fusion is increasingly used to combine the outputs of different types of digital or analogue sensors.

Data fusion has been successfully used in different fields, such as medicine, defence and information retrieval. The findings on combining more than one query in the field of information retrieval were first discussed by Belkin, Kantor, Fox and Shaw (1995). The study on data fusion was carried out in two different projects, at Rutgers University and the Virginia

Technology Institute. Together, these projects found that fusing the multiple queries is far more effective in increasing search performance, yielding better retrieval rates than using a single query.

In virtual screening, many studies related to data fusion have been carried out, especially regarding similarity searching. Similarity search is based on three main components: the molecule representation used to describe the molecular structures, the weighting scheme used to compute the score of a particular compound structure to produce compound rankings, and the similarity coefficient used to calculate the degree of similarity between the reference molecule and the database molecules. Essentially, data fusion in similarity searching can be further divided into similarity fusion and group fusion. Similarity fusion is the combination of scores gathered from multiple similarity measures by using a single reference structure for searching a chemical database (Whittle, Gillet, Willett, Alex & Loesel, 2004). For instance, the data fusion ranking is obtained by combining three rankings from different similarity coefficients, for example Tanimoto, Dice and Cosine. Several studies on similarity fusion were carried out by fusing different similarity coefficients in a similarity search (Sheridan & Kearsley, 2002; Ginn, Willett & Bradshaw, 2000; Salim, Holliday & Willett, 2003). The group fusion approach fuses rankings produced from different reference structures by using the same similarity coefficient and molecular representation (Hert et al., 2004b). Group fusion can utilise either similarity scores or rankings (Willett, 2013). For instance, assuming one type of 2D descriptor such as the MDL fingerprints, the similarities between reference structure and other structures in the database are measured using the Tanimoto coefficient. They are then ranked in descending order based on their similarity score.

Comparing the two fusion techniques, similarity fusion tends to perform better than group fusion when the actives are strongly clustered in structural space (Whittle, Gillet, Willett & Loesel, 2006). By contrast, group fusion is best employed when the actives are structurally diverse (Hert et al., 2006). Numerous studies have compared these two data fusion techniques in similarity searching. Other studies have found that group fusion is effective as a general approach in similarity searching (Chen et al., 2010, Whittle et al., 2006; Williams, 2006; 2006; Hert et al., 2004a, Hert et al., 2004b). Based on the encouraging results obtained using GA and GP in the previous chapters, the application of data fusion to the GP-based SSA and GA-based SSA weighting schemes are examined in this chapter in order to enhance the retrieval performances of 2D-based fingerprint predictive method. Extensive studies on data

fusion have been carried out on similarity-based rankings, but there is still a lack of findings on data fusion using genetic algorithm techniques in chemoinformatics.

7.3 Experimental details

7.3.1 Datasets

For this experiment, the available ten runs of the GA-and-GP-based SSA generated in the previous chapters, and for each class in the MDDR, WOMBAT and ChEMBL databases were used. Tables 5.10 and 6.10 represent the number of actives retrieved in the top 1% of the ten GA and GP runs respectively. Alternatively, their resulting enrichment factor of actives are also summarised in Table 5.11 and 6.11.

7.3.2 Fusion rules

In order to perform data fusion, it was necessary to extract the ranking output of the ten runs (of the GA and the GP) for each activity class to be fused. Five types of fusion rules, namely the SUM, MAX, MED, MIN and RKP rules were identified. Most rules were first discussed by Belkin, Kantor, Fox and Shaw (1995); however, the RKP was initially described by Nuray and Can (2006). These rules are presented in Figure 7.1. In the figure, d_j denotes an individual compound listed in the sets of machine learning technique rankings, $ML_i \{d_j\}$ which consists of n GA or GP rankings. Observing the first fusion rule, SUM computes the mean value of the compound scores or ranks of the rankings. In this case, this is achieved by aggregating all the scores of each database structure, then dividing the score by n . For the MAX, MIN and MED fusion rules, the scores for each database structure d_j are computed by taking the largest, the smallest and the middle score (or median) in the n rankings, respectively. The final rule used for consensus scoring is known as the RKP fusion rule, whereby a compound d_j score is computed by adding the reciprocal of the non-zero scores after the ranking is truncated to a certain percentage p ; for instance, 100% (i.e. the whole database), 50%, 5% and 1%. Notably, the formula of the RKP rule is measured by using the rank position of each molecule to be fused as used by Nuray and Can (2006) in text retrieval.

Fusion Rule	Formula
SUM	$\frac{1}{n} \sum_{i=1}^n ML_i (d_j)$
MAX	$\max\{ML_1 (d_j), ML_2 (d_j), \dots ML_i (d_j), \dots ML (d_j)\}$
MED	$\text{med}\{ML_1 (d_j), ML_2 (d_j), \dots ML_i (d_j), \dots ML (d_j)\}$
MIN	$\min\{ML_1 (d_j), ML_2 (d_j), \dots ML_i (d_j), \dots ML (d_j)\}$
RKP	$\sum_{i=1}^p \frac{1}{ML_i (d_j)}$

Figure 7.1: Fusion rules

Several studies have reported on success of similarity fusion using the SUM fusion rule in applications of similarity searching (Ginn, Willet and Bradshaw, 2000; Whitle, Gillet Willet and Loese, 2006). Other studies have reported that the MAX rule is the best fusion rule for group fusion in similarity searching (Hert, Willett, Wilton, Acklin, Azzaoui, Jacobyn & Schuffenhauer, 2004a; Hert, Willett, Wilton, Acklin, Azzaoui, Jacoby & Schuffenhauer, 2004b; Nasr, Swamidass & Baldi, 2009). Several comparisons on consensus scoring were also reported with applications in docking (Oda, Tsuchida, Takakura, Yamaotsu & Hirono, 2006; Yang, Chen, Shen, Kristal & Hsu, 2005) and in 2D and 3D similarity searching (Zhang & Muegge, 2006). In 2009, however, Cormack, Clarke and Buettcher reported that RKP fusion is the most effective fusion rule for combining multiple document rankings from an information retrieval system (Cormack, Clarke, & Buettcher, 2009). In another study, Chen, Mueller and Willett (2010) found that group fusion can even be superior to similarity fusion. This is case when the RKP fusion rule is applied for combining individual search outputs in similarity-based virtual screening.

Following the fusion rules criteria, it was determined that two variables could be used in the computation of the GA-and-GP-based fusion scores: (1) the score of compounds, which is the sum of GA and GP weights, or (2) the ranking of compounds in the ten sets of GA and GP runs. The first four rules, SUM, MAX, MED and MIN were used to fuse the ten sets of GA and GP runs using both score-based and rank-based data. For the RKP equation, the rule is

applicable only fusing n sets of ranks; hence the RKP rule was applied with rank-based data only. For this study, p value was set as 100%, which otherwise means that the whole database of ranked outputs were fused. In total, nine fusion rules were employed in which a number of the rules are based on ranking information of the data, with the rules listed as *Rank RKP*, *Rank Max*, *Rank Sum*, *Rank Med* and *Rank Min*. The other fusion rules are based on the scoring information of compounds in a dataset, where these fusion rules are referred to as *Score Max*, *Score Sum*, *Score Med* and *Score Min*.

7.4 Results and discussion

7.4.1 GA and GP-based fusion performance analysis

The ranked compounds output of the ten GA-and GP-based SSA for each activity class were combined and used by data fusion using the nine fusion rules mentioned in Section 7.3.2. The enrichment factor of actives retrieved in the top 1% obtained by GA-based fusion based on various fusion rules is shown in Table 7.1. The fusion results are listed and compared against the mean of the ten GA-runs results. Likewise, the GP-based fusion results are listed in Table 7.2 and compared against both the mean of the ten GP-runs results. The highest values are shown as lightly shaded.

Visual inspection of Tables 7.1 indicates that the performance of the GA-based fusion is seen to be more effective than the mean GA results for all activity classes in all three databases. In the case of the GP-based fusion as shown in Table 7.2, data fusion manages to improve the individual rankings in several cases, outperforming the GP-based results in 23 out of 40 activity classes from all MDDR, WOMBAT and ChEMBL databases. In contrast, the remaining 17 activity classes attained similar results when using data fusion. These are (D2, SUBP and COX classes from the MDDR database; ACHE, AT1, PDE, SUBP and THRM from WOMBAT and AT1, COX, D2, FXA, MMP, PDE, PKC, SUBP and THRM classes from ChEMBL database). Based on the number of actives retrieved in the top 1% for all activity classes, the differences between data fusion methods to either GA or GP-based SSA is often very small. On average, there is about one to five to fifteen active compounds retrieval differences recorded.

7.4.2 Kendall's W analysis

The impact of the GA and GP-based fusion were studied further by employing the Kendall's W test of statistical significance to measure the agreement of the fusion rules performance in all three databases. The results are discussed below.

GA-based fusion

For the GA-based fusion, the results obtained from Kendall's W analysis are presented in Table 7.3. The table shows the performance of the nine fusion rules in terms of its rank positions in each activity class for the three databases. The rankings are determined based on the enrichment factor of actives in the top 1% (or 1% cut-off value). The rankings were listed in decreasing order.

Kendall's W analysis of the fusion rules in MDDR classes is listed in Table 7.3(a), in which the total computed value of W is 0.46. The significance of this value was tested using X^2 distribution, giving a value of 45.38 for X^2 at a significance level of $p < 0.01$. The analysis therefore suggests the following ranking:

Rank RKP > Score Max > Rank Max > Score Sum > Rank Sum > Rank Min > Score Med > Rank Med > Score Min > Mean GA

Similar to the MDDR case, the results in Table 7.3(b) indicates a similar ranking trend for the fusion rules in WOMBAT-based classes. The value obtained for W is computed as 0.30 and the significance of the X^2 distribution is valued at 42.26, at a significance level of $p < 0.01$. The following ranks the fusion rules, from the best to worst performing ones:

Rank RKP > Rank Max > Score Max > Score Sum > Rank Sum > Score Min > Rank Min > Score Med > Rank Med > Mean GA

In the case of fusion on ChEMBL classes as shown in Table 7.3(c), it was found that the value W was equivalent to 0.38, and 50.92 for X^2 ($p < 0.01$) followed by the ranking determination as below:

Rank RKP > Rank Max > Score Sum > Score Max > Rank Sum > Rank Min > Rank Med > Score Min > Mean GA > Score Med

In essence, at a significance level of $p < 0.01$, the *Rank RKP* was found to be the best performing rule for the GA-based fusion in MDDR datasets for WOMBAT and ChEMBL sets. The rest of the fusion rules exhibit mixed results across all three databases. The worst performing method can be seen in the *Mean GA* and *Score Med* which were consistently placed in the lower tier of the ranking position.

GP-based fusion

The fusion ranking results of the Kendall's W analysis for all databases are shown in Table 7.4. Following the Kendall's W test, a value of χ^2 as 23.44 and the value of W is 0.21 were derived for the MDDR case (Table 7.4a). This yields a significance value at $p < 0.01$. Based on the mean of ranks information, the following performance were summarised in decreasing order:

Rank RKP > Rank Max > Score Sum > Score Max > Rank Sum > Score Med > Rank Min > Mean GP > Score Min > Rank Med

Similarly, Table 7.4(b) shows the result of the rank position based on the mean recall for the WOMBAT. Here, the value of χ^2 is 16.07, and the value of W is 0.13, resulting in a highly significant value at $p < 0.05$. The resulting ranking for the nine fusion rules is as follows:

Rank RKP > Rank Max > Score Sum > Score Max > Score Med > Rank Sum > Rank Min > Rank Med > Score Min > Mean GP

In the case of fusion on ChEMBL classes (Table 7.4c), the computed value for W is 0.21, yielding a value of 27.92 for χ^2 , hence the following rankings are suggested below at $p < 0.05$ level:

Rank RKP > Score Max > Rank Sum > Rank Max > Score Sum > Score Med > Score Min > Mean GP > Rank Med > Rank Min

Based on the analysis above, similar performance behaviour to the GA-fusion was observed. The *Rank RKP* rule was shown to perform the best for the MDDR, WOMBAT and ChEMBL. *Rank Med*, *Mean GP* and *Rank Min* were the lowest in terms of the number of actives count in the top 1% of the ranking for MDDR, WOMBAT and ChEMBL respectively.

Table 7.5 highlighted fusion rules based on the mean rank for all three databases (i.e., MDDR, WOMBAT and ChEMBL). Here, it was possible to obtain the following observations for GA-based fusion (Table 7.5a), whereby the calculated value $W = 0.94$ which gives a value of $X^2 = 25.40$. For the GP-based fusion shown in Table 7.5(b), Kendall's W results for all three databases yields the value of $W = 0.88$, and the value of $X^2 = 23.79$. Both these results denote a significant value at $p < 0.01$. Subsequently, the best overall ranking of data fusion using GA-based fusion is as follows:

Rank RKP > Rank Max > Score Max > Score Sum > Rank Sum > Rank Min > Rank Med > Score Med > Score Min > Mean GA

For the GP-based fusion, the following ranking is derived:

Rank RKP > Score Max > Rank Max > Score Sum > Rank Sum > Score Med > Rank Min > Score Min > Rank Med > Mean GP

It was concluded that when comparing all the nine fusion rules using the ten runs of GA-based SSA and GP-based SSA methods in all MDDR, WOMBAT and ChEMBL activity classes, the RKP rule was found to perform better than other rules in most cases. This is in agreement with the results reported by Chen, Mueller and Willett (2010), who found that RKP is superior to the other rules in group fusion. In contrast to the best fusion rule determined, it was consistently observed that both *Score* and *Rank MED* and *MIN* rules jointly yielded the worst performances in recall rate for most classes from both databases. This occurred when applying fusion on GA-based SSA and the GP-based compounds ranking results.

7.4.3 Wilcoxon signed rank test

Based on the Kendall's W analysis performed, the GA-based data fusion was deemed more effective than the mean of ten GA-runs in all databases. Thus, it was necessary to use the Wilcoxon signed rank test to quantify the significance of the difference between the performance of data fusion and the mean of the ten GA and GP runs. To conduct the test, the enrichment factor results in the top 1% from both the the mean of ten runs GA and the GP, against the best data fusion rule were observed. A measure of significance following W is measured by referring to the table of critical values for the Wilcoxon test (i.e. $W_{critical}$). In the

Wilcoxon signed rank test, if two scores of any pair are equal (i.e. there is no difference between the two compared entities), then such pairs are discarded from the analysis. This can be observed in the case of the WOMBAT database, where the sample is reduced to 12 (i.e. $N-2 = 12$) since the THRM and HIVP classes had tied recall rates. These were consequently ignored (Ott & Longnecker, 2015). Further information about the statistical tests can also be found in Chapter 3, Section 3.5.3.

For the case of the MDDR database and comparing the mean of ten GA runs against the RKP method, the Wilcoxon signed rank test showed a value of $W = 0$ and the critical value of W for $N = 11$ at $p < 0.01$ is 5. In the case of WOMBAT, the value of $W = 3$ and the critical value of W for $N = 14$ at $p < 0.01$ is 12. Finally, for the ChEMBL sets, the value of $W = 0$ with the value of $W_{critical}$ for $N = 15$ at $p < 0.01$ is 15. Overall, the data fusion results appear to be significant when compared against mean results of the GA runs. Hence, it can be concluded that *Rank RKP* fusion rules provide good enhancement of recall rates when compared with the mean of individual GA.

Similar to the comparison of mean of ten GP runs against the best fusion rule *Rank RKP* for both MDDR and WOMBAT sets, and the ChEMBL, W values were determined to be equal to 0, 16 and 3, calculated for N equivalent to 10, 10 and 10 based on the MDDR, WOMBAT and ChEMBL sets respectively. The observed difference between data fusion method and the mean result of the ten GP runs in this analysis was significant in the MDDR and ChEMBL dataset. It was, however, not significant for the case of WOMBAT dataset. The argument here is that the GP results are highly consistent between the multiple runs for each class in all three databases. Referring to Table 6.10, there is a very high degree of consistency in the retrieval of identical actives compounds from the ten different GP runs. The number of actives retrieved in the top 1% for the ten runs were also similar. This indicates that actives compounds are clustered closely in the top rank of the ranked data. This is in agreement with the results reported by Wu and Huang. It was found that the diversity of the relevant documents in the ranked list of documents can affect the performance of data fusion. Better performance of data fusion is more likely with a higher rate of diversity in the fused input (Wu & Huang, 2014).

7.5 Conclusion

This chapter described the investigation of data fusion which sought to combine retrieval results from multiple, individual GA-based SSA and GP-based SSA results for each activity class. From the experiment and various analyses performed. It can be concluded that, the data fusion was found to perform better in each activity class from the three databases utilising the *Rank RKP*. By contrast, the *Rank Min*, *Score Min*, *Rank Med* and *Score Med* and fusion rules were found to be the worst fusion rule performers in the MDDR, WOMBAT and ChEMBL databases. It was also found that for the comparison of data fusion to the mean of ten runs of the GA and GP, the difference was found to be significant for both the GA-based SSA and GP-based fusion. However, for the GP case of WOMBAT dataset, the difference was instead, not significant. The GA and GP is essentially a robust and non-deterministic process. Hence, data fusion can be used as a deterministic measure to produce a single, unified outcome. It was found that the fusion of multiple rankings of the GA-based SSA and GP-based SSA produced a significant improvement in the final ranking results, with easy implementation. These conclusions confirm that the data fusion approach SSA is found to be highly effective technique in enhancing the retrieval performance of SSA specifically for the GA-based SSA and GP-based SSA.

Table 7.1: Enrichment factor of actives when using combination of different GA runs for top 1% in (a) MDDR dataset for eleven activity classes (b) WOMBAT dataset for fourteen activity classes and (c) ChEMBL dataset for fifteen activity classes

Activity Class	Actives	Actives Retrieved									
		Mean GA	Rank RKP	Rank Max	Rank Med	Rank Sum	Rank Min	Score Med	Score Sum	Score Max	Score Min
(a)											
5HT3	677	41.18	43.57	43.43	41.80	42.84	42.54	42.69	43.28	42.69	41.65
5HT1A	744	19.22	20.16	19.76	19.09	19.89	19.35	19.62	19.89	19.62	19.62
5HT	323	17.89	18.89	18.58	18.27	19.20	19.20	19.50	18.89	18.89	17.96
D2	356	17.39	18.26	17.98	14.33	17.13	19.38	19.66	17.42	19.38	14.04
RNN	1017	70.42	72.17	72.47	71.78	71.98	71.29	71.88	71.78	72.27	69.32
AT1	849	47.96	49.00	49.00	49.47	48.88	48.76	48.53	48.88	49.35	47.94
THRM	723	48.33	49.52	49.79	48.96	49.93	49.10	47.58	49.93	48.96	48.82
SUBP	1121	29.16	30.87	30.42	28.90	30.24	29.97	28.90	30.51	30.51	28.90
HIVP	675	47.04	49.04	49.04	47.70	48.74	47.70	45.33	48.89	49.48	47.85
COX	572	29.28	30.24	30.59	30.77	30.77	29.55	29.02	30.24	30.77	30.94
PKC	408	29.02	31.13	30.88	28.43	29.41	30.39	29.66	30.64	31.37	26.72
(b)											
5HT1A	533	55.65	58.72	57.60	58.54	57.41	57.04	53.10	57.41	56.47	55.72
5HT3	198	43.48	44.44	44.44	43.43	44.44	43.94	43.43	44.44	46.46	43.43
ACHE	453	51.13	51.43	52.10	50.33	51.88	50.99	53.20	51.21	49.89	54.30
AT1	652	80.69	82.98	82.98	83.74	83.44	83.13	82.52	82.98	82.36	83.59
COX	869	67.42	67.55	67.55	67.43	67.43	67.43	66.86	67.55	67.55	67.78
D2	819	40.75	41.88	41.03	42.74	41.76	40.90	42.00	41.39	41.39	41.27
FXA	758	45.67	47.76	44.99	41.16	43.01	47.89	43.27	44.59	40.37	45.65
HIVP	1015	53.56	57.34	57.14	51.33	54.68	55.47	55.47	55.67	56.35	54.19
MMP	625	63.41	64.64	64.96	64.64	64.64	63.68	63.84	64.64	64.96	62.56
PDE	536	49.27	51.87	52.05	49.81	51.68	51.68	49.81	51.87	52.80	48.32
PKC	128	73.99	72.66	73.44	71.88	71.88	71.88	72.66	71.88	75.00	71.09
RNN	427	78.76	80.56	80.56	80.56	80.80	80.33	77.52	81.03	81.03	81.73
SUBP	502	49.22	51.59	52.59	49.80	52.79	50.80	50.80	52.19	51.00	51.39
THRM	379	56.12	58.84	58.58	55.15	57.78	58.58	58.84	57.78	58.05	55.15

Activity Class	Actives	Actives Retrieved									
		Mean GA	Rank RKP	Rank Max	Rank Med	Rank Sum	Rank Min	Score Med	Score Sum	Score Max	Score Min
					(c)						
5HT1A	1335	39.72	40.22	40.45	40.45	40.67	40.15	39.48	40.52	40.60	40.52
5HT3	192	55.05	59.38	56.25	57.81	58.33	53.65	53.13	57.29	56.25	55.73
5HT	2202	34.52	35.10	35.42	33.42	34.51	35.06	32.97	34.74	35.79	33.92
ACHE	665	36.77	37.14	40.90	33.83	35.79	41.50	41.20	38.20	35.79	31.13
AT1	95	81.37	84.21	84.21	85.26	85.26	81.05	82.11	84.21	85.26	84.21
COX	125	38.32	40.00	39.20	40.00	39.20	37.60	39.20	37.60	39.20	35.20
D2	1672	31.82	32.48	33.37	31.46	32.95	34.03	31.70	33.13	30.68	30.56
FXA	1352	47.40	50.00	49.63	46.97	48.45	47.86	47.56	49.11	50.00	47.34
HIVP	1941	64.85	68.38	67.51	65.86	67.25	66.84	63.80	67.66	66.63	65.04
MMP	356	68.79	69.66	69.38	67.98	69.38	69.10	67.98	69.10	69.94	67.13
PDE	229	40.22	41.48	44.54	39.30	41.48	42.36	39.30	42.79	40.61	45.85
PKC	190	58.63	60.00	58.42	58.42	58.42	59.47	57.37	58.42	58.95	58.95
RNN	884	56.64	58.71	59.39	56.45	58.26	58.37	56.11	58.94	56.90	56.33
SUBP	762	70.08	72.97	73.23	70.47	71.78	72.44	71.39	71.65	72.83	70.47
THRM	754	46.76	48.28	48.67	48.28	48.41	48.01	47.48	48.81	48.28	45.36

Table 7.2: Enrichment factor of actives when using a combination of different GP runs for the top 1 % in (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset for fifteen activity classes

Activity Class	Actives	Actives Retrieved									
		Mean GP	Rank RKP	Rank Max	Rank Med	Rank Sum	Rank Min	Score Med	Score Sum	Score Max	Score Min
(a)											
5HT3	677	26.65	26.88	25.85	26.14	26.44	25.85	26.29	27.03	26.74	26.44
5HT1A	744	15.75	16.40	15.99	13.84	15.05	15.73	15.46	15.99	16.53	15.19
5HT	323	19.81	21.05	20.74	20.12	21.05	19.81	18.89	20.43	20.43	20.43
D2	356	22.47	22.75	21.91	22.75	22.19	21.91	23.03	22.47	23.03	21.63
RNN	1017	63.60	63.62	63.82	62.73	63.32	63.72	64.01	63.13	62.54	63.82
AT1	849	48.50	48.65	48.88	48.41	48.53	47.23	48.41	48.29	48.53	48.53
THRM	723	44.48	45.23	46.06	44.67	45.64	46.06	43.98	45.64	44.95	44.12
SUBP	1121	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55	23.55
HIVP	675	40.04	42.52	41.48	40.59	41.93	40.74	40.00	41.78	39.85	40.30
COX	572	31.08	31.47	31.47	31.47	30.94	31.29	31.64	31.29	31.64	30.24
PKC	408	25.10	25.74	25.74	24.51	25.25	26.23	25.49	25.98	25.49	24.51
(b)											
5HT1A	533	48.95	49.53	48.97	49.91	49.53	48.78	48.97	49.91	49.34	48.97
5HT3	198	41.97	41.92	42.42	43.94	41.92	42.42	37.88	41.41	40.91	40.91
ACHE	453	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45	49.45
AT1	652	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99	78.99
COX	869	64.03	64.33	64.21	63.75	64.10	64.10	64.10	64.21	64.10	63.75
D2	819	35.90	36.51	36.39	34.19	35.16	35.78	36.02	36.39	36.63	35.78
FXA	758	40.24	39.71	41.03	39.05	38.92	40.24	41.03	39.05	41.03	39.05
HIVP	1015	42.28	43.55	42.66	42.56	42.17	42.07	42.76	42.56	42.46	42.07
MMP	625	62.88	61.44	63.20	64.16	64.32	63.20	63.20	63.20	63.84	63.20
PDE	536	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96	44.96
PKC	128	76.88	78.13	77.34	75.00	75.78	76.56	77.34	78.13	77.34	77.34
RNN	427	74.82	76.81	75.18	74.71	76.11	75.88	74.71	76.35	76.11	75.88
SUBP	502	46.73	46.81	47.01	44.82	46.81	47.01	47.01	46.81	46.81	47.01
THRM	379	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15	55.15

Activity Class	Actives	Actives Retrieved									
		Mean GP	Rank RKP	Rank Max	Rank Med	Rank Sum	Rank Min	Score Med	Score Sum	Score Max	Score Min
					(e)						
5HT1A	1335	32.67	33.71	32.81	32.73	33.11	32.51	32.43	32.96	33.26	31.76
5HT3	192	51.35	52.08	51.04	51.56	52.08	51.04	51.56	52.08	51.56	51.56
5HT	2202	29.72	30.43	29.70	29.79	30.20	29.43	29.79	30.11	29.79	29.56
ACHE	665	29.50	30.08	30.08	30.08	30.38	27.22	28.87	29.92	29.92	29.92
AT1	95	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00
COX	125	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20	31.20
D2	1672	28.06	28.11	28.23	27.63	28.11	28.11	27.81	28.11	28.23	28.23
FXA	1352	39.25	39.20	39.28	39.13	39.20	39.20	39.20	39.20	39.28	39.13
HIVP	1942	49.63	53.99	50.54	49.46	50.80	49.41	49.92	49.92	52.55	49.05
MMP	356	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03	73.03
PDE	229	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69	29.69
PKC	190	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05	61.05
RNN	884	51.06	51.58	51.47	50.79	51.36	52.04	51.13	51.58	51.13	51.13
SUBP	762	69.99	69.95	69.95	69.95	69.95	69.42	70.08	69.95	70.08	70.08
THRM	754	36.82	37.93	37.00	37.00	37.00	36.07	37.53	36.74	37.93	37.53

Table 7.3: Kendall’s W analysis for the number of actives retrieved in top 1% of the GA searches and after application of data fusion on (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset of fifteen

(a)

Fusion Rules	Activity Class											Mean Rank	Rank Position
	5HT3	5HT1A	5HT	D2	RNN	AT1	THRM	SUBP	HIVP	COX	PKC		
Rank RKP	9.00	9.00	5.00	6.00	7.00	6.50	6.00	9.00	7.50	3.50	8.00	6.95	1
Score Max	4.50	4.00	5.00	7.50	8.00	8.00	3.50	7.50	9.00	7.00	9.00	6.64	2
Rank Max	8.00	6.00	3.00	5.00	9.00	6.50	7.00	6.00	7.50	5.00	7.00	6.36	3
Score Sum	7.00	7.50	5.00	4.00	3.50	4.50	8.50	7.50	6.00	3.50	6.00	5.73	4
Rank Sum	6.00	7.50	7.50	2.00	6.00	4.50	8.50	5.00	5.00	7.00	3.00	5.64	5
Rank Min	3.00	2.00	7.50	7.50	2.00	3.00	5.00	4.00	2.50	2.00	5.00	3.95	6
Score Med	4.50	4.00	9.00	9.00	5.00	2.00	0.00	1.00	0.00	0.00	4.00	3.50	7
Rank Med	2.00	0.00	2.00	1.00	3.50	9.00	3.50	1.00	2.50	7.00	1.00	2.95	8
Score Min	1.00	4.00	1.00	0.00	0.00	0.00	2.00	1.00	4.00	9.00	0.00	2.00	9
Mean GA	0.00	1.00	0.00	3.00	1.00	1.00	1.00	3.00	1.00	1.00	2.00	1.27	10

(b)

Fusion Rules	Activity Class														Mean Rank	Rank Position
	5HT1A	5HT3	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
Rank RKP	9.00	6.50	5.00	4.00	6.50	7.00	8.00	9.00	5.50	6.50	5.50	4.00	6.00	8.50	6.50	1
Rank Max	7.00	6.50	7.00	4.00	6.50	2.00	5.00	8.00	8.50	8.00	7.00	4.00	8.00	6.50	6.29	2
Score Max	3.00	9.00	0.00	1.00	6.50	4.50	0.00	7.00	8.50	9.00	9.00	7.50	4.00	5.00	5.29	3
Score Sum	5.50	6.50	4.00	4.00	6.50	4.50	4.00	6.00	5.50	6.50	2.50	7.50	7.00	3.50	5.25	4
Rank Sum	5.50	6.50	6.00	7.00	3.00	6.00	2.00	3.00	5.50	4.50	2.50	6.00	9.00	3.50	5.00	5
Score Min	2.00	1.00	9.00	8.00	9.00	3.00	6.00	2.00	0.00	0.00	0.00	9.00	5.00	0.50	3.89	6
Rank Min	4.00	4.00	2.00	6.00	3.00	1.00	9.00	4.50	2.00	4.50	2.50	2.00	2.50	6.50	3.82	7
Score Med	0.00	1.00	8.00	2.00	0.00	8.00	3.00	4.50	3.00	2.50	5.50	0.00	2.50	8.50	3.46	8
Rank Med	8.00	1.00	1.00	9.00	3.00	9.00	1.00	0.00	5.50	2.50	2.50	4.00	1.00	0.50	3.43	9
Mean GA	1.00	3.00	3.00	0.00	1.00	0.00	7.00	1.00	1.00	1.00	8.00	1.00	0.00	2.00	2.07	10

(c)

Fusion Rules	Activity Class															Mean Rank	Rank Position
	5HT1A	5HT3	5HT	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
Rank RKP	3.00	9.00	7.00	5.00	4.50	8.50	5.00	8.50	9.00	8.00	4.50	9.00	7.00	8.00	5.00	6.73	1
Rank Max	4.50	4.50	8.00	7.00	4.50	5.50	8.00	7.00	7.00	6.50	8.00	2.50	9.00	9.00	8.00	6.60	2
Score Sum	6.50	6.00	5.00	6.00	4.50	1.50	7.00	6.00	8.00	4.50	7.00	2.50	8.00	4.00	9.00	5.70	3
Score Max	8.00	4.50	9.00	2.50	8.00	5.50	1.00	8.50	4.00	9.00	3.00	6.50	4.00	7.00	5.00	5.70	4
Rank Sum	9.00	8.00	3.00	2.50	8.00	5.50	6.00	5.00	6.00	6.50	4.50	2.50	5.00	5.00	7.00	5.57	5
Rank Min	2.00	1.00	6.00	9.00	0.00	1.50	9.00	4.00	5.00	4.50	6.00	8.00	6.00	6.00	3.00	4.73	6
Rank Med	4.50	7.00	1.00	1.00	8.00	8.50	2.00	0.00	3.00	1.50	0.50	2.50	2.00	1.50	5.00	3.20	7
Score Min	6.50	3.00	2.00	0.00	4.50	0.00	0.00	1.00	2.00	0.00	9.00	6.50	1.00	1.50	0.00	2.47	8
Mean GA	1.00	2.00	4.00	4.00	1.00	3.00	4.00	2.00	1.00	3.00	2.00	5.00	3.00	0.00	1.00	2.40	9
Score Med	0.00	0.00	0.00	8.00	2.00	5.50	3.00	3.00	0.00	1.50	0.50	0.00	0.00	3.00	2.00	1.90	10

Table 7.4: Kendall's W analysis for the number of actives retrieved in top 1% of the GP searches and after application of data fusion on (a) MDDR dataset of eleven activity classes (b) WOMBAT dataset of fourteen activity classes and (c) ChEMBL dataset of fifteen

(a)

Fusion Rules	Activity Class											Mean Rank	Rank Position
	5HT3	5HT1A	5HT	D2	RNN	AT1	THRM	SUBP	HIVP	COX	PKC		
Rank RKP	8.00	8.00	8.50	6.50	5.00	8.00	5.00	4.50	9.00	6.00	6.50	6.82	1
Rank Max	0.50	6.50	7.00	1.50	7.50	9.00	8.50	4.50	6.00	6.00	6.50	5.77	2
Score Sum	9.00	6.50	5.00	4.50	2.00	1.00	6.50	4.50	7.00	3.50	8.00	5.23	3
Score Max	7.00	9.00	5.00	8.50	0.00	6.00	4.00	4.50	0.00	8.50	4.50	5.18	4
Rank Sum	4.50	1.00	8.50	3.00	3.00	6.00	6.50	4.50	8.00	1.00	3.00	4.45	5
Score Med	3.00	3.00	0.00	8.50	9.00	2.50	0.00	4.50	1.00	8.50	4.50	4.05	6
Rank Min	0.50	4.00	1.50	1.50	6.00	0.00	8.50	4.50	5.00	3.50	9.00	4.00	7
Mean GP	6.00	5.00	1.50	4.50	4.00	4.00	2.00	4.50	2.00	2.00	2.00	3.41	8
Score Min	4.50	2.00	5.00	0.00	7.50	6.00	1.00	4.50	3.00	0.00	0.50	3.09	9
Rank Med	2.00	0.00	3.00	6.50	1.00	2.50	3.00	4.50	4.00	6.00	0.50	3.00	10

(b)

Fusion Rules	Activity Class														Mean Rank	Rank Position
	5HT1A	5HT3	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
Rank RKP	6.50	4.50	4.50	4.50	9.00	8.00	4.00	9.00	0.00	4.50	8.50	9.00	3.50	4.50	5.71	1
Rank Max	3.00	7.50	4.50	4.50	7.50	6.50	8.00	7.00	4.00	4.50	5.50	3.00	7.50	4.50	5.54	2
Score Sum	8.50	3.00	4.50	4.50	7.50	6.50	2.00	5.50	4.00	4.50	8.50	8.00	3.50	4.50	5.36	3
Score Max	5.00	1.50	4.50	4.50	4.50	9.00	8.00	4.00	7.00	4.50	5.50	6.50	3.50	4.50	5.18	4
Score Med	3.00	0.00	4.50	4.50	4.50	5.00	8.00	8.00	4.00	4.50	5.50	0.50	7.50	4.50	4.57	5
Rank Sum	6.50	4.50	4.50	4.50	4.50	1.00	0.00	2.00	9.00	4.50	1.00	6.50	3.50	4.50	4.04	6
Rank Min	0.00	7.50	4.50	4.50	4.50	2.50	5.50	0.50	4.00	4.50	2.00	4.50	7.50	4.50	4.04	7
Rank Med	8.50	9.00	4.50	4.50	0.50	0.00	2.00	5.50	8.00	4.50	0.00	0.50	0.00	4.50	3.71	8
Score Min	3.00	1.50	4.50	4.50	0.50	2.50	2.00	0.50	4.00	4.50	5.50	4.50	7.50	4.50	3.54	9
Mean GP	1.00	6.00	4.50	4.50	2.00	4.00	5.50	3.00	1.00	4.50	3.00	2.00	1.00	4.50	3.32	10

(c)

Fusion Rules	Activity Class															Mean Rank	Rank Position
	5HT1A	5HT3	5HT	ACHE	AT1	COX	D2	FXA	HIVP	MMP	PDE	PKC	RNN	SUBP	THRM		
Rank RKP	9.00	8.00	9.00	7.00	4.50	4.50	4.50	4.00	9.00	4.50	4.50	4.50	7.50	3.00	8.50	6.13	1
Score Max	8.00	4.50	5.00	4.00	4.50	4.50	8.00	8.50	8.00	4.50	4.50	4.50	3.00	8.00	8.50	5.87	2
Rank Sum	7.00	8.00	8.00	9.00	4.50	4.50	4.50	4.00	7.00	4.50	4.50	4.50	5.00	3.00	4.00	5.47	3
Rank Max	5.00	0.50	2.00	7.00	4.50	4.50	8.00	8.50	6.00	4.50	4.50	4.50	6.00	3.00	4.00	4.83	4
Score Sum	6.00	8.00	7.00	4.00	4.50	4.50	4.50	4.00	4.50	4.50	4.50	4.50	7.50	3.00	1.00	4.80	5
Score Med	1.00	4.50	5.00	1.00	4.50	4.50	1.00	4.00	4.50	4.50	4.50	4.50	3.00	8.00	6.50	4.07	6
Score Min	0.00	4.50	1.00	4.00	4.50	4.50	8.00	0.50	0.00	4.50	4.50	4.50	3.00	8.00	6.50	3.87	7
Mean GP	3.00	2.00	3.00	2.00	4.50	4.50	2.00	7.00	3.00	4.50	4.50	4.50	1.00	6.00	2.00	3.57	8
Rank Med	4.00	4.50	5.00	7.00	4.50	4.50	0.00	0.50	2.00	4.50	4.50	4.50	0.00	3.00	4.00	3.50	9
Rank Min	2.00	0.50	0.00	0.00	4.50	4.50	4.50	4.00	1.00	4.50	4.50	4.50	9.00	0.00	0.00	2.90	10

Table 7.5: Kendall's W analysis for the top 1% based on the average of enrichment factor actives in the top 1% of (a) The GA-based SSA and (b) GP-based SSA from the MDDR, WOMBAT and ChEMBL databases

(a)

Fusion Rules	Databases			Mean Rank	Rank Position
	MDDR	WOMBAT	ChEMBL		
Rank RKP	6.95	6.50	6.73	6.73	1
Rank Max	6.36	6.29	6.60	6.42	2
Score Max	6.64	5.29	5.70	5.87	3
Score Sum	5.73	5.25	5.70	5.56	4
Rank Sum	5.64	5.00	5.57	5.40	5
Rank Min	3.95	3.82	4.73	4.17	6
Rank Med	2.95	3.43	3.20	3.19	7
Score Med	3.50	3.46	1.90	2.95	8
Score Min	2.00	3.89	2.47	2.79	9
Mean GA	1.27	2.07	2.40	1.91	10

(b)

Fusion Rules	Databases			Mean Rank	Rank Position
	MDDR	WOMBAT	ChEMBL		
Rank RKP	6.82	5.71	6.13	6.22	1
Score Max	5.18	5.18	5.87	5.41	2
Rank Max	5.77	5.54	4.83	5.38	3
Score Sum	5.23	5.36	4.80	5.13	4
Rank Sum	4.45	4.04	5.47	4.65	5
Score Med	4.05	4.57	4.07	4.23	6
Rank Min	4.00	4.04	2.90	3.65	7
Score Min	3.09	3.54	3.87	3.50	8
Rank Med	3.41	3.32	3.50	3.41	9
Mean GP	3.00	3.71	3.57	3.43	10

Chapter 8

Conclusion and Future Work

8.1 Introduction

This chapter summarises the use of evolutionary algorithms for the purpose of chemical substructure search and analysis. It is discussed in general both the performance and impact of the experimented GA and GP methods for SSA, and their subsequent data fusion application. Also considered are several key points related to potential future work that follow this study's findings.

Virtual screening methods are increasingly used to improve the cost-effectiveness of drug discovery programmes. In the drug discovery pipeline, virtual screening is performed during the lead identification process. This mainly involves the screening and analysis of large-scale data, usually amounting to millions of compounds. Hence, a small enhancement in the virtual screening methods has the potential to improve the effectiveness of the drug discovery life-cycle. There have been many published works on various ligand-based virtual screening methods, such as similarity searching, QSAR and pharmacophore mapping (Frearson & Collie, 2009; Lengauer et al., 2004; McInnes, 2007; Willett, 2009). Such progress, however, is limited with regards to the SSA ever since its introduction by Cramer et al. (1974, 1976), apart from published works by Hodes et al. (1977), Klopman (1984, 1992), Ormerod et al. (1998), Gillet et al. (1998), Cosgrove et al. (1998), Wilton et al. (2003) and Hert et al. (2006). This largely reflects industry's preferences for other methods: e.g. similarity methods when compared to SSA. The challenge lies in improving the SSA technique to allow for better identification of suitable and diverse potentially active compounds. It is also important to further demonstrate a higher degree of sophistication in the data analysis of these compounds. SSA has the potential to yield improved and efficient results when optimising for lead searching, as currently, it is relatively simple (albeit logical) in implementation. The present study resolved to quantify the possibility of meaningful increase in active compounds rankings by harnessing the non-deterministic nature of GA and GP. Such methods are known to allow for a greater degree of freedom in determining independent parameters and relationships, which are otherwise constrained in traditional weighting calculations.

8.2 Contributions

8.2.1 The comparison of existing SSA weighting schemes

Various weighting schemes are readily available in SSA, some of which are closely related to each other. While several studies have been conducted to analyse the effectiveness of such weighting schemes (Ormerod, 1992), these evaluations were restricted to legacy molecular databases with limited activity classes. Based on the results described in Chapter 4, an updated analysis of the effectiveness of established SSA weighting schemes was presented to measure the predictive performance of a given biological activity class using three large databases, MDDR, WOMBAT and ChEMBL. In this analysis, a predictive experiment was conducted and the molecules were represented using the MDL fragment description, consisting of a 166 key-set. SSA was applied in three general steps: First, weights of the individual fragments were determined based on the equation governed by the chosen SSA weighting scheme. Second, either the total or mean value of the scores of each compound in the training set were calculated, again depending on the SSA scheme used. Finally, ranking of the compounds in the database was achieved in order of descending score value. Ten SSA weighting schemes were evaluated in comparison to previous works performed (Ormerod, Willett and Bawden, 1989; Cosgrove and Willett, 1998; Wilton, Willett, Lawson and Mullier, 2003). In the earlier findings, it was found that the R2 scheme proved to be more effective in SSA than other existing weighting schemes. Such test, however, was conducted on a relatively small dataset.

Through rigorous experimentation as described in this thesis, it was possible to establish several trends. The first and the best trend was discovered to be the R4 and R3 weighting schemes, which performed well for a majority of activity classes from the three databases. A second trend follows the R1, R2 and NBC weighting schemes; all placed in the mid-tier of the weighting scheme performance. The third trend is the less effective schemes which include the original SAF, AVID, WT2 and the WT1 schemes. These schemes usually have fairly average retrieval performance for most activity classes. The worst performing weighting scheme is found to be the SAS weighting scheme, annotated as the fourth trend. It should be noted that the R3 and R4 schemes, alongside the R1 and R2 were in fact originally designed for document-based classification and retrieval. From the experiments, it was found that the Robertson-Sparck Jones R4 weighting scheme generally works best for the majority of the activity classes tested. This scheme was subsequently selected as a benchmark for performance analysis to other methods proposed in this thesis. The results shown above

signify that the choice of weighting scheme is deemed significant to the distribution of actives / inactives especially in the top 1% of ranked compounds.

8.2.2 The use of GA to the SSA method

Chapter 5 discussed the application of the GA in SSA for fragment weighting determination. In this section of the work, the possibility of uplift in the approach of a GA on fragment weighting determination when compared to the SSA R4 weights was evaluated. For the GA, the chromosome is a vector containing N integers, where the i -th element is the fragment weight for the i -th bit in the fingerprint. The fitness function for the GA is the number of active molecules that occur in the top 1% of a ranked training set based on the N weights representing the chromosomes.

From the parameterisation tests, it was discovered that the GA-based SSA can be successfully implemented when taking into account the influence of SSA R4 weights (i.e. its weights polarity). This was applied during the initialisation of the chromosome population and for subsequent genetic operations. For instance, if the R4 scheme determines that a particular fragment contains negative weights, it was necessary to put in place a similar restriction during the construction and manipulation of the chromosome weights. This was conducted in order to retain its weighting characteristics following R4. The GA is run for a pre-set number of generations or until the best GA weights have stabilised. It thus provides an estimate of the best possible GA-based SSA weights that can be obtained using a training set. The resulting weights are then applied to a separate test set, and its performance evaluated.

The experimental results from Chapter 5 confirmed that the developed GA-based SSA weighting scheme performed consistently well by yielding improved trends of higher active retrieval rates in the first 1% ranking for all of the activity classes in all of the databases, MDDR, WOMBAT and ChEMBL, compared to the SSA R4. The findings of this study suggest that the GA provides a possible non-deterministic method for generating fragment weights to be used in SSA-based virtual screening method. More importantly, this study strengthens the view that the GA-based SSA method was able to produce improved active retrieval performance when compared to the existing weighting schemes in the SSA. Despite the GA's evolutionary and non-deterministic nature, this study proved the effectiveness of such a method in improving virtual screening methods for lead identifications in drug discovery. Taken together, these findings indicate that the GA-based SSA method is

applicable and yields better performance of the active retrieval rate than the existing SSA techniques.

8.2.3 Investigation of the use of GP in the SSA method

Chapter 6 discussed in detail the investigation into the use of GP for the SSA. The GP method is fundamentally similar to the GA in terms of the manipulation of chromosomes to evolve into solutions via genetic operations. The difference in the GP method is found in its use of variables and terminals. These are formed into equations to represent the chromosomes as the main entities on which to perform evolution, instead of the binary or decimal vectors used in GA. In one sense, this approach is similar to any existing weighting schemes under SSA, such as the SAF weighting scheme, or the Robertson-Sparck Jones schemes (i.e. R1, R2, R3 and R4). Here, an equation is translated to fragment weights and subsequently applied for compound ranking purposes. The approach to the GP was to utilise all of the variables, mathematical operators and sub-terms found in those SSA weighting equations. The aim was to find through evolutionary means a new equation which may improve active retrieval performance compared to the methods explored earlier. Based on the analysis of the GP-based SSA application, it is possible to identify several important facts, which are discussed below.

First, it was found that the GP-based SSA was able to outperform the SSA R4 weighting schemes in all of the activity classes across the three databases. The improvement, however, varied significantly from borderline to good. Regarding its comparison with the GA-based SSA, the GP-based SSA recorded a higher increase in active retrieval performance for several activity classes across the three databases, such as the MDDR COX and RNN classes, the WOMBAT PKC, and the ChEMBL 5HT3 class. In the rest of the classes, however, the GP runs recorded a varying degree of performance. In the majority of cases, they were not able to achieve a similar level of active retrieval rates compared to the GA-based SSA.

This finding provided insights into the advantages of the GA and its relatively superior performance compared to both the GP method and the traditional SSA schemes. It was argued that the GA-based SSA's superiority lies in its ability to evolve weights directly for each fragment in order to elevate active compounds to the top 1% of the ranked data. This is in contrast to the GP-based SSA and the traditional SSA weighting schemes, which require fragment weight determination to be dictated and translated based on the equation that

follows. Nevertheless, the GP-based SSA still manages to deliver positive improvements over the existing SSA methods.

8.2.4 Investigation into the use of data fusion to the GA-and GP-based SSA

The study further explored both GA and GP's potential for further upper-bound improvements in the retrieval of actives, considering that multiple runs of the two methods above for each activity class is available. Based on the findings in Chapter 7, the fusion experiments discussed in this thesis fused rankings and compound scores using nine fusion rules. These were *Rank RKP*, *Rank Max*, *Rank Sum*, *Rank Med*, *Rank Min*, *Score Max*, *Score Sum*, *Score Med* and *Score Min*. The results obtained from the fused GA-based SSA and GP-based SSA searches were compared to the mean of the multiple GA and GP runs. The findings from this study offer several contributions to the current literature. First, the results of this investigation show that the fusion of multiple rankings of the GA-based SSA and GP-based SSA produced significant results in terms of an improvement in the active retrieval rate.

The second major finding was that the most effective fusion rule to be used in the GA and GP SSA-based weighting schemes were determined as the *Rank RKP* rule. On the other hand, the fusion rules *MED* and *MIN* showed the worst performances relative to the three MDDR, WOMBAT and ChEMBL databases. The findings of this investigation also complement those of earlier studies which found that the resultant combined ranking will generally be, at least as good as, and often superior to, the individual ranking (Ginn et al., 2006, Hert et al., 2006). The findings of this experiment could be used to help the standard practice of data fusion in virtual screening, and to guide further enhancement in SSA.

8.3 Suggestion for future work

There were several alternative experimentation methods and testing choices available which were not pursued during the course of this research. These are listed below as suggestions to further the findings of the research in this thesis:

- i. In this study, the MDL fingerprint was used to evaluate the performance of the GA-and GP-based SSA weighting schemes developed. It would be interesting to assess the performance of the GA-based SSA and GP-based SSA when other popular binary descriptors in chemoinformatics are used, such as ECFP₄, BCI and Sunset fingerprints.
- ii. A simple approach to the GA-and GP-based SSA methods was developed. While it

can be seen that the GA-based SSA was able to outperform and significantly increase the upper-bound performance of active retrievals in some cases, the GP-based SSA remains largely less effective than its GA counterpart. This was despite the fact that both methods were shown to improve on the traditional SSA weighting schemes. For the GP-based SSA method, the following are suggested for improvements. (i) One recommended approach is to test a different chromosome population, perhaps by introducing a number of different variables defining the characteristics of the fragments and molecules that make up the molecule set, other than those already used by the SSA equations; (ii) It is beneficial to consider cross-validation methods based on the training set and test set combination. This is to serve two goals: The first goal follows the current one, which aims to increase the chance of finding a good solution by maximising actives retrieval. The second goal is as a support method to the fitness function in validating chromosome suitability (reducing the chance of overfitting). Indeed, further tests of the cross-validation method could also be applied to the GA-based SSA too.

- iii. To explore more sophisticated fitness function such as the implementation of multi-objective GA and GP approach. Other information can be utilised for fitness determination such as properties defined by Lipinski's rule of 5. Example properties are molecular weight, hydrogen bond donors and bond acceptors, log P and molecular mass. Also, by checking the solubility of the compounds. This is to lower the cost of high clinical failure rates during the drug development process.
- iv. To explore parallelisation and code optimisation of the GA-and GP-based SSA. Benchmark results have shown that the run-time of both methods is normally acceptable. It is stressed, however, that the application run-time can be enhanced further by exploring the parallel computing option. Developing the GA and GP program using a low-level language that dictates memory usage and monitoring may also help with runtime efficiencies. This would be especially useful when performing the GA-and GP-based SSA on a much larger database than the ChEMBL (itself comprising of over a million compounds). This is also beneficial in circumstances where a retrospective study needs to be conducted on one whole database.

8.4 Conclusion

The findings from this study offer several contributions to the current literature, which are discussed below:

- i. Application of evolutionary algorithms via GA and GP were investigated for SSA methods. Here, we found good to significant improvements of such methods on SSA performance in ligand-based virtual screening. Specifically, it was shown that the GA-based SSA is consistently superior to both the GP-based SSA and traditional SSA methods. We conclude that the GA-based SSA is clearly a step-up as an alternative to the NBC and other existing SSA weighting scheme introduced nearly 40 years ago. We have also demonstrated the real world practicality aspect of a GA-based SSA approach based on successful investigation of such method on large datasets (ChEMBL) on standard machine resources, with room for more deployment optimisation.
- ii. Secondly, our study also revealed that the GA-based SSA results can be further enhanced using data fusion method to provide a deterministic measure of providing a single unified outcome. The study showed that fusion of GA-based SSA produced somewhat effective and significant final ranking results, with relatively easy implementation. This result overall complements previous findings on data fusion's advantages in information retrieval and chemoinformatics. We hence recommend this as the most effective method overall for GA-based SSA virtual screening.
- iii. The outcomes from this study may hopefully help pharmaceutical researchers to increase the chance of lead identification through alternative method of SSA via evolutionary-based approaches. This study has shown a positive indication of potential future researches into machine learning methods in ligand-based virtual screening. In conclusion, this research has proven the GA-based SSA, and machine learning method in general, to be a novel, positive addition to the armamentarium of drug discovery methods in medicinal chemistry.

Appendix A: The GA-based SSA pseudocode

```
// Main program execution
set chromosomes_size, num_of_weights, min_weight_value, max_weight_value
set crossover_rate, mutation_rate, maximum_iterations, elite_chromosomes_to_keep
initialise ideal_fitness_value // function InitIdealFitness
initialise chromosomes // function AssignChromosomes

REPEAT
  FOR (parents_to_keep + 1) to end_of_chromosomes
    // function SelectTwoParents
    select two chromosomes from chromosomes
      // function PerformCrossover
      if (crossover_rate) perform crossover operation
      // function PerformMutation
      if (mutation_rate) perform mutation operation
      insert chromosome_child into new_chromosomes
  ENDFOR
  insert chromosome_1 to elite_chromosomes_to_keep into new_chromosomes

  // function CalculateFitness
  calculate new_chromosomes_fitness
  // function EvalFitness
  evaluate new_chromosomes_fitness
  sort new_chromosomes based on new_chromosomes_fitness
  set chromosomes as new_chromosomes
UNTIL iteration equals maximum_iterations or fitness_condition met

// Function to randomise the chromosomes initial values
Function AssignChromosomes(chromosomes_size, num_of_weights, min_weight_value, max_weight_value)
  set weight_polarity based on SSA R4 weight value
  set chromosomes as randomised array based on chromosomes_size, ... num_of_weights, min_weight_value and
  max_weight_value and weight_polarity

// Function to calculate the most ideal active rate (100% active at the very top)
Function InitIdealFitness(compounds)
  set ideal_fitness_value as the maximum number of active compounds in the top 1% of the given dataset

// Function to select two parents for GA operation, using roulette wheel method
Function SelectTwoParents(chromosomes)
  set total_fitness as sum of chromosome_fitness
  set a roulette_value from a random number not more than total_fitness
  select chromosome_parent1 where roulette_value points at in total_fitness
  select chromosome_parent2 where roulette_value points at in total_fitness
  and not equal to chromosome_parent1

// Function to perform GA crossover operation
Function PerformCrossover(chromosome_parent1, chromosome_parent2)
  set index from a random number not more than num_of_weights
  set order_of_crossover from a random condition of either parent1 or parent2
  first
  IF order_of_crossover equals left
    set chromosome_child as joining of chromosome_parent1(1,index) and ...
    chromosome_parent2(index+1,end)
  ELSE
    set chromosome_child as joining of chromosome_parent2(1,index) and ...
    chromosome_parent1(index+1,end)
  ENDIF

// Function to perform GA mutation operation
Function PerformMutation(chromosome_child)
  set index from a random number not more than num_of_weights
```

```

    set index of chromosome_child to a random, new weight value based on
        weight_polarity

// Function to calculate chromosomes fitness
Function CalculateFitness(new_chromosomes, compounds)
    FOR each new_chromosomes
        FOR each compound
            set compound_score as compound * chromosome
        ENDFOR
        sort compounds based on chromosome_score
        set new_chromosome_fitness as the rate of active compounds in sorted
            top 1%
    ENDFOR

// Function to evaluate chromosomes fitness
Function EvalFitness(new_chromosomes, fitness_score)
    sort new_chromosomes based on fitness_score
    FOR each new_chromosomes
        if (new_chromosome_fitness >= ideal_fitness_value) terminate program
    ENDFOR

```

Appendix B: The GP-based SSA pseudocode

```
// Main program execution
set chromosomes_size, terminal_set_definition, function_set_definition
set tree_method, maximum_depth, maximum_size
set crossover_rate, mutation_rate, maximum_iterations, elite_chromosomes_to_keep
initialise ideal_fitness_value //function InitIdealFitness
initialise chromosomes //function AssignChromosomes

REPEAT
  FOR (parents_to_keep + 1) to end_of_chromosomes
    //function SelectTwoParents
    select two chromosomes from chromosomes
      //function PerformCrossover
      if (crossover_rate) perform crossover operation
      //function PerformMutation
      if (mutation_rate) perform mutation operation
      insert chromosome_child into new_chromosomes
  ENDFOR
  insert chromosome_1 to elite_chromosomes_to_keep into new_chromosomes

  //function CalculateFitness
  calculate new_chromosomes_fitness
  //function EvalFitness
  evaluate new_chromosomes_fitness
  sort new_chromosomes based on new_chromosomes_fitness
  set chromosomes as new_chromosomes
UNTIL iteration equals maximum_iterations or fitness_condition met

// Function to randomise the chromosomes initial values
Function AssignChromosomes(chromosomes_size, terminal_set_definition, ... function_set_definition, tree_method)
  set chromosomes as randomised equation based on chromosomes_size, ... terminal_set_definition,
  function_set_definition and tree_method

// Function to calculate the most ideal active rate (100% active at the very top)
Function InitIdealFitness(compounds)
  set ideal_fitness_value as the maximum number of active compounds in the top 1% of the given dataset

// Function to select two parents for GP operation, using roulette wheel method
Function SelectTwoParents(chromosomes)
  set total_fitness as sum of chromosome_fitness
  set a roulette_value from a random number not more than total_fitness
  select chromosome_parent1 where roulette_value points at in total_fitness
  select chromosome_parent2 where roulette_value points at in total_fitness
  and not equal to chromosome_parent1

// Function to perform GP crossover operation
Function PerformCrossover(chromosome_parent1, chromosome_parent2)
  set index from a random number not more than maximum_depth
  set order_of_crossover from a random condition of either parent1 or parent2
  first
  IF order_of_crossover equals left
    set chromosome_child as joining of chromosome_parent1(1,index) and
    chromosome_parent2(index+1,end)

  ELSE
    set chromosome_child as joining of chromosome_parent2(1,index) and
    chromosome_parent1(index+1,end)
  ENDF

// Function to perform GP mutation operation
Function PerformMutation(chromosome_child)
```

```

    set index from a random number not more than maximum_depth
    set index of chromosome_child to a random, new equation

// Function to calculate chromosomes fitness
Function CalculateFitness(new_chromosomes, compounds)
  FOR each new_chromosomes
    translate equation to weights
    FOR each compound
      set compound_score as compound * weights
    ENDFOR
    sort compounds based on chromosome_score
    set new_chromosome_fitness as the rate of active compounds in sorted
      top 1%
  ENDFOR

// Function to evaluate chromosomes fitness
Function EvalFitness(new_chromosomes, fitness_score)
  sort new_chromosomes based on fitness_score
  FOR each new_chromosomes
    if (new_chromosome_fitness >= ideal_fitness_value) terminate program
  ENDFOR

```

REFERENCES

- Abdo, A., Chen, B., Mueller, C., Salim, N., & Willett, P. (2010). Ligand-based virtual screening using Bayesian networks. *Journal of Chemical Information and Modeling*, 50(6), 1012-1020. doi:10.1021/ci100090p
- Adamson, G. W., Cowell, J., Lynch, M. F., McLure, A. H., Town, W. G., & Yapp, A. M. (1973). Strategic considerations in the design of a screening system for substructure searches of chemical structure files. *Journal of Chemical Documentation*, 13(3), 153-157. doi:10.1021/c160050a013
- Angelopoulos, N., Hadjiprocopis, A., & Walkinshaw, M. D. (2009). Bayesian model averaging for ligand discovery. *Journal of Chemical Information and Modeling*, 49(6), 1547-1557. doi: 10.1021/ci900046u
- Anzali, S., Barnickel, G., Cezanne, B., Krug, M., Filimonov, D., & Poroikov, V. (2001). Discriminating between drugs and nondrugs by prediction of activity spectra for substances (PASS). *Journal of Medicinal Chemistry*, 44(15), 2432-2437. doi:10.1021/jm0010670
- Arif, S. M., Holliday, J. D., & Willett, P. (2009). Analysis and use of fragment-occurrence data in similarity-based virtual screening. *Journal of Computer-Aided Molecular Design*, 23(9), 655-668. doi:10.1007/s10822-009-9285-0
- Avidon, V. V., Pomerantsev, I. A., Golender, V. E., & Rozenblit, A. B. (1982). Structure-activity relationship oriented languages for chemical-structure representation. *Journal of Chemical Information and Computer Sciences*, 22(4), 207-214. doi:10.1021/ci00036a006
- Azaria, Y., & Sipper, M. (2005). GP-gammon: Genetically programming backgammon players. *Genetic Programming and Evolvable Machines*, 6(3), 283-300. doi: 10.1007/s10710-005-2990-0
- Bajorath, J. (2001). Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *Journal of Chemical Information and Computer Sciences*, 41(2), 233-245. doi:10.1021/Ci0001482
- Balaban, A. T. (1982). Highly discriminating distance-based topological index. *Chemical Physics Letters*, 89(5), 399-404. doi:10.1016/0009-2614(82)80009-2
- Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming: an introduction (Vol. 1)*. San Francisco: Morgan Kaufmann.

- Barnard, J. M., & Downs, G. M. (1997). Chemical fragment generation and clustering software. *Journal of Chemical Information and Computer Sciences*, 37(1), 141-142. doi:10.1021/ci960090k
- Bawden, D. (1993). Molecular dissimilarity in chemical information systems. In *Chemical Structures 2*, (pp. 383-388). The Netherlands: Springer Berlin Heidelberg.
- Bawden, D., Catlow, J. T., Devon, T. K., Dalton, J. M., Lynch, M. F., & Willett, P. (1981). Evaluation and implementation of topological codes for online compound search and registration. *Journal of Chemical Information and Computer Sciences*, 21(2), 83-86.
- Bayley, M. J., Jones, G., Willett, P., & Williamson, M. P. (1998). GENFOLD: A genetic algorithm for folding protein structures using NMR restraints. *Protein Science*, 7(2), 491-499. doi:10.1002/pro.5560070230
- Belkin, N. J., Kantor, P., Fox, E. A., & Shaw, J. A. (1995). Combining the evidence of multiple query representations for information retrieval. *Information Processing & Management*, 31(3), 431-448. doi:10.1016/0306-4573(94)00057-A
- Benbassat, A., & Sipper, M. (2010). Evolving lose-checkers players using genetic programming. *Computational Intelligence and Games (CIG), 2010 IEEE Symposium*, 30-37. doi: 10.1109/ITW.2010.5593376
- Bender, A., Jenkins, J. L., Scheiber, J., Sukuru, S. C. K., Glick, M., & Davies, J. W. (2009). How similar are similarity searching methods? A principal component analysis of molecular descriptor space. *Journal of Chemical Information and Modeling*, 49(1), 108-119. doi:10.1021/ci800249s
- Bender, A., Mussa, H. Y., Glen, R. C., & Reiling, S. (2004). Molecular similarity searching using atom environments, information-based feature selection, and a naive Bayesian classifier. *Journal of Chemical Information and Computer Sciences*, 44(1), 170-178. doi:10.1021/Ci034207y
- Bender, A., Mussa, H. Y., Glen, R. C., & Reiling, S. (2004). Similarity searching of chemical databases using atom environment descriptors (MOLPRINT 2D): evaluation of performance. *Journal of Chemical Information and Computer Sciences*, 44(5), 1708-1718. doi:10.1021/ci0498719
- Birchall, K. (2011). *Reduced graph approaches to analysing high-throughput screening data*. (PhD thesis, University of Sheffield, Sheffield, United Kingdom).
- Bishop, N., Gillet, V. J., Holliday, J. D., & Willett, P. (2003). Chemoinformatics research at the University of Sheffield: a history and citation analysis. *Journal of Information Science*, 29(4), 249-267. doi:10.1177/01655515030294003

- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.
- Brown, N., McKay, B., Gilardoni, F., & Gasteiger, J. (2004). A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of chemical information and computer sciences*, *44*(3), 1079-1087.
- Brown, R. D., Jones, G., Willett, P., & Glen, R. C. (1994). Matching two-dimensional chemical graphs using genetic algorithms. *Journal of Chemical Information and Computer Sciences*, *34*(1), 63-70. doi:10.1021/ci00017a008
- Byvatov, E., Fechner, U., Sadowski, J., & Schneider, G. (2003). Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences*, *43*(6), 1882-1889. doi: 10.1002/chin.200405237
- Cambridge Crystallographic Data Centre. (2015, May 27). Retrieved from <http://www.ccdc.cam.ac.uk/pages/Home.aspx>
- Capelli, A. M., Feriani, A., Tedesco, G., & Pozzan, A. (2006). Generation of a focused set of GSK compounds biased toward ligand-gated ion-channel ligands. *Journal of Chemical Information and Modeling*, *46*(2), 659-664. doi:10.1021/ci050353n
- Carhart, R. E., Smith, D. H., & Venkataraghavan, R. (1985). Atom pairs as molecular-features in structure activity studies - definition and applications. *Journal of Chemical Information and Computer Sciences*, *25*(2), 64-73. doi:10.1021/ci00046a002
- CAS. (2016, January 10). Retrieved from <http://www.cas.org/>
- Chakravarti, S., Saiakhov, R., & Klopman, G. (2012). Optimizing predictive performance of CASE Ultra expert system models using the applicability domains of individual toxicity alerts. *Journal of Chemical Information and Modeling*, *52*(10), 2609-2618. doi:10.1021/ci300111r
- Clark, D. E. (1999). Rapid calculation of polar molecular surface area and its application to the prediction of transport phenomena. 1. Prediction of intestinal absorption. *Journal of Pharmaceutical Sciences*, *88*(8), 807-814.
- Clark, R. D., & Webster-Clark, D. J. (2008). Managing bias in ROC curves. *Journal of computer-aided molecular design*, *22*(3-4), 141-146. doi: 10.1007/s10822-008-9181-z
- Chen, B., Mueller, C., & Willett, P. (2010). Combination rules for group fusion in similarity-based virtual screening. *Molecular Informatics*, *29*(6-7), 533-541. doi:10.1002/minf.201000050

- Chen, J., Holliday, J., & Bradshaw, J. (2009). A machine learning approach to weighting schemes in the data fusion of similarity coefficients. *Journal of chemical information and modeling*, 49(2), 185-194. doi: 10.1021/ci800292d
- Chen, S. H., & Yeh, C. H. (2002). On the emergent properties of artificial stock markets: the efficient market hypothesis and the rational expectations hypothesis. *Journal of Economic Behavior & Organization*, 49(2), 217-239. doi: 10.1016/S0167-2681(02)00068-9
- Chen, S. H., & Liao, C. C. (2005). Agent-based computational modeling of the stock price–volume relation. *Information Sciences*, 170(1), 75-100. doi: 10.1016/j.ins.2003.03.026
- Cormack, G. V., Clarke, C. L., & Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 758-759. doi:10.1145/1571941.1572114
- Cosgrove, D. A., & Willett, P. (1998). SLASH: A program for analysing the functional groups in molecules. *Journal of Molecular Graphics & Modelling*, 16(1), 19-32. doi:10.1016/s1093-3263(98)00014-x
- Cottrell, S., J., Gillet, V., J., Taylor, R. & Wilton, D., J. (2004). Generation of multiple pharmacophore hypothesis using multiobjective optimisation techniques. *Journal of Computer-Aided Molecular Design*, 18(11), 665-682. doi:10.1007/s10822-004-5523-7
- Cramer, R. D., Redl, G., & Berkoff, C. E. (1974). Substructural analysis - novel approach to problem of drug design. *Journal of Medicinal Chemistry*, 17(5), 533-535. doi:10.1021/jm00251a014
- Cummins, R., & O’Riordan, C. (2006). Evolved term-weighting schemes in Information Retrieval: an analysis of the solution space. *Artificial Intelligence Review*, 26(1-2), 35-47. doi: 10.1007/s10462-007-9034-5
- Cunningham, P., & Delany, S.J. (2007). *k-Nearest neighbour classifiers* (UCD-CSI-2007-4).
- Cruciani, G., Crivori, P., Carrupt, P. A., & Testa, B. (2000). Molecular fields in quantitative structure–permeation relationships: the VolSurf approach. *Journal of Molecular Structure: THEOCHEM*, 503(1), 17-30.
- Dahlin, J. L., & Walters, M. A. (2014). The essential roles of chemistry in high-throughput screening triage. *Future medicinal chemistry*, 6(11), 1265-1290.
- Davis, L. (1985). Applying adaptive algorithms to epistatic domains. *Proceedings of the International Joint Conference on Artificial Intelligence*, 11, 162-164.

- Deconinck, E., Zhang, M. H., Coomans, D., & Vander Heyden, Y. (2006). Classification tree models for the prediction of blood-brain barrier passage of drugs. *Journal of Chemical Information and Modeling*, *46*(3), 1410-1419. doi:10.1002/chin.200631207
- Duch, W., Swaminathan, K., & Meller, J. (2007). Artificial intelligence approaches for rational drug design and discovery. *Current Pharmaceutical Design*, *13*(14), 1497-1508. doi:10.2174/138161207780765954
- Dos Santos, E. M., Sabourin, R., & Maupin, P. (2009). Overfitting cautious selection of classifier ensembles with genetic algorithms. *Information Fusion*, *10*(2), 150-162. doi:10.1016/j.inffus.2008.11.003
- Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87. doi:10.1145/2347736.2347755
- Durant, J. L., Leland, B. A., Henry, D. R., & Nourse, J. G. (2002). Reoptimisation of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, *42*(6), 1273-1280. doi:10.1021/ci010132r
- Eberhart, R. C., & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. *Evolutionary Programming VII* (pp. 611-616). Berlin Heidelberg: Springer.
- Ertl, P., Roggo, S., & Schuffenhauer, A. (2008). Natural product-likeness score and its application for prioritization of compound libraries. *Journal of Chemical Information and Modeling*, *48*(1), 68-74. doi:10.1021/Ci700286x
- Fan, W., Fox, E. A., Pathak, P., & Wu, H. (2004). The effects of fitness functions on genetic programming-based ranking discovery for Web search. *Journal of the American Society for Information Science and Technology*, *55*(7), 628-636. doi:10.1002/asi.20009
- Fernandez, M., Caballero, J., Fernandez, L., & Sarai, A. (2011). Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM). *Molecular diversity*, *15*(1), 269-289.
- Fix, E., & Hodges Jr, J. L. (1951). *Discriminatory analysis-nonparametric discrimination: consistency properties*. California, University Berkeley. doi:10.2307/1403797
- Fjell, C. D., Jenssen, H., Cheung, W. A., Hancock, R. E., & Cherkasov, A. (2011). Optimization of antibacterial peptides by genetic algorithms and cheminformatics. *Chemical biology & drug design*, *77*(1), 48-56.

- Frearson, J. A., & Collie, I. T. (2009). HTS and hit finding in academia - from chemical genomics to drug discovery. *Drug Discovery Today*, 14(23-24), 1150-1158. doi:10.1016/j.drudis.2009.09.004
- Fogel, D. B. (1995, November). Phenotypes, genotypes, and operators in evolutionary computation. In *Evolutionary Computation, 1995., IEEE International Conference on* (Vol. 1, p. 193). IEEE
- Gardiner, E. J., Gillet, V. J., Haranczyk, M., Hert, J., Holliday, J. D., Malim, N., Patel, Y., & Willett, P. (2009). Turbo similarity searching: Effect of fingerprint and dataset on virtual-screening performance. *Statistical Analysis and Data Mining*, 2(2), 103-114. doi:10.1002/sam.10037
- Gardiner, E. J., Willett, P., & Artymiuk, P. J. (2001). Protein docking using a genetic algorithm. *Proteins: Structure, Function, and Bioinformatics*, 44(1), 44-56. doi:10.1002/prot.1070
- Gardiner, E. J., Willett, P., & Artymiuk, P. J. (2003). GAPDOCK: A genetic algorithm approach to protein docking in CAPRI round 1. *Proteins: Structure, Function, and Bioinformatics*, 52(1), 10-14. doi:10.1002/prot.10386
- Gasteiger, J., & Engel, T. (2006). *Chemoinformatics*: Wiley-VCH.
- Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B. & Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1), D1100-D1107. doi: 10.1093/nar/gkr777
- Gillet, V. J. (2004). Designing combinatorial libraries optimized on multiple objectives. *Methods in Molecular Biology*, 275, 335-354. doi: 10.1385/1-59259-802-1:335
- Gillet, V. J., Willett, P., & Bradshaw, J. (1997). The effectiveness of reactant pools for generating structurally-diverse combinatorial libraries. *Journal of Chemical Information and Computer Sciences*, 37(4), 731-740.
- Gillet, V. J., Willett, P., Bradshaw, J., & Green, D. V. (1999). Selecting combinatorial libraries to optimize diversity and physical properties. *Journal of Chemical Information and Computer Sciences*, 39(1), 169-177. doi:10.1021/ci980332b
- Gillet, V. J., Willett, P., & Bradshaw, J. (1998). Identification of biological activity profiles using substructural analysis and genetic algorithms. *Journal of Chemical Information and Computer Sciences*, 38(2), 165-179. doi:10.1021/ci970431+

- Gillet, V. J., Khatib, W., Willett, P., Fleming, P. J., & Green, D. V. (2002). Combinatorial library design using a multiobjective genetic algorithm. *Journal of Chemical Information and Computer Sciences*, 42(2), 375-385. doi:10.1021/ci010375j
- Ginn, C.M.R., Willett, P. & Bradshaw, J. (2000). Combination of molecular similarity measures using data fusion. *Perspectives in Drug Discovery and Design*, 20 (1), 1-16. doi:10.1007/0-306-46883-2_1
- Glen, R. C., Bender, A., Arnby, C. H., Carlsson, L., Boyer, S., & Smith, J. (2006). Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. *IDrugs : The Investigational Drugs Journal*, 9(3), 199-204.
- Gleeson, M. P., Waters, N. J., Paine, S. W., & Davis, A. M. (2006). In silico human and rat V ss quantitative structure-activity relationship models. *Journal of Medicinal Chemistry*, 49(6), 1953-1963. doi: 10.1021/jm051007d
- Glick, M., Klon, A. E., Acklin, P., & Davies, J. W. (2004). Enrichment of extremely noisy high-throughput screening data using a naive Bayes classifier. *Journal of Biomolecular Screening*, 9(1), 32-36. doi:10.1177/1087057103260590
- Glick, M., Jenkins, J. L., Nettles, J. H., Hitchings, H., & Davies, J. W. (2006). Enrichment of high-throughput screening data with increasing levels of noise using support vector machines, recursive partitioning, and Laplacian-modified naive Bayesian classifiers. *Journal of Chemical Information and Modeling*, 46(1), 193-200. doi:10.1021/ci050374h
- Globus, A., Lawton, J., & Wipke, T. (1999). Automatic molecular design using evolutionary techniques. *Nanotechnology*, 10(3), 290. doi:10.1088/0957-4484/10/3/312/meta
- Goldberg, D. E., Deb, K., & Clark, J. H. (1991). Genetic algorithms, noise, and the sizing of populations. *Complex systems*, 6, 333-362. Retrieved from <http://www.cse.msu.edu/~cse848/2011/Popsizing.pdf>
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2), 95-99.
- Güner, O. F. (2000). *Pharmacophore perception, development, and use in drug design*. LaJolla, Calif.: International University Line.
- Hall, D.L. & McMullen, S.A.H. (2004). *Mathematical techniques in multisensor data fusion*. Boston: Artech House.

- Halperin, I., Ma, B., Wolfson, H., & Nussinov, R. (2002). Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Bioinformatics*, 47(4), 409-443. doi:10.1002/prot.10115
- Handley, S. (1993). Automated learning of a detector for alpha-helices in protein sequences via genetic programming. *IEEE World Congress on Computational Intelligence*, 474-479. doi: 10.1109/ICEC.1994.349904
- Hansch, C. (1969). A quantitative approach to biochemical structure-activity relationships. *Accounts of Chemical Research*, 2(8), 232-&. doi:10.1021/ar50020a002
- Harper, G., Bradshaw, J., Gittins, J. C., Green, D. V. S., & Leach, A. R. (2001). Prediction of biological activity for high-throughput screening using binary kernel discrimination. *Journal of Chemical Information and Computer Sciences*, 41(5), 1295-1300. doi:10.1021/Ci000397q
- Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1), 1-12. doi:10.1021/ci0342472
- Heikamp, K., & Bajorath, J. (2011). Large-scale similarity search profiling of ChEMBL compound data sets. *Journal of Chemical Information and Modeling*, 51(8), 1831-1839. doi: 10.1021/ci200199u
- Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D., & Pletnev, I. (2013). InChI - The worldwide chemical structure identifier standard. *Journal of Cheminformatics*, 5(1), 7. doi:10.1186/1758-2946-5-7
- Hert, J., Willett, P., Wilton, D. J., Acklin, P., Azzaoui, K., Jacoby, E., & Schuffenhauer, A. (2006). New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *Journal of Chemical Information and Modeling*, 46(2), 462-470. doi:10.1021/ci050348j
- Hert, J., Willett, P., Wilton, D.J., Acklin, P., Azzaoui, K., Jacoby, E. & Schuffenhauer, A. (2004a). Comparison of fingerprint-based methods for virtual screening using multiple bioactive reference structures. *Journal of Chemical Information and Computer Sciences*, 44(3), 1177-1185. doi: 10.1021/ci034231b
- Hert, J., Willett, P., Wilton, D. J., Acklin, P., Azzaoui, K., Jacoby, E., & Schuffenhauer, A. (2004b). Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Organic & Biomolecular Chemistry*, 2(22), 3256-3266. doi: 10.1039/B409865J
- Hiemstra, D. (2001). *Using language models for information retrieval*. Enschede, The Netherlands: Taaluitgeverij Neslia Paniculata.

- Hodes, L. (1976). Selection of descriptors according to discrimination and redundancy. Application to chemical structure searching. *Journal of Chemical Information and Computer Sciences*, 16(2), 88-93. doi:10.1021/ci60006a012
- Hodes, L. (1981). Computer-aided selection of compounds for antitumor screening: validation of a statistical-heuristic method. *Journal of Chemical Information and Computer Sciences*, 21(3), 128-132. doi:10.1021/ci00031a003
- Hodes, L. (1981). Selection of molecular fragment features for structure-activity studies in antitumor screening. *Journal of Chemical Information and Computer Sciences*, 21(3), 132-136. doi:10.1021/ci00031a004
- Hodes, L., Hazard, G., Geran, R., & Richman, S. (1977). A statistical-heuristic method for automated selection of drugs for screening. *Journal of Medicinal Chemistry*, 20(4), 469-475. doi:10.1021/jm00214a002
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Machigan: M.I.T.P.
- Holliday, J. D., Ranade, S. S., & Willett, P. (1995). A fast algorithm for selecting sets of dissimilar molecules from large chemical databases. *Quantitative Structure-Activity Relationships*, 14(6), 501-506.
- Holliday, J. D., & Willett, P. (1996). Definitions of "dissimilarity" for dissimilarity-based compound selection. *Journal of Biomolecular Screening*, 1(3), 145-151.
- Holliday, J. D., & Willett, P. (1997). Using a genetic algorithm to identify common structural features in sets of ligands. *Journal of Molecular Graphics and Modelling*, 15(4), 221-232. doi:10.1016/S1093-3263(97)00080-6
- Horai, H., Arita, M., Kanaya, S., Nihei, Y., Ikeda, T., Suwa, K., ... & Oda, Y. (2010). MassBank: a public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry*, 45(7), 703-714. doi:10.1002/jms.1777
- Hou, T., Wang, J., & Li, Y. (2007). ADME evaluation in drug discovery. 8. The prediction of human intestinal absorption by a support vector machine. *Journal of Chemical Information and Modeling*, 47(6), 2408-2415. doi: 10.1021/ci7002076
- Hu, Y., Lounkine, E., & Bajorath, J. (2009). Improving the search performance of extended connectivity fingerprints through activity-oriented feature filtering and application of a bit-density-dependent similarity function. *ChemMedChem*, 4(4), 540-548. doi:10.1002/cmdc.200800408
- James C, Weininger D (2006). *Daylight theory manual, version 4.9*. Retrieved from <http://www.daylight.com/dayhtml/doc/theory/index.html>

- Jaśkowski, W., Krawiec, K., & Wieloch, B. (2008). Winning ant wars: Evolving a human-competitive game strategy using fitnessless selection. *Genetic Programming* (pp 13-24). Berlin Heidelberg: Springer.
- Jenkins, J. L., Kao, R. Y., & Shapiro, R. (2003). Virtual screening to enrich hit lists from high-throughput screening: A case study on small-molecule inhibitors of angiogenin. *Proteins: Structure, Function, and Bioinformatics*, 50(1), 81-93.
- Johnson, M. A., & Maggiora, G. M. (1990). *Concepts and applications of molecular similarity*: Wiley New York.
- Jones, G. & Willett, P. (1995). Docking small molecule ligands into active sites. *Current Opinion in Biotechnology*, 6(6), 652-656. doi:10.1016/0958-1669(95)80107-3
- Jones, G., Willett, P. & Glen, R. C. (1995a). Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology*, 254(1), 43-53. doi:10.1016/0958-1669(95)80107-3
- Jones, G., Willett, P., & Glen, R. C. (1995b). A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *Journal of Computer-Aided Molecular Design*, 9(6), 532-549. doi:10.1007/BF00124324
- Jones, G., Willett, P., Glen, R. C., Leach, A. R., & Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267(3), 727-748. doi:10.1006/jmbi.1996.0897
- Jones, G., Willett, P., Glen, R. C., Leach, A. R., & Taylor, R. (1999). Further development of a genetic algorithm for ligand docking and its application to screening combinatorial libraries. In *ACS Symposium Series* (Vol. 719, pp. 271-291). doi:10.1021/bk-1999-0719.ch018
- Jorissen, R. N., & Gilson, M. K. (2005). Virtual screening of molecular databases using a support vector machine. *Journal of Chemical Information and Modeling*, 45(3), 549-561. doi: 10.1002/chin.200532197
- Kaboudan, M. (2005). Extended daily exchange rates forecasts using wavelet temporal resolutions. *New Mathematics and Natural Computation*, 1(01), 79-107. doi: 10.1142/S1793005705000056
- Kaya, M. (2011). The effects of two new crossover operators on genetic algorithm performance. *Applied Soft Computing*, 11(1), 881-890. doi:10.1016/j.asoc.2010.01.00
- Kaya, Y., Uyar, M., & Tekm R. (2011). A novel crossover operator for genetic algorithms: ring crossover. *arXiv preprint arXiv:1105.0355*.

- Khanna, V., & Ranganathan, S. (2011). Molecular similarity and diversity approaches in chemoinformatics. *Drug Development Research*, 72(1), 74-84. doi:10.1002/ddr.20404
- Klebe, G. (Ed.). (2000). *Virtual screening: An alternative or complement to high throughput screening*. Kluwer, Dordrecht: Springer.
- Klon, A. E., Glick, M., & Davies, J. W. (2004). Combination of a naive Bayes classifier with consensus scoring improves enrichment of high-throughput docking results. *Journal of Medicinal Chemistry*, 47(18), 4356-4359. doi:10.1021/jm049970d
- Klopman, G. (1984). Artificial-Intelligence approach to structure activity studies-compute automated structure evaluation of biological-activity of organic-molecules. *Journal of the American Chemical Society*, 106(24), 7315-7321. doi:10.1021/ja00336a004
- Klopman, G. (1992). A hierarchical computer automated structure evaluation program. *Quantitative Structure-Activity Relationships*, 11(2), 176-184. doi:10.1002/qsar.19920110208
- Klopman, G., & Kalos, A. N. (1985). Causality in structure-activity studies. *Journal of Computational Chemistry*, 6(5), 492-506. doi:10.1002/jcc.540060520
- Koza, J. R. (1992). *Genetic Programming: On the programming of computers by means of natural selection*. United State of America: MIT press.
- Koza, J. R., & Andre, D. (1996). Classifying protein segments as transmembrane domains using architecture-altering operations in genetic programming. *Advances in genetic programming*, 2, 155-176. doi: 10.1887/0750308958/b386c109
- Koza, J. R., Keane, M. A., & Streeter, M. J. (2004). Routine high-return human-competitive evolvable hardware. *Proceedings. 2004 NASA/DoD Conference IEEE*, 3-17. doi: 10.1109/EH.2004.1310803
- Kuczkowski, L., Kolendo, P., Jaworski, B. & Smierzchalski, R. (2012). Mean crossover in evolutionary path planning method for maritime collision avoidance. *Scientific Journals Maritime Academy of Szczecin*. 30(102), 70-77.
- Langdon, W. B., & Poli, R. (2002). *Foundations of genetic programming*. Berlin: Springer.
- Langer, T., Hoffmann, R., Bryant, S., & Lesur, B. (2009). Hit finding: towards 'smarter' approaches. *Current Opinion in Pharmacology*, 9(5), 589-593. doi:10.1016/j.coph.2009.06.001
- Lajiness, M. S. (1997). Dissimilarity-based compound selection techniques. *Perspectives in drug discovery and design*, 7, 65-84.
- Lamanna, C., Bellini, M., Padova, A., Westerberg, G., & Maccari, L. (2008). Straightforward recursive partitioning model for discarding insoluble compounds in the drug

- discovery process. *Journal of Medicinal Chemistry*, 51(10), 2891-2897. doi: 10.1021/jm701407x
- Lavecchia, A. (2015). Machine-learning approaches in drug discovery: methods and applications. *Drug Discovery Today*, 20(3), 318-331. doi:10.1016/j.drudis.2014.10.012
- Lavine, B. K., Davidson, C. E., & Moores, A. J. (2002). Innovative genetic algorithms for chemoinformatics. *Chemometrics and Intelligent Laboratory Systems*, 60(1), 161-171.
- Leach, A. R., & Gillet, V. J. (2007). *An Introduction to Chemoinformatics*. Dordrecht, Netherlands: Springer.
- Lengauer, T., Lemmen, C., Rarey, M., & Zimmermann, M. (2004). Novel technologies for virtual screening. *Drug Discovery Today*, 9(1), 27-34. doi:10.1016/s1359-6446(04)02939-3
- Leszczynski, J., & Shukla, M. K. (2009). *Practical aspects of computational chemistry: methods, concepts and applications*. New York: Springer.
- Li, M., Yang, S., & Liu, X. (2014). Diversity comparison of Pareto front approximations in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12), 2568-2584. doi: 10.1109/TCYB.2014.2310651
- Li, M., Yang, S. & Liu, X. (2015a). Pareto or non-pareto: Bi-criterion evolution in multi-objective optimization. *IEEE Transactions on Evolutionary Computation*. IEEE, 1(99), 1-1. doi:10.1109/TEVC.2015.2504730.
- Li, M., Yang, S., & Liu, X. (2015b). Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228, 45-65. doi: 10.1016/j.artint.2015.06.007
- Mannhold, R., Kubinyi, H., & Timmerman, H. (2008). *Evolutionary algorithms in molecular design*. D. E. Clark (Ed.). Germany: John Wiley & Sons
- Martin, E. J., Blaney, J. M., Siani, M. A., Spellmeyer, D. C., Wong, A. K., & Moos, W. H. (1995). Measuring diversity: experimental design of combinatorial libraries for drug discovery. *Journal of medicinal chemistry*, 38(9), 1431-1436.
- Maurer, W. D., & Lewis, T. G. (1975). Hash table methods. *ACM Computing Surveys*, 7(1), 5-19. doi:10.1145/356643.356645
- McPhee, N. F., Poli, R., & Langdon, W. B. (2008). *Field Guide to Genetic Programming*. Retrieved from <http://www.gp-field-guide.org.uk/>
- McInnes, C. (2007). Virtual screening strategies in drug discovery. *Current Opinion in Chemical Biology*, 11(5), 494-502. doi:10.1016/j.cbpa.2007.08.033

- Mendes, J., M (2013). A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. *WSEAS Transactions on Computers*, 12(4), 164-173.
- Mensch, J., Oyarzabal, J., Mackie, C., & Augustijns, P. (2009). In vivo, in vitro and in silico methods for small molecule transfer across the BBB. *Journal of Pharmaceutical Sciences*, 98(12), 4429-4468. doi:10.1002/jps.21745
- Mente, S. R., & Lombardo, F. (2005). A recursive-partitioning model for blood–brain barrier permeation. *Journal of Computer-Aided Molecular Design*, 19(7), 465-481. doi: 10.1007/s10822-005-9001-7
- Merlot, C., Domine, D., Cleva, C., & Church, D. J. (2003). Chemical substructures in drug discovery. *Drug Discovery Today*, 8(13), 594-602. doi:10.1016/s1359-6446(03)02740-5
- Meyer, E., & Sens, E. (1988). Kowist: a computer-assisted systematic approach to finding new compound classes of pesticides. *Analytica Chimica Acta*, 210, 135-142. doi:10.1016/S0003-2670(00)83885-4
- Mitchell, M. (1998). *An introduction to genetic algorithms*. United State of America: MIT press.
- Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45. doi: 10.1007/978-1-4613-2279-5
- Nachbar, R. B. (2000). Molecular evolution: automated manipulation of hierarchical chemical topology and its application to average molecular structures. *Genetic Programming and Evolvable Machines*, 1(1-2), 57-94. doi:10.1023/A:1010072431120
- Nasr, R. J., Swamidass, S. J., & Baldi, P. F. (2009). Large scale study of multiple-molecule queries. *Journal of Cheminformatics*, 1(1), 7. doi:10.1186/1758-2946-1-7
- Nicolaou, C. A. & Brown, N. (2013). Multi-objective optimization methods in drug design. *Drug Discovery Today: Technologies*. 10(3), 427-435. doi: 10.1016/j.ddtec.2013.02.001
- Nicolotti, O., Gillet, V. J., Fleming, P. J., & Green, D. V. (2002). Multiobjective optimization in quantitative structure-activity relationships: Deriving accurate and interpretable QSARs. *Journal of Medicinal Chemistry*, 45(23), 5069-5080. doi:10.1021/jm020919o
- Novartis. (2012). Novartis . Available: <http://www.novartis.com> [Accessed 2012].
- Nuray, R., & Can, F. (2006). Automatic ranking of information retrieval systems using data fusion. *Information Processing & Management*, 42(3), 595-614. doi:10.1016/j.ipm.2005.03.023

- O'Boyle, N. M. (2012). Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics*, 4 (22). doi: 10.1186/1758-2946-4-22
- Olah, M., Bologa, C. & Oprea, T.I. (2004). An automated PLS search for biologically relevant QSAR descriptors. *Journal of Computer-Aided Molecular Design*, 18 (7), 437-449. doi: 10.1007/s10822-004-4060-8
- Oprea, T. I., & Matter, H. (2004). Integrating virtual screening in lead discovery. *Current Opinion in Chemical Biology*, 8(4), 349-358. doi:10.1016/j.cbpa.2004.06.008
- Ott, L., Longnecker, M., & Ott, R. L. (2001). *An introduction to statistical methods and data analysis* (Vol. 511). Pacific Grove, CA: Duxbury.
- Ormerod, A., Willett, P., & Bawden, D. (1989). Comparison of fragment weighting schemes for substructural analysis. *Quantitative Structure-Activity Relationships*, 8(2), 115-129. doi:10.1002/qsar.19890080207
- Panda, M., & Patra, M. R. (2007). Network intrusion detection using naive bayes. *International Journal of Computer Science and Network Security*, 7(12), 258-263. doi:10.1.1.128.936&rep=rep1&type=pdf
- Poirrette, A. R., Artymiuk, P. J., Rice, D. W., & Willett, P. (1997). Comparison of protein surfaces using a genetic algorithm. *Journal of Computer-Aided Molecular Design*, 11(6), 557-569. doi:10.1023/A:1007966011516
- Poli, R., & Langdon, W. B. (1998). Schema theory for genetic programming with one-point crossover and point mutation. *Evolutionary Computation*, 6(3), 231-252. doi: 10.1162/evco.1998.6.3.231
- Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). *A field guide to genetic programming*. California, USA: Creative Commons.
- Prabhakar, Y. S., & Gupta, M. K. (2008). Chemical structure indices in in silico molecular design. *Scientia Pharmaceutica*, 76(2), 101. doi:10.3797/scipharm.0804-12
- Randic, M. (1975). Characterization of molecular branching. *Journal of the American Chemical Society*, 97(23), 6609-6615. doi:10.1021/ja00856a001
- Ras, Z. W., & Michalewicz, M. (1996). *Foundations of Intelligent Systems: 9th International Symposium, ISMIS'96, Zakopane, Poland, June (9-13), 1996. Proceedings* (Vol. 9). Springer Science & Business Media.
- Ray, L. C., & Kirsch, R. A. (1957). Finding chemical records by digital computers. *Science*, 126 (3278), 814-819. doi:10.1126/science.126.3278.814

- Reddy, A. S., Pati, S. P., Kumar, P. P., Pradeep, H. N., & Sastry, G. N. (2007). Virtual screening in drug discovery - A computational perspective. *Current Protein & Peptide Science*, 8(4), 329-351. doi:10.2174/138920307781369427
- Robertson, A. M., & Willett, P. (1996). An upperbound to the performance of ranked-output searching: optimal weighting of query terms using a genetic algorithm. *Journal of Documentation*, 52(4), 405-420. doi:10.1108/eb026973
- Robertson, S. E., & Sparck-Jones, K. (1976). Relevance Weighting of Search Term. *Journal of the American Society for Information Science*, 27(3), 129-146. doi:10.1002/asi.4630270302
- Roeva, O., Fidanova, S., & Paprzycki, M. (2014). Population Size Influence on the Genetic and Ant Algorithms Performance in Case of Cultivation Process Modeling. In F. Stefka (Ed.). *Recent Advances in Computational Optimization*, (pp. 107-120). Switzerland: Springer International Publishing.
- Rogers, D., Brown, R. D., & Hahn, M. (2005). Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up. *Journal of Biomolecular Screening*, 10(7), 682-686. doi:10.1177/1087057105281365
- Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5), 742-754. doi:10.1021/ci100050t
- Saiakhov, R., Chakravarti, S., & Klopman, G. (2013). Effectiveness of CASE ultra expert system in evaluating adverse effects of drugs. *Molecular Informatics*, 32(1), 87-97. doi:10.1002/minf.201200081
- Sakiyama, Y., Yuki, H., Moriya, T., Hattori, K., Suzuki, M., Shimada, K., & Honma, T. (2008). Predicting human liver microsomal stability with machine learning techniques. *Journal of Molecular Graphics and Modelling*, 26(6), 907-915. doi:10.1016/j.jm gm.2007.06.005
- Salim, N., Holliday, J. & Willett, P. (2003). Combination of fingerprint-based similarity coefficients using data fusion. *Journal of Chemical Information and Computer Sciences*, 43(2), 435-442. doi:10.1021/ci025596j
- Samanidou, E., Zschischang, E., Stauffer, D., & Lux, T. (2007). Agent-based models of financial markets. *Reports on Progress in Physics*, 70(3), 409. doi: 10.1088/0034-4885/70/3/R03
- Samuel, A. L. (1983). AI, Where It Has Been and Where It Is Going. *Proceeding of the Eighth International Joint Conference on Artificial Intelligence*, 129-174.

- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229. doi:10.1147/rd.441.0206.
- Sekhar, Y. N., Nayana, M., Sivakumari, N., Ravikumar, M., & Mahmood, S. K. (2008). 3D-QSAR and molecular docking studies of 1, 3, 5-triazene-2, 4-diamine derivatives against r-RNA: Novel bacterial translation inhibitors. *Journal of Molecular Graphics and Modelling*, 26(8), 1338-1352. doi:10.1016/j.jmgm.2008.01.008
- Sherhod, R. (2011). *Development of a data mining tool for the identification of toxicophores*. (PhD thesis, University of Sheffield, Sheffield, United Kingdom).
- Sheridan, R. P. & Kearsley, S. K. (2002). Why do we need so many chemical similarity search methods?. *Drug Discovery Today*, 7(17), 903-911. doi:10.1016/S1359-6446(02)02411-X
- Shoichet, B. K. (2004). Virtual screening of chemical libraries. *Nature*, 432(7019), 862-865. doi:10.1038/nature03197
- Todeschini, R., & Consonni, V. (2008). *Handbook of molecular descriptors*. Germany: Wiley-vch.
- Tong, W., Welsh, W. J., Shi, L. M., Fang, H., & Perkins, R. (2003). Structure-activity relationship approaches and applications. *Environmental Toxicology and Chemistry*, 22(8), 1680-1695. doi:10.1897/01-198
- Tripos. In Tripos Molecular Holograms. Retrieved 3 March 2015, from <http://www.tripos.com/index.php>
- Tripos Inc. (2010). *Unity version 4.4.1*. Retrieved from <http://www.tripos.com/index.php>
- Trotman, A. (2005). Learning to rank. *Information Retrieval*, 8(3), 359-381. doi: 10.1007/s10791-005-6991-7
- Truchon, J. F., & Bayly, C. I. (2007). Evaluating virtual screening methods: good and bad metrics for the “early recognition” problem. *Journal of Chemical Information and Modeling*, 47(2), 488-508. doi: 10.1021/ci600426e
- Valencia, P., Haak, A., Cotillon, A., & Jurdak, R. (2014). Genetic programming for smart phone personalisation. *Applied Soft Computing*, 25, 86-96. doi: 10.1016/j.asoc.2014.08.058
- Vekaria, K. & Clack, C. (1998). Selective crossover in genetic algorithms: an empirical study. *Parallel Problem Solving from Nature—PPSN V*, 1498, 438-447. doi: 10.1007/Bfb0056843.
- Villar, H. O., Hansen, M. R., & Kho, R. (2007). Substructural analysis in drug discovery. *Current Computer-Aided Drug Design*, 3, 59-67. doi:10.2174/157340907780058745

- Vollmer, J. J. (1983). Wiswesser Line Notation: an introduction. *Journal of Chemical Education*, 60(3), 192. doi:10.1021/ed060p192
- Wale, N. (2011). Machine learning in drug discovery and development. *Drug Development Research*, 72(1), 112-119. doi:10.1002/ddr.20407
- Walters, W. P., Stahl, M. T., & Murcko, M. A. (1998). Virtual screening - an overview. *Drug Discovery Today*, 3(4), 160-178. doi:10.1016/s1359-6446(97)01163-x
- Wang, J., Krudy, G., Xie, X. Q., Wu, C., & Holland, G. (2006). Genetic algorithm-optimized QSPR models for bioavailability, protein binding, and urinary excretion. *Journal of chemical information and modeling*, 46(6), 2674-2683.
- Wang, L. (2005). *Support Vector Machines: theory and applications* (Vol. 177). Springer Science & Business Media.
- Wang, S., & Summers, R. M. (2012). Machine learning and radiology. *Medical Image Analysis*, 16(5), 933-951. doi: <http://dx.doi.org/10.1016/j.media.2012.02.005>
- Warmuth, M. K., Liao, J., Rätsch, G., Mathieson, M., Putta, S., & Lemmen, C. (2003). Active learning with support vector machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, 43(2), 667-673. doi: 10.1002/chin.200322232
- Wiener, H. (1947). Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1), 17-20. doi:10.1021/ja01193a005
- Weininger, D. (1988). SMILES, a chemical language and information system. *Journal of Chemical Information and Computer Sciences*, 28(1), 31-36. doi:10.1021/ci00057a005
- Whittaker, M., Law, R. J., Ichihara, O., & Hesterkamp, T. (2010). Fragments: Past, present and future. *Drug Discovery Today (DDT)*, 7, 163-171. doi:10.1016/j.ddtec.2010.11.007
- Whittle, M., Gillet, V.J., Willett, P., Alex, A. & Loesel, J. (2004). Enhancing the effectiveness of virtual screening by fusing nearest neighbor lists: A comparison of similarity coefficients. *Journal of Chemical Information and Computer Sciences*, 44 (5), 1840-1848. doi: 10.1021/ci049867x
- Whittle, M., Gillet, V.J., Willett, P. & Loesel, J. (2006). Analysis of data fusion methods in virtual screening: Similarity and group fusion. *Journal of Chemical and Information Modelling*, 46 (6), 2206-2219. doi: 10.1021/ci0496144
- Willett, P. (2013). Combination of similarity rankings using data fusion. *Journal of Chemical Information and Modeling*, 53(1), 1-10. doi: 10.1021/ci300547g

- Williams, C. (2006). Reverse fingerprinting, similarity searching by group fusion and fingerprint bit importance. *Molecular Diversity*, 10 (3), 311-332. doi: 10.1007/s11030-006-9039-z
- Wild, D. J., & Willett, P. (1996). Similarity searching in files of three-dimensional chemical structures. alignment of molecular electrostatic potential fields with a genetic algorithm. *Journal of Chemical Information and Computer Sciences*, 36(2), 159-167. doi:10.1021/ci9500851
- Willett, P. (1987). *Similarity and clustering in chemical information systems*. New York, USA: John Wiley & Sons, Inc..
- Willett, P. (2005). Searching techniques for databases of two- and three-dimensional chemical structures. *Journal of Medicinal Chemistry*, 48(13), 4183-4199. doi:10.1021/jm0582165
- Willett, P. (2009). Similarity methods in chemoinformatics. *Annual Review of Information Science and Technology*, 43, 3-71. doi:10.1002/aris.2009.1440430108
- Willett, P., Barnard, J. M., & Downs, G. M. (1998). Chemical similarity searching. *Journal of chemical information and computer sciences*, 38(6), 983-996.
- Willett, P. & Winterman, V. (1986). A comparison of some measures for the determination of inter-molecular structural similarity measures of inter-molecular structural similarity. *Quantitative Structure-Activity Relationships*, 5(1), 18-25.
- Wilton, D., Willett, P., Lawson, K., & Mullier, G. (2003). Comparison of ranking methods for virtual screening in lead-discovery programs. *Journal of Chemical Information and Computer Sciences*, 43(2), 469-474. doi:10.1021/ci025586i
- Wu, S., & Huang, C. (2014, July). Search result diversification via data fusion. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 827-830. doi: 10.1145/2600428.2609451
- Xia, X. Y., Maliski, E. G., Gallant, P., & Rogers, D. (2004). Classification of kinase inhibitors using a Bayesian model. *Journal of Medicinal Chemistry*, 47(18), 4463-4470. doi:10.1021/Jm0303195
- Yirka, B. (2015, November 12). *Computer scientist claims to have solved the graph isomorphism problem*. Retrieved from <http://phys.org/news/2015-11-scientist-graph-isomorphism-problem.html>
- Zernov, V. V., Balakin, K. V., Ivaschenko, A. A., Savchuk, N. P., & Pletnev, I. V. (2003). Drug discovery using support vector machines. The case studies of drug-likeness,

agrochemical-likeness, and enzyme inhibition predictions. *Journal of Chemical Information and Computer Sciences*, 43(6), 2048-2056. doi: 10.1002/chin.200405244