



The
University
Of
Sheffield.

Improving the Accuracy and Interpretability of Machine Learning Models for Toxicity Prediction

By

Moritz Walter

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

The University of Sheffield
Faculty of Social Sciences
Information School

October 2022

Acknowledgements

“Perseverance is not a long race; it is many short races one after the other.”

Walter Elliott

I would like to thank the people who supported me through all the short races of my PhD journey in the past three years.

First, my supervisors Professor Val Gillet, Dr. Antonio de la Vega de León and Dr. Sam Webb for their hard work, patience and trust. Whether supervised in person or remotely, I felt greatly supported at all times. Your knowledge and experience guided me through the highs and lows which one inevitably has to pass to reach the finish line.

I'm grateful for the members of the Sheffield Chemoinformatics group I met: Christina, Giammy, Jess, James (II), Zied, James (III), Hanz, Nahal, Terence and Savins. Thank you for the fruitful discussions in the meetings, your feedback to my work and the joy we had in the office, group lunches and abroad at conferences. In particular, I'd like to thank Luke Allen who supported my project with his outstanding Master dissertation. Thanks also to the other PhD students in the iSchool who form a pleasant community and the administrative staff who make things run smoothly in the department.

Also I'd like to thank everyone else being part of my life in Sheffield, especially my friends from DV and SLOG. I'll never forget the intense football matches, the pub, the hikes in the Peaks and the eventful trip to Hamburg. I could not have asked for a better group of mates.

Finally, I'm most grateful for my family and friends in Germany. I was provided with a wonderful home and constant support when working from Germany during phases of the COVID-19 pandemic.

Declaration

I, the author, confirm that the thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously presented for an award at this, or any other, university.

Luke Allen, a former MSc student at the Information School, carried out some preliminary experimental work on the ToxCast dataset and reported it in his Master dissertation. He used a subset of the ToxCast dataset to compare single task (random forest) and multi-task imputation techniques (random forest Feature Nets and Macau). He also tested the GHOST technique to adapt classification thresholds and analysed the relationship between assay similarity and increases in multi-task model performance for individual assays. This dissertation project was devised and supervised by me. For Chapter 6 of the thesis, I extended his work by using the entire ToxCast dataset, testing additional modelling techniques (XGBoost and multi-task DNNs), and performing the experiments on dataset sparsity (section 6.3.3) and selection of auxiliary assays (section 6.3.4).

Parts of the thesis (content from Chapter 5 and Chapter 6) have been published in the Journal of Cheminformatics (see below). I performed all the analyses described in Chapter 5 and the analyses in Chapter 6 were performed by Luke Allen and me as described above. I created the figures and produced the writing, while taking into account comments from my co-authors.

Publication:

Walter, M., Allen, L. N., de la Vega de León, A., Webb, S. J., & Gillet, V. J. (2022). Analysis of the benefits of imputation models over traditional QSAR models for toxicity prediction. *Journal of Cheminformatics*, 14, 32. <https://doi.org/10.1186/s13321-022-00611-w>

Abstract

Humans are exposed to a multitude of chemicals (e.g. pharmaceuticals and cosmetics) and the safety of these needs to be demonstrated. Quantitative structure-activity relationship (QSAR) models provide an alternative to undesired animal studies for this purpose. However, in practice their use is often limited either due to insufficient model accuracy or due to a lack of model interpretability.

This thesis addresses current limitations of QSAR models used for toxicity prediction. Firstly, it was investigated whether multi-task and imputation modelling yield more accurate models compared to standard single task QSAR models. Secondly, attempts were made to improve the interpretability of neural networks used for QSAR modelling. In particular, a method was developed to extract information about chemical features learned in the hidden layers of neural networks.

While no significant differences in performance were found between single task models and traditional multi-task models (using only chemical descriptors for test compounds), multi-task imputation models (using experimental data labels of related assays for test compounds) were found to clearly outperform single task models on in vitro toxicity datasets. Imputation is therefore a promising tool to improve the performance of QSAR models for toxicity prediction.

The novel method developed to interpret neural network models, called IG_hidden, makes use of integrated gradients to identify neurons relevant for individual predictions. Then, substructures found to be relevant for activation of these neurons are used to visualise which atoms of a compound are responsible for the model predictions. IG_hidden was compared to an established method for interpreting neural networks (i.e. applying integrated gradients to input features) using Lhasa's Derek alerts for mutagenicity as a ground truth. The overall performance of IG_hidden was found to be comparable to the published method in terms of the quality of the model explanations that were found. However, the approaches were complementary with each method performing better on certain subsets of the dataset.

Table of Contents

Acknowledgements.....	iii
Declaration.....	v
Abstract.....	vii
Abbreviations and acronyms	xiv
List of Figures	xvi
List of Tables	xix
Chapter 1 Introduction.....	1
Chapter 2 Toxicity assessment of chemicals	4
2.1 Introduction to Toxicology.....	4
2.2 Chemical risk assessment	5
2.2.1 Traditional risk assessment.....	5
2.2.2 Alternative methods for risk assessment	8
2.3 Mutagenicity	9
2.3.1 Introduction	9
2.3.2 Mutagenic chemicals	10
2.3.3 Ames test for mutagenicity.....	12
2.4 Conclusion.....	12
Chapter 3 Introduction to machine learning.....	14
3.1 Tasks in machine learning.....	14
3.2 The machine learning workflow.....	15
3.2.1 Data collection and curation.....	15
3.2.2 Model training and evaluation.....	16
3.2.3 Model deployment.....	18
3.3 Evaluation metrics.....	19
3.3.1 Regression models	19
3.3.2 Classification models.....	19
3.4 Machine learning algorithms	21
3.4.1 k-Nearest neighbours.....	21
3.4.2 Linear regression.....	22
3.4.3 Logistic regression.....	24
3.4.4 Support Vector Machines	24
3.4.5 Decision Trees	25
3.4.6 Random forest	27
3.4.7 Gradient tree boosting.....	28
3.4.8 Artificial neural networks and deep neural networks	30

3.4.9	Matrix factorisation	33
3.5	Conclusion.....	34
Chapter 4	QSAR modelling for toxicity prediction.....	35
4.1	Representation of chemical structures in computers	35
4.2	Molecular similarity	39
4.3	Basic principles of QSAR modelling.....	41
4.4	Preparing chemical structures for QSAR modelling.....	41
4.5	Toxicity data for QSAR modelling	42
4.5.1	In vitro toxicity data	42
4.5.2	In vivo toxicity data	43
4.6	Determinants of successful QSAR modelling.....	43
4.7	Use of QSAR models in regulatory toxicology	46
4.8	Use of neural networks in QSAR modelling	48
4.9	Conclusion.....	50
Chapter 5	Multi-task and imputation modelling for toxicity prediction	51
5.1	Introduction	51
5.1.1	Multi-task and imputation models	51
5.1.2	Multi-task models in QSAR modelling	52
5.1.3	Imputation models in QSAR modelling.....	54
5.1.4	Objectives.....	55
5.2	Methodology.....	55
5.2.1	Datasets	55
5.2.2	Data processing.....	57
5.2.3	Data splitting.....	58
5.2.4	Modelling.....	59
5.2.4.1	Single task models.....	60
5.2.4.2	Multi-task and imputation models	62
5.2.4.3	Model training.....	65
5.2.5	Model evaluation	65
5.2.6	Rationalisation of the imputation models' performances.....	66
5.2.6.1	Roles of chemical similarity.....	66
5.2.6.2	Role of data sparsity.....	67
5.2.6.3	Pairwise and leave-one-assay-out Feature Net models	67
5.2.7	Exploration of various classification thresholds for Macau models	68
5.3	Results.....	69
5.3.1	Single task models.....	69

5.3.2	Traditional multi-task models	70
5.3.3	Imputation models.....	76
5.3.4	Roll of chemical similarity in imputation models.....	78
5.3.5	Role of data sparsity in imputation models	82
5.3.6	Analysis on the impact of assay relatedness on imputation models.....	84
5.3.7	Analysis of the classification threshold used in Macau models.....	91
5.4	Discussion.....	94
5.4.1	Comparison of traditional single task and multi-task models	94
5.4.2	Comparison of single task and multitask imputation models	96
5.5	Conclusion.....	100
Chapter 6 Imputation on a large-scale toxicity dataset		101
6.1	Introduction	101
6.2	Methodology.....	101
6.2.1	Dataset	101
6.2.2	Model training and evaluation.....	102
6.2.3	GHOST methodology.....	102
6.2.4	Experiments on sparsity.....	103
6.2.5	Impact of assay relatedness on model performance.....	103
6.3	Results.....	104
6.3.1	Comparison of single task and multi-task imputation models	104
6.3.2	Using GHOST to adjust classification thresholds	105
6.3.3	Effect of sparsity on model performance	107
6.3.4	Impact of assay relatedness on model performance.....	109
6.4	Discussion.....	114
6.5	Conclusion.....	115
Chapter 7 Interpretation of neural networks for QSAR modelling.....		117
7.1	Feature attribution methods	117
7.2	Interpretation of hidden layers.....	121
7.3	Objectives.....	122
Chapter 8 Exploration of chemical features learned in hidden neurons of neural networks		124
8.1	Introduction	124
8.2	Methodology.....	125
8.2.1	Dataset	125
8.2.2	Model training.....	126
8.2.3	Neural network structure	127
8.2.4	Exploration of neuron activations.....	129

8.2.5	Exploration of chemical features learned in hidden neurons.....	129
8.2.6	Network-wide activation analysis for prototypical toxicophore compounds	130
8.3	Results and Discussion	130
8.3.1	Model evaluation	130
8.3.2	Exploration of neuron activations.....	131
8.3.3	Exploration of weights for fingerprint bits.....	136
8.3.4	Exploration of compounds strongly activating neurons	138
8.3.5	Exploration of fingerprint bits with high weight.....	148
8.3.6	Network-wide activation analysis for prototypical toxicophore compounds	155
8.4	Conclusion.....	163
Chapter 9 Automatic extraction of chemical features activating hidden neurons of neural networks		165
.....		165
9.1	Introduction	165
9.2	Methodology.....	165
9.2.1	Inclusion of compounds and fingerprint bits.....	165
9.2.2	Formal Concept Analysis for substructure extraction	166
9.2.3	Summary of the workflow	172
9.3	Illustration of the workflow	173
9.4	Conclusion.....	184
Chapter 10 Evaluation and optimisation of the model explanation approach.....		185
10.1	Introduction	185
10.2	Methodology.....	186
10.2.1	Dataset	186
10.2.2	Model training.....	186
10.2.3	Local evaluation	187
10.2.3.1	Attribution methods	187
10.2.3.2	IG_input.....	187
10.2.3.3	IG_hidden.....	190
10.2.3.4	Evaluation of attributions	191
10.2.3.5	Alert-specific attribution scores.....	192
10.2.3.6	Depiction of atom attributions	192
10.2.4	Global evaluation	193
10.2.5	Modifications to automatic substructure extraction.....	194
10.2.6	Final evaluation on the test set.....	196
10.2.7	Analysis of proportion of attributions accounted for in IG_hidden.....	197
10.2.8	Analysis of predictions for a model based on experimental Ames labels	197

10.3	Results.....	198
10.3.1	Model evaluation	198
10.3.2	Local evaluation of IG_input	199
10.3.3	Global evaluation of extracted substructures.....	205
10.3.4	Local evaluation of IG_hidden	209
10.3.4.1	Analysis of overall and alert-specific performances	209
10.3.4.2	Analysis of individual compounds.....	215
10.3.5	Final evaluation on the test set.....	221
10.3.6	Analysis of the proportion of attributions accounted for in IG_hidden	222
10.3.7	Analysing predictions for a model based on experimental Ames labels	229
10.3.7.1	Analysis of TP compounds that are also positive in Derek	229
10.3.7.2	Analysis of TP compounds that are negative in Derek	230
10.3.7.3	Analysis of TN compounds	232
10.4	Discussion.....	234
10.4.1	Modifications to the substructure extraction workflow.....	235
10.4.2	Comparison of IG_hidden with IG_input	236
10.4.3	Applying IG_hidden to a model trained on experimental Ames labels	237
10.4.4	Depiction of atom attributions	238
10.5	Conclusion.....	240
Chapter 11 Explaining predictions for deep neural networks		241
11.1	Introduction	241
11.2	Methodology.....	241
11.2.1	Model training and evaluation.....	241
11.2.2	Interpreting DNN models by applying IG_hidden to the first hidden layer.....	243
11.2.3	Exploration of neurons in the second hidden layer	243
11.3	Results.....	244
11.3.1	Evaluation of Derek models	244
11.3.2	Interpreting the DNN by applying IG_hidden to the first hidden layer	244
11.3.3	Exploration of neurons in the second hidden layer	246
11.3.4	Evaluation and exploration of additional toxicity and bioactivity datasets.....	254
11.4	Discussion.....	264
11.5	Conclusion.....	265
Chapter 12 Conclusions and future work		266
12.1	Summary	266
12.1.1	Multi-task and imputation modelling for toxicity prediction	266
12.1.2	Using extracted substructures to explain neural network models.....	267

12.2	Limitations and future work	268
12.2.1	Multi-task and imputation modelling for toxicity prediction	268
12.2.2	Using extracted substructures to explain neural network models.....	270
12.3	Final conclusions	273
Appendix	275
Appendix A	275
Appendix B	278
Appendix C	288
Appendix D	289
Bibliography	292

Abbreviations and acronyms

AD: Applicability domain.
ADME: Absorption, distribution, metabolism and excretion.
AhR: Aryl hydrocarbon receptor.
AI: Artificial intelligence.
AR: Androgen receptor.
AR-LBD: Androgen receptor ligand-binding domain.
ARE: Antioxidant response element.
ATAD5: ATPase family AAA domain containing 5.
BCE: Binary cross entropy.
BPFM: Bayesian probabilistic matrix factorisation.
CEBS: chemical effects in biological systems.
CGX: Carcinogenicity Genotoxicity experience.
CNN: Convolutional neural network.
DNN: Deep neural network.
DPAR: Direct Peptide Reactivity Assay.
EBI: European Bioinformatics Institute.
ECFP: extended connectivity fingerprint.
EPA: United States Environmental Protection Agency.
ER: Estrogen receptor.
ER-LBD: Estrogen receptor ligand-binding domain.
EURL-ECVAM: European Union Reference Laboratory for Alternatives to Animal Testing.
FC: Formal Concept.
FCA: Formal Concept Analysis
FN: False negatives.
FP: False positives.
FPR: False positive rate.
GCN: Graph-convolutional neural network.
GHOST: Generalized threshold shifting procedure.
GLP: Good Laboratory Practice.
HSE: Heat shock element.
HTS: High-throughput screening.
ICH: International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use.
IG: Integrated gradients.
InChI: International Chemical Identifier.
ISSTOX: Istituto Superiore di Sanità Toxicity.
k-NN: k-nearest neighbours.
LIME: Local interpretable model-agnostic explanations.
LOAO: Leave-one-assay-out.
LSTM: Long Short-Term Memory.
MCC: Matthews correlation coefficient.
MCMC: Markov chain Monte Carlo.
MI: Mutual information.
ML: Machine learning.
MMP: Mitochondrial membrane potential.
NCBI: National Center for Biotechnology Information.
NICEATM: National Toxicology Program Interagency Center for the Evaluation of Alternative Toxicological Methods.
NOAEL: No Observed Adverse Effect Level.
NR: Nuclear receptor.
NRC: United States National Research Council.
NTP: National Toxicology Program.
OECD: Organisation for Economic Co-operation and Development.
PAH: Polycyclic aromatic hydrocarbon.

PLS: Partial least squares.
PMF: Probabilistic matrix factorisation.
PPAR-gamma: Peroxisome proliferator-activated receptor gamma.
pQSAR: profile-QSAR.
QSAR: Quantitative structure-activity relationship.
REACH: Registration, Evaluation, Authorisation and Restriction of Chemicals.
ReLU: Rectified linear unit.
RF: Random forest.
RMSE: Root Mean Square Error.
RNN: Recurrent neural network.
ROC-AUC: Area under the Receiver Operating Characteristics curve.
SDF: Structure-data file.
SGD: Stochastic gradient descent.
SHAP: Shapley Additive Explanations.
SMILES: Simplified Molecular Input Line Entry System.
SOHN: Self-organising hypothesis network.
SR: Stress response.
SVM: Support Vector Machine.
TN: True negatives.
Tox21: Toxicology in the 21st century.
TP: True positives.
TPR: True positive rate.
TPSA: Topological polar surface area.
XGB: XGBoost (extreme gradient boosting)

List of Figures

Figure 2-1 Risk assessment of chemicals.	5
Figure 2-2 Exemplary mutagenic compound classes.	11
Figure 3-1 K-fold cross-validation.	17
Figure 3-2 ROC curves.	21
Figure 3-3 k-NN principle.	22
Figure 3-4 Example of linear regression.	23
Figure 3-5 Example of SVM.	25
Figure 3-6 Decision tree predicting whether passengers survived the sinking of a ship.	26
Figure 3-7 Random forest principle.	28
Figure 3-8 Gradient tree boosting for regression.	29
Figure 3-9 Architecture of feedforward DNN models.	31
Figure 4-1 Representation of molecules using ECFPs.	39
Figure 5-1 Schematic comparison of single task, multi-task and imputation models.	52
Figure 5-2 Comparison of splitting strategies.	59
Figure 5-3 Schematic depiction of Feature Net models.	63
Figure 5-4 Performance of single task QSAR models.	70
Figure 5-5 Performance of multi-task QSAR models.	72
Figure 5-6 Performance of Feature Net models on the Ames dataset.	74
Figure 5-7 Performance of Feature Net models on the Tox21 dataset.	75
Figure 5-8 Performance of imputation models.	77
Figure 5-9 Performance of imputation models according to test compound chemical similarity: Ames.	80
Figure 5-10 Performance of imputation models according to test compound chemical similarity: Tox21.	82
Figure 5-11 Performance of imputation models according to test compound data label availability.	84
Figure 5-12 Performance of the pairwise FN models.	85
Figure 5-13 Performance of the leave-one-assay-out FN models.	87
Figure 5-14 Concordance between the pairwise and LOAO FN models.	89
Figure 5-15 Effect of assay relatedness on FN models: Ames.	90
Figure 5-16 Effect of assay relatedness on FN models: Tox21.	91
Figure 5-17 Predicted probability scores for the p53 assay.	93
Figure 6-1 Performance of imputation models on the ToxCast dataset.	104
Figure 6-2 Performance of the imputation models using the GHOST approach.	107
Figure 6-3 Performance of models trained on the training set with increased sparsity.	108
Figure 6-4 Correlation between MCC changes and mean MI-entropy ratio.	110
Figure 6-5 Auxiliary assay selection for the assay 'TOX21-Aromatase-Inhibition'.	113
Figure 8-1 Architecture of a DNN.	127
Figure 8-2 Performance of the neural network model.	131
Figure 8-3 Maximum and mean activation of hidden neurons.	132
Figure 8-4 Activations for individual neurons.	133
Figure 8-5 Pairwise correlations of neuron activations.	134
Figure 8-6 Confidence analysis of hidden neurons.	135
Figure 8-7 Weight distributions for individual neurons.	137
Figure 8-8 Analysis of compounds strongly activating neuron 1-43.	139
Figure 8-9 Analysis of compounds strongly activating neuron 1-69.	141

Figure 8-10 Analysis of compounds strongly activating neuron 1-153.....	143
Figure 8-11 Analysis of compounds strongly activating neuron 1-180.....	144
Figure 8-12 Analysis of compounds strongly activating neuron 1-18.....	146
Figure 8-13 Analysis of compounds strongly activating neuron 1-128.....	147
Figure 8-14 Analysis of bits with high weight for neuron 1-43.....	149
Figure 8-15 Analysis of bits with high weight for neuron 1-69.....	150
Figure 8-16 Analysis of bits with high weight for neuron 1-153.....	151
Figure 8-17 Analysis of bits with high weight for neuron 1-180.....	152
Figure 8-18 Analysis of bits with high weight for neuron 1-18.....	153
Figure 8-19 Analysis of bits with high weight for neuron 1-18.....	154
Figure 8-20 Neuron activation by azide compounds.....	161
Figure 8-21 Analysis of compounds strongly activating neuron 1-382.....	162
Figure 8-22 Analysis of bits with high weight for neuron 1-382.....	163
Figure 9-1 Hasse diagram depicting the lattice derived using FCA.....	169
Figure 9-2 Overview of the developed method for extracting fragments responsible for neuron activation.	173
Figure 9-3 Selected compounds per neuron.....	174
Figure 9-4 Selected compounds (A) and fingerprint bits (B) for neuron 1-69.....	175
Figure 9-5 Support and potential for neuron activation of FCs.	176
Figure 9-6 Fingerprint bits forming the intent of the selected FC.....	177
Figure 9-7 Compounds forming the extent of the selected FC.....	178
Figure 9-8 Substructures forming Subnetwork 3 and Subnetwork 7.	181
Figure 9-9 Substructures forming Subnetwork 1.....	182
Figure 9-10 Fragments forming Subnetwork 2.	183
Figure 10-1 Illustration of IG_input.....	189
Figure 10-2 Illustration of IG_hidden.....	191
Figure 10-3 Scheme for estimating the neuron activation of a substructure.	195
Figure 10-4 Attribution AUC scores for IG_input.....	199
Figure 10-5 Atom attributions using IG_input.....	201
Figure 10-6 Alert-specific AUC scores for IG_input.	204
Figure 10-7 Coverage of Derek alerts for original extraction workflow.	206
Figure 10-8 Exemplary substructures related to aromatic nitro (Alert42).	207
Figure 10-9 Exemplary substructures related to nitrogen or sulphur mustard (Alert39).	208
Figure 10-10 Comparison of attribution AUC scores for different attribution methods.....	212
Figure 10-11 Comparisons of compound AUCs and alert AUCs.	213
Figure 10-12 Comparison of atom attributions for individual compounds (Part 1).	216
Figure 10-13 Comparison of atom attributions for individual compounds (Part 2).	218
Figure 10-14 Comparison of atom attributions for individual compounds (Part 3).	220
Figure 10-15 Comparison of atom attributions for individual compounds (Part 4).	221
Figure 10-16 Comparisons of compound AUCs and alert AUCs between IG_input and IG_hidden on the test set.	222
Figure 10-17 Analysis of positive attributions accounted for in model explanations by IG hidden...	223
Figure 10-18 Analysis of positive attributions accounted for in compound 6549.....	225
Figure 10-19 Analysis of positive attributions accounted for in compound 5949.....	226
Figure 10-20 Analysis of positive attributions accounted for in compound 6439.....	227
Figure 10-21 Analysis of positive attributions accounted for in compound 6431.....	228
Figure 10-22 Evaluation of individual compounds and alerts for model trained on experimental Ames labels.....	230

Figure 10-23 Explanations for TP compounds that are negative in Derek.	231
Figure 10-24 Explanations for TN compounds.	233
Figure 11-1 Comparisons of compound AUCs and alert AUCs between IG_input and IG_hidden on the validation set.	245
Figure 11-2 Correlation analysis of hidden neurons.	247
Figure 11-3 Search for 2 nd layer neurons detecting novel chemical features.	248
Figure 11-4 Top-3 compounds for neurons 2_46, 2_89, 2_278.	250
Figure 11-5 ROC_AUC scores for individual neurons.	252
Figure 11-6 Pairwise correlations of selected neurons in Derek model.	253
Figure 11-7 Training compound activations for neuron pairs.	254
Figure 11-8 Comparison of best 1-layer and 2-layer model for Tox21, ToxCast and ChEMBL datasets.	255
Figure 11-9 Analysis of second layer neurons: ATG_ERa_TRANS_up.	257
Figure 11-10 Analysis of second layer neurons: ATG_RXRb_TRANS_up.	258
Figure 11-11 Analysis of second layer neurons: Adenosine A1 receptor.	259
Figure 11-12 Analysis of second layer neurons: Peroxisome proliferator activated receptor alpha.	260
Figure 11-13 Follow-up analyses on 2-layer model trained on ATG_RXRb_TRANS_up.	262
Figure 11-14 Analysis of second layer neurons after 10 training epochs on initial model on ATG_RXRb_TRANS_up.	263

List of Tables

Table 3-1 Confusion matrix of a binary classification task.....	19
Table 3-2 Classification model metrics.	20
Table 4-1 Various chemical representations of acetaminophen (paracetamol).	37
Table 5-1 The Ames dataset.....	56
Table 5-2 The Tox21 dataset.....	57
Table 5-3 Random forest hyperparameters considered in optimisation.	60
Table 5-4 XGBoost hyperparameters considered in optimisation.....	61
Table 5-5 Deep neural network hyperparameters considered in optimisation.	62
Table 5-6 Macau hyperparameters considered in optimisation.	65
Table 5-7 Median MCC scores of single task QSAR models.....	70
Table 5-8 Median MCC scores of multi-task QSAR models.	72
Table 5-9 Median MCC scores of Feature Net QSAR models.	76
Table 5-10 Median MCC scores of imputation models.....	78
Table 5-11 Analysis of the classification threshold.....	94
Table 6-1 Overview of performances for different techniques on the ToxCast dataset.....	105
Table 6-2 Selected auxiliary assays for TOX21-Aromatase-Inhibition.	112
Table 8-1 Parameters used for hyperparameter optimisation.....	126
Table 8-2 Percentiles of distributions for maximum and mean activation of neurons.	132
Table 8-3 Neuron activation by compounds with defined toxicophores.	156
Table 8-4 Pairwise correlations of neurons activated by phenyl azide.	157
Table 8-5 Detailed fingerprint bit weight analysis for neurons activated by phenyl azide.	158
Table 9-1 Binary relations between compounds and fingerprint bits as basis for FCA.....	167
Table 9-2 Extracted substructures for the selected FC.....	179
Table 9-3 Subnetworks of extracted substructures for neuron 1-69.	180
Table 10-1 Model instances used in the chapter.....	187
Table 10-2 Varied parameters for automatic substructure extraction.	196
Table 10-3 Classification metrics for models.	198
Table 10-4 Explanatory performance of best_model and dropout_model.....	205
Table 10-5 Global evaluation for various substructure extraction workflows.	205
Table 10-6 Local evaluation metrics for different model instances and substructure extraction workflows.....	210
Table 10-7 Average alert AUCs.	214
Table 10-8 Overview of atom attributions accounted for by substructure matches for the IG_hidden method.....	215
Table 10-9 Evaluation of explanations on the test set.	222
Table 10-10 Evaluation of explanations extracted from the model trained on experimental Ames labels.	230
Table 10-11 Overview of atom attributions accounted for by substructure matches in IG_hidden method for compounds shown in Figure 10-24.	234
Table 11-1 Parameters used for hyperparameter optimisation (Derek dataset).....	242
Table 11-2 Parameters used for hyperparameter optimisation (Tox21, ToxCast and ChEMBL dataset).	242
Table 11-3 Classification metrics for Derek models.....	244
Table 11-4 Evaluation of explanations for the 2-layer Derek model.....	245
Table 11-5 Most relevant features detected in neurons.	249

Table 11-6 ROC_AUC scores for exemplary neurons.....	253
Table 11-7 Selected assays for further analysis.....	256

Chapter 1 Introduction

Humans are exposed to a multitude of chemicals, for instance, through food ingredients, cosmetics and pharmaceuticals. The companies that produce and market these kinds of products need to ensure that no unacceptable risks arise from the chemicals contained in these products. To that end, the companies are required to conduct various toxicity tests. The legal requirements for tests vary between different industries and geographical regions. Historically, animal (in vivo) tests have been the most important approach to test chemicals for toxicity and these tests are still mandatory in many regions. However, animal tests are associated with a number of drawbacks. Importantly, there are considerable differences in the biology of humans and test species (e.g. rats, rabbits), so that animal studies may fail to detect toxic effects relevant to humans. Moreover, animal studies are undesirable from an ethical perspective and their acceptance in society is low. In addition, animal tests are very expensive and time-consuming. For all these reasons alternative methods to test toxicity are required. Available alternatives include in vitro tests (in simple test systems like cell cultures) and computational (in silico) tests.

The most common computational approaches are QSAR (Quantitative structure-activity relationship) models, which predict the toxicity of chemicals based on their structure by using the principle that similar chemicals tend to have similar properties (Maggiore et al., 2014). While QSAR models are widely used for screening purposes (e.g. selection of molecules for experimental testing as potential drug candidates), their application to predict the absence of toxicity in an approval process is limited to few examples (e.g. mutagenicity assessment for impurities in pharmaceuticals) (European Medicines Agency, 2018). A reason for this is the limited performance of QSAR models to predict complex in vivo toxicity endpoints (e.g. liver toxicity), which may be caused by a variety of different biological effects (Cherkasov et al., 2014).

Strategies to improve model accuracy include using sophisticated deep learning algorithms as well as multi-task and imputation modelling approaches. A drawback of these approaches is the difficulty of interpreting the predictions made. According to OECD (Organisation for Economic Co-Operation and Development) rules, which are followed by many regulatory agencies, interpretability of QSAR models should be considered (if possible) if the models are to be used in a regulatory context. Deep neural networks (DNNs), which have become very popular for QSAR, are especially difficult to interpret due to their complex structures, which is why they are often referred to as black box models. The aim of

Chapter 1: Introduction

the work in this thesis is to address two current limitations in toxicity prediction: the limited accuracy of prediction methods; and the lack of interpretability of prediction models.

Chapter 2 provides an introduction to toxicity assessment of chemicals. After a general introduction, the toxicity endpoint mutagenicity is described because mutagenicity data is used for QSAR modelling in various experiments throughout the thesis.

Chapter 3 introduces general principles of machine learning (ML) techniques, as these form the basis of QSAR modelling. This includes a description of a typical ML workflow with a focus on model evaluation as well as an overview of relevant ML algorithms.

Chapter 4 describes the use of QSAR models for toxicity prediction. It begins with a description of the key concepts in chemoinformatics and QSAR modelling: chemical representation and molecular similarity. The chapter also provides an overview of factors determining the success of QSAR models, summarises the current use of QSAR models in regulatory toxicity, and describes how neural networks have been used for toxicity prediction.

Chapters 5 and 6 investigate the use of multi-task and imputation models for toxicity prediction. In Chapter 5, traditional multi-task models (no experimental toxicity data for test compounds available) and multi-task imputation models are developed for two multi-target in vitro toxicity datasets (Ames mutagenicity and Tox21 data) and compared to single task QSAR models. Furthermore, the impact of chemical similarity, sparsity and assay relatedness on the performance of the imputation models is investigated. Chapter 6 extends the methods from Chapter 5 to an in vitro toxicity dataset of larger scale (ToxCast; several hundred assays) to see if the findings are generalisable.

Chapter 7 provides an overview of approaches used to interpret QSAR models based on neural networks. Chapters 8 to 11 describe the work carried out to develop and test a novel strategy to interpret neural networks by leveraging chemical features learned in hidden layers of neural networks. Due to their reduced complexity, initial experiments (Chapter 8 to 10) were conducted on neural networks consisting of a single hidden layer. The rationale behind this was to use relatively simple models as a proof of concept before moving on to more complex architectures.

Chapter 8 investigates which chemical features are learned in hidden layers of neural networks. This includes an analysis of the compounds that most strongly activate a neuron as well as the input features that are assigned high weights with respect to a given hidden neuron. A neural network trained to predict Ames mutagenicity was used for this analysis.

In Chapter 9, the development of a method to automatically extract chemical substructures responsible for the activation of hidden neurons is described. The method uses both information

Chapter 1: Introduction

about compounds strongly activating a neuron and learned network weights. Formal Concept Analysis (FCA) is used to systematically combine the information sources.

In Chapter 10, the usefulness of automatically extracted substructures to explain predictions made by neural networks is evaluated. The substructures are combined with a measure of how important a neuron is for a given prediction using integrated gradients (IG). The developed method (called IG_hidden) is compared to an established way of interpreting neural networks (IG applied to input features which is referred to as IG_input). Lhasa's Derek alerts are used as a ground truth to measure the quality of model explanations.

In Chapter 11, the suitability of the developed method for interpreting DNNs (i.e. neural networks with more than one hidden layer) is investigated.

Finally, Chapter 12 summarises the findings of the different experimental chapters, points out limitations in the conducted studies and provides suggestions for future steps to build on the results presented in this thesis.

Chapter 2 Toxicity assessment of chemicals

2.1 Introduction to Toxicology

Toxicology is defined as the scientific discipline that studies adverse effects of chemicals on living organisms (Klaassen, 2008). Due to the complexity of biological processes occurring in organisms, a vast number of adverse effects caused by chemicals may potentially occur. These may include local effects at the site of chemical contact (e.g. skin irritation caused by acids) or systemic effects after a chemical has entered the body. Systemic toxic effects generally are the result of a chemical interacting with one or more target biomolecules. Biomolecules of toxicological relevance include proteins (involved in signal transduction, metabolism, biosynthesis), lipids (forming cell membranes) and nucleic acids (genetic information). The interaction of a chemical with biological targets and the resulting effects are described by a chemical's toxicodynamics. Another key factor for the occurrence of toxic effects are a chemical's toxicokinetics. This term describes the fate of a chemical within the body including absorption, distribution, metabolism and excretion, often summarised as ADME properties of a chemical. ADME properties determine (i) if and to what extent a chemical enters the body, (ii) which compartments of the body are reached by the chemical, (iii) if the chemical is metabolised (which may decrease or increase its toxicity), (iv) how quickly a chemical is excreted to terminate its effect.

Chemicals of toxicological relevance are any that humans or other organisms may be exposed to. This includes both deliberate exposure (e.g. pharmaceuticals, food additives, cosmetics) and unintentional exposure (e.g. natural or synthetic food contaminants, industrial chemicals at a work site). To prevent adverse effects caused by chemicals, their toxicity needs to be determined. A fundamental rule of toxicology states that the toxicity of a chemical is determined by the dose an organism is exposed to. This relationship was first described in the 16th century by Paracelsus in his famous quote: "All things are poison and nothing is without poison, the dosage alone makes it so a thing is no poison" (Paracelsus, 1965). This statement is supported by the finding that even water, a chemical generally considered as harmless and essential for life, may cause death in unnaturally high doses (Gardner, 2002). Hence, it is impossible to universally distinguish toxic from non-toxic chemicals. Instead, it needs to be determined whether the exposure to a certain chemical is associated with unacceptable risks. The process of chemical risk assessment is described in the following section.

2.2 Chemical risk assessment

2.2.1 Traditional risk assessment

To assess the risk of a chemical, one has to consider three distinct aspects namely hazard, the dose-response relationship and exposure as stated in the so-called risk assessment paradigm (Omenn, 1995). This widely recognised framework will be described briefly in the following paragraphs according to the NRC (United States National Research Council) and is summarised visually in Figure 2-1 (NRC, 1983, 1994).

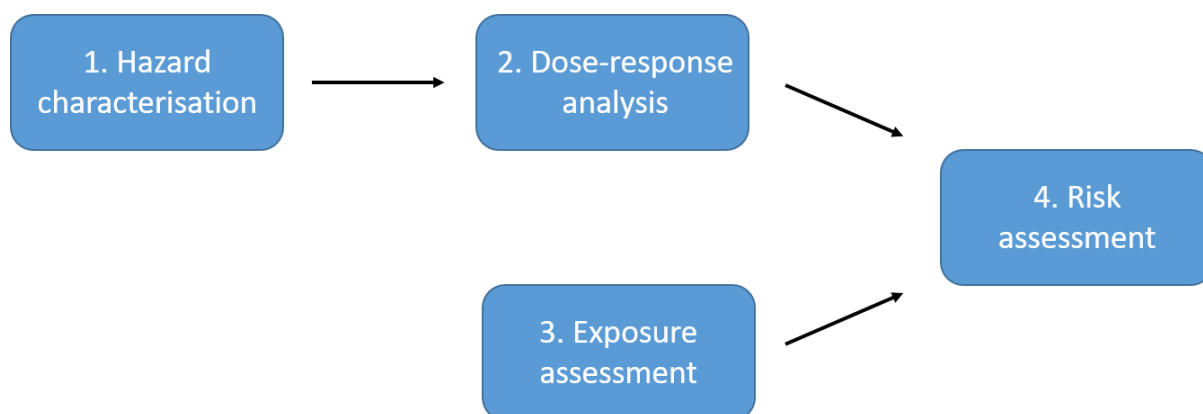


Figure 2-1 Risk assessment of chemicals.

Hazard qualitatively describes potential adverse outcomes (e.g. skin sensitisation, liver fibrosis, bladder cancer) caused by a chemical and is an inherent property of a substance. A dose-response relationship describes what dose of a chemical is necessary to cause a certain effect in a human. However, it must be stated that the susceptibility of a human toward a chemical (in addition to the dose) is determined by many factors including sex, age, genetics, medical conditions or presence of other chemicals. Therefore, the relationship between dose and response may vary substantially among different individuals and this needs to be taken into account in the risk assessment. Typically, hazard and dose-response relationships are studied in animal experiments, resulting in a qualitative description of potential hazards and a NOAEL (No Observed Adverse Effect Level), which describes the highest tested dose that did not cause any adverse effect. To account for uncertainties arising from different susceptibilities among different individuals (see above) and from potential biological differences between the experimental animal species and humans, the NOAEL is divided by so-called

uncertainty factors to obtain a risk value, which describes the dose considered to be safe for the human population.

Exposure means the amount of a chemical an organism comes in contact with, which can happen via different routes (mainly orally, dermally or via inhalation). Exposure to harmful chemicals may result in local effects (i.e. at the site of exposure; e.g. skin irritation) or in systemic effects if the chemical passes the barriers of the human body (skin, lung, stomach/intestine). Systemic effects may occur in any part of the organism according to the distribution of the chemical within the body. In the simplest case, the amount an individual is exposed to may be known (e.g. dose of a drug), whereas, in other cases, the exposure needs to be estimated using analytical measurements and models. For instance, to determine the exposure of a worker to a gas present at a working site, the exposure can be estimated with calculations taking into account the concentration of the gas at the site, the time of exposure and the volume of air inhaled per time interval by the worker.

A risk assessment involves first evaluating whether a chemical possesses a concerning hazard profile (hazard characterisation). If this is the case, in the next step the known or estimated exposure to the chemical is compared to the risk value obtained from hazard characterisation and dose-response analysis. If the exposure to the chemical exceeds the risk value, the outcome of the assessment is that a risk of toxicological effects exists. However, it must be stated that this is a semi-quantitative approach that does not allow the calculation of the probability that a given individual will develop symptoms in a particular exposure scenario. On the other hand, if there is no or negligible exposure, there will be no risk, even though a chemical may be hazardous.

The range of toxicity studies that need to be conducted is legally defined and varies between different industries and countries. A common principle is that animal studies have to be conducted in a standardised manner according to testing guidelines which are issued by various organisations such as the OECD (Organisation for Economic Co-operation and Development), the ICH (International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use) and the EPA (United States Environmental Protection Agency). Parameters usually defined by testing guidelines include selection of animal species, number of animals, housing conditions, substance dosing, study duration, parameters to be measured and form of reporting the obtained data. Most of the studies are conducted in rodents (rats, mice), but sometimes a study in a non-rodent (e.g. rabbit, dog) is also required. A framework aiming to further enhance quality and reproducibility of toxicological studies is GLP (Good Laboratory Practice) (Weinberg, 2003). In essence, GLP defines how toxicological studies must be run in terms of planning, standardising procedures, monitoring, data recording, storing and reporting. The most important types of toxicological studies are briefly

introduced below. Different testable types of toxicity are often referred to as toxicological endpoints. Generally, the studies aim to identify hazards of the tested chemicals and to determine a dose-response relationship for identified hazards.

Acute studies: These evaluate whether a chemical induces mortality after single administration of the chemical. Application routes may be oral (OECD, 2008), dermal (OECD, 2017) and inhalation (OECD, 2018c), depending on the chemical's anticipated exposure routes.

Repeated-dose studies: In contrast to acute studies, the chemical is administered repeatedly, usually once a day. Common durations of studies are 28 days (subacute) (OECD, 2018a), 90 days (subchronic) (OECD, 2018b) and 12 months (chronic) (OECD, 2018d). Similar to acute studies, different application routes (oral, dermal, inhalation) can be studied. Parameters that are measured in repeated-dose studies include morbidity and mortality, weight and food consumption of the animals, biochemical and haematological measurements, gross necropsy (determination of organ weights and examination of the whole organs) and histopathology (microscopic tissue examination). The aim of these studies is to detect effects on organ systems that only occur after long-term exposure to chemicals.

Genotoxicity studies: Genotoxicity describes the property of a chemical to damage the genetic information of cells, which may lead to mutations and ultimately may cause cancer. There are several assays (i.e. test systems) to test genotoxicity in both in vivo (animals) and in vitro (cultivated cells) systems. To assess the genotoxic potential of a chemical usually a set of different in vitro and in vivo tests, which cover different potential genotoxic mechanisms, needs to be conducted (Müller et al., 1999). The Ames Test for mutagenicity is a commonly used in vitro test conducted in bacteria. Since Ames Test data was used within this thesis to build predictive models, its principle will be described in more detail in a separate Mutagenicity section below (2.3).

Carcinogenicity studies (OECD, 2011): Carcinogenicity studies test the capability of chemicals to cause cancer. The focus of the studies is to detect neoplasms (i.e. abnormal tissue growth), which may be benign or malignant (i.e. cancer). Carcinogenic chemicals often are genotoxic, yet there are other mechanisms by which a chemical may induce cancer, such as by increasing tissue proliferation and thus promoting spontaneous mutations. The design of carcinogenicity studies resembles that of chronic toxicity studies. However, the duration of these studies is typically two years, which exceeds chronic toxicity studies.

Reproductive and developmental studies: These studies evaluate adverse effects on sexual function, fertility and development of both male and female animals. In order to do so, animals

across two generations of offspring are treated with the chemical (OECD, 2001). Parameters of the studies include mating behaviour, histopathology of sexual organs, malformation in foeti and developmental abnormalities in the offspring.

2.2.2 Alternative methods for risk assessment

Toxicity studies and risk assessment procedures have in essence been the same for many decades, despite having clear weaknesses and limitations. On the one hand, the risk assessment relies on very large numbers of animals, which is questionable from an ethical perspective. As early as 1959, Russell and Burch stated their 3R principles (replacement, reduction and refinement), which aim to improve animal welfare in animal studies. The principles state that wherever possible, (i) animal studies should be replaced with studies on insentient material, (ii) the number of animals should be reduced as much as possible to still obtain sufficient information and (iii) the studies should be refined in such a way that the occurrence of inhumane procedures is minimised (Tannenbaum & Bennett, 2015). Moreover, due to significant biological differences between humans and experimental species, the procedures always bear the risk of eliminating harmless substances (toxic only in experimental species) or worse, not detecting hazardous substances (toxic only in humans).

These shortcomings were identified by the NRC in an influential report leading to the inception of a federal inter-agency collaboration named “Toxicology in the 21st century” (Tox21) (NRC, 2007). The aim is “to move toxicology from a predominantly observational science at the level of disease-specific models to a predominantly predictive science focused upon a broad inclusion of target-specific, mechanism-based, biological observations.” (National Toxicology Program, 2004). In other words, instead of merely observing toxicity in animals, the concept envisions to obtain a broad mechanistic understanding of biological perturbations triggered by chemicals. An integral part of the programme was a massive HTS (high-throughput screening) project, in which approximately 10,000 chemical substances were tested in approximately 70 in vitro assays covering targets and pathways of presumed toxicological relevance. The aim was to identify in vitro test systems that are predictive for adverse in vivo outcomes.

While in the United States federal agencies have been the main driver for innovations in toxicity testing, in the European Union recent legislation, namely the new legal framework for regulation of industrial chemicals REACH (Registration, Evaluation, Authorisation and Restriction of Chemicals) (EU, 2006) and the ban on the use of animal studies for evaluating the safety of cosmetics (EU, 2009), has pushed companies to use alternative ways to evaluate safety. REACH requires companies to use

existing data and knowledge to evaluate safety. This includes existing *in vivo* data, *in vitro* data, and *in silico* methods (i.e. computational models). Conducting new *in vivo* animal studies is considered only as last resort when available data is not sufficient to evaluate the safety.

Computational methods for safety evaluation aim to predict the toxicity of molecules and are based on the principle that structurally similar chemicals tend to have similar properties. This paragraph will introduce briefly three different approaches, namely read-across, expert systems and QSAR models.

The assumption behind the read-across method is that unknown toxicological activities can be inferred within a category of chemicals without additional testing. To that end, a chemical without data for a particular toxicological endpoint (target chemical) is grouped with similar chemicals for which data are available (Cronin, 2013). Assessment of similarity in this context is based on common structural features that are knowingly associated with the respective toxicity. Read-across is a widely used approach within the REACH regulation. Another *in silico* method is the so-called expert system. These systems condense toxicological knowledge into abstract rules which may concern the presence of certain structural features that are related to a type of toxicity (structural alerts) or the role of certain physicochemical properties for a certain type of toxicity. Derek Nexus is an expert system developed by Lhasa Limited which is widely used across different industries to predict various toxicity endpoints (including mutagenicity, carcinogenicity and reproductive toxicity) (Marchant et al., 2008). Predictions made by the Derek software are a likelihood term (e.g. 'certain', 'probable', 'plausible') and are supported by the evidence used to obtain the result (i.e. literature references, exemplary compounds for alerts). The third method is QSAR (Quantitative structure-activity relationship) modelling, which aims to describe a certain property or activity of a molecule as a function of its chemical structure. QSAR modelling is a well-established method with wide applications beyond toxicity predictions. Currently, both expert systems and QSAR models are mainly used for screening purposes (Greene & Naven, 2009; Roncaglioni et al., 2013). Screening in this context means that, for instance, potential drug candidates are assessed computationally for potential toxicological liabilities in order to find substances that are less likely to fail in preclinical and clinical studies.

2.3 Mutagenicity

2.3.1 Introduction

Mutagenicity describes the potential of a chemical to modify an organism's genetic information. Different QSAR models to predict mutagenicity of chemicals will be used in different studies

throughout this thesis. Therefore, a detailed introduction to this toxicity endpoint will be instrumental to understand the studies. This section will firstly explain molecular foundations of mutagenicity as well as present chemical groups known to be mutagenic. Then, the Ames test will be introduced as a widely applied in vitro test for mutagenicity.

The DNA encodes genetic information as a sequence of four different base compounds (adenine, thymine, guanine, cytosine) stored as a macromolecule. Mutagenicity is often the result of a compound chemically modifying the DNA (Dipple, 1995). The DNA bases possess nucleophilic nitrogen and oxygen atoms which may react with electrophilic compounds to form adducts. Examples of electrophiles are shown below when presenting various chemical classes with mutagenic properties. Some chemicals may react at the same time with bases of both DNA strands to form cross-links (Dronkert & Kanaar, 2001). Another modification of DNA bases may occur due to a hydroxylation by chemicals (Poulsen et al., 1998). Some chemicals (typically polycyclic compounds with planar structure) may modify the DNA structure by intercalating between the strands of the DNA double helix structure (Ferguson & Denny, 2007). All these primary chemical modifications of the DNA may result in different types of DNA damage due to downstream processes such as replication or attempted DNA repair. Commonly occurring DNA damages include:

- Substitutions: change of a DNA base. These may result in a wrong amino acid being built into a protein (Freese, 1959).
- Deletions: removal of bases from the DNA. Since a sequence of three DNA bases encode a certain amino acid in a synthesised protein, deletions lead to a frameshift during protein synthesis (Ripley, 1990).
- Double-strand break: If not repaired, these may lead to cell death via apoptosis or create mutations on the level of chromosomes. (Jackson, 2002)

Various mutations may accumulate in cells which ultimately may lead to the development of tumour cells.

2.3.2 Mutagenic chemicals

In this section, several classes of mutagenic chemicals are briefly introduced. In many cases, metabolic activation is essential to form electrophilic species which ultimately lead to DNA adducts. Exemplary compound structures along with the mutagenic forms are shown in Figure 2-2.

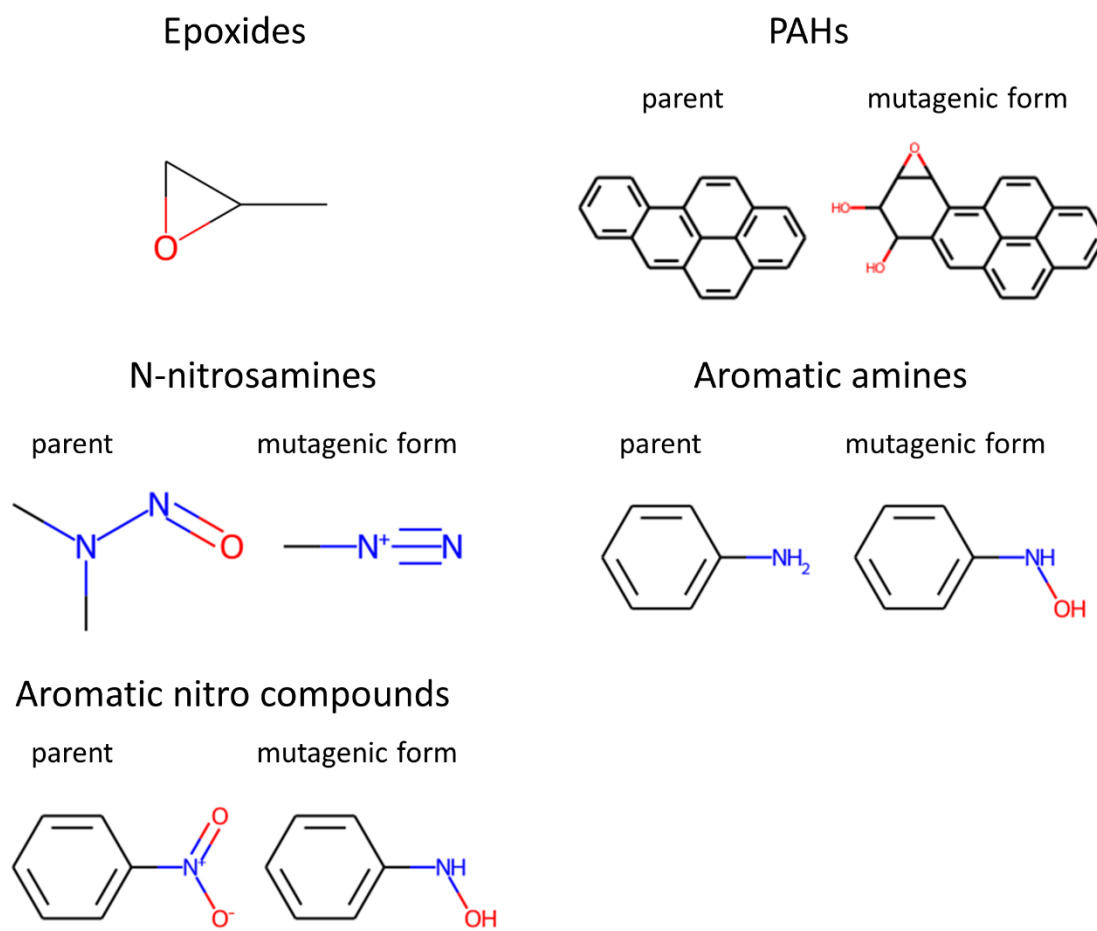


Figure 2-2 Exemplary mutagenic compound classes. Shown are parent compound as well as the transformed species responsible for DNA adduct formation.

Epoxides: Epoxides, which are widely used industry chemicals (Manson, 1980), are cyclic ethers with a three-membered ring. Due to their strained structure, the carbon ring atoms are highly electrophilic enabling chemical a chemical reaction (ring opening) with nucleophilic DNA bases (Wade et al., 1978).

Polycyclic aromatic hydrocarbons (PAHs): PAHs, naturally occurring in coal and formed in combustion of organic materials (Achten & Hofmann, 2009), consist of fused aromatic hydrocarbon rings. Especially Benzo[a]pyrene has been extensively studied for its mutagenic properties. The mutagenicity of PAHs is mediated through metabolites formed via cytochrome P450 enzymes. In case of Benzo[a]pyrene, a covalent bond to DNA bases is formed via an electrophilic epoxide metabolite (Baird et al., 2005). Intercalation between the DNA double strand (possible due to their planar structure) has been described as prerequisite for formation of covalent DNA adducts (Harvey & Geacintov, 1988).

N-nitrosamines: N-Nitrosamines, relevant for instance due to their occurrence as impurities in pharmaceuticals (Tuesuwan & Vongsutilers, 2021), are another example of compounds' mutagenicity

arising from metabolic activation by cytochrome P450 enzymes. Following initial hydroxylation of an α -carbon atom, highly reactive alkyl diazonium cations or carbo cations may be formed in a multi-stage mechanism (Cross & Ponting, 2021).

Aromatic amine and nitro compounds: Aromatic amines and nitro compounds are important classes for the synthesis of many chemicals including pharmaceuticals, pesticides and dyes (Ju & Parales, 2010). For both classes of compounds mutagenicity is mediated through N-hydroxyarylamine metabolites formed via oxidative or reductive reactions, respectively (Benigni et al., 2000).

2.3.3 Ames test for mutagenicity

A popular method to assess chemical mutagenicity is the Ames test, named after its developer Bruce Ames (Ames et al., 1975) and adapted by the OECD as a guideline study (OECD, 1997). The assay tests for the potential of a chemical to induce point mutations, which may be substitutions, additions or deletions of DNA base pairs. The assays are based on bacteria strains that have been artificially mutated in such a way that they lack the ability to produce essential amino acids. Mutagenic chemicals can cause mutations that reverse the artificially introduced mutation so that the bacteria regain the ability to synthesise the amino acid. Only such re-mutated cells will be able to grow in a medium lacking the essential amino acid, which enables the detection of mutagenic chemicals. It is often the case that chemicals in their original form do not react with DNA but are metabolised by animals or humans to reactive molecules. Therefore, it is common to mimic the metabolism of higher organisms in *in vitro* systems by adding appropriate enzymes, typically in form of a liver homogenate (S9 fraction). The Ames test can be conducted in a range of different bacteria strains which were designed to detect different types of DNA mutagenic mechanisms (Hamel et al., 2016). A compound needs to be tested in different bacteria strains in order to cover all potential mechanisms (Williams et al., 2019).

2.4 Conclusion

The present chapter introduced how chemicals are assessed for toxicity. Due to the relevance of mutagenicity in various studies within this thesis, special emphasis was put on introducing this toxicity endpoint. The overarching theme of the PhD project was investigating QSAR models predicting toxicity of chemicals, which may be used to replace *in vivo* and *in vitro* studies for toxicity. In order to understand the principles behind those models, the following chapters will introduce machine

Chapter 2: Toxicity assessment of chemicals

learning (ML) in general (Chapter 2) and QSAR models in specific including basic principles on computational representation of chemicals (Chapter 3).

Chapter 3 Introduction to machine learning

Machine learning (ML) is defined as a computer solving a task without being explicitly instructed how to do it (Samuel, 1967). In contrast to following an algorithmic step-by-step procedure, ML methods seek to develop a solution by analysing patterns in the provided data. ML is part of the wider term artificial intelligence (AI), which can be defined as a computer behaving in a way that humans consider as intelligent (i.e., human-like or rational) (Kok et al., 2002). The relevance of ML in society has increased considerably in recent years and current applications of ML include, amongst many others, image recognition (Pak & Kim, 2017), machine translation (Y. Wu et al., 2016), recommendation systems (Gomez-Uribe & Hunt, 2015) and self-driving cars (Bojarski et al., 2016). Relevant to this work, ML methods can be applied to build QSAR models, which were already mentioned in Chapter 2 and will be introduced in more detail in Chapter 4. This chapter introduces common ML algorithms and describes the construction and evaluation of ML models.

3.1 Tasks in machine learning

Depending on the nature the learning process, ML can be divided in supervised and unsupervised learning (Géron, 2019). In either case, learning (or fitting) of a model occurs using a training data set containing a number of individual data instances. In supervised learning, a data instance is provided to the algorithm with d feature values (x : a d -dimensional vector) and a label (y : a numerical value or class label). The label represents the ground truth for a given data instance and the feature values are properties of the instance. The aim is to fit a model that predicts the label for a data instance using the provided feature data. Formally, a fit model can be described as a function mapping a d -dimensional vector in feature space to a prediction value. Supervised learning can be used to either predict numerical values (regression models) or categorical values (classification models). A classification model may discriminate between two classes (binary classification) or more than two classes (multi-class classification). In the simplest case, a ML model is trained to predict one label per data instance (single task model). In contrast, multi-task ML models are trained simultaneously predict several tasks. Multi-task modelling is based on the rationale that transfer of knowledge between related tasks may increase predictive performance (Caruana, 1997).

In unsupervised learning, no labels are given to the learning algorithm. Instead of predicting labels, the aim is to predict patterns in the dataset, merely by considering the features of data instances. For

example, clustering algorithms attempt to find groups of similar data points (i.e. clusters), whereas, dimensionality reduction techniques aim to find a data representation for the instances of lower dimensionality than the number of input features. The following descriptions in this chapter will focus on supervised learning, as these are the techniques suitable for QSAR modelling. In the following sections of this chapter, the stages of a typical ML workflow will be described, and finally relevant ML algorithms will be introduced.

3.2 The machine learning workflow

A ML project normally consists of the steps: data collection, data curation, model training, model evaluation and selection, model deployment.

3.2.1 Data collection and curation

The data collection step depends on the problem that is to be solved. In general, data of both sufficient quantity and quality need to be retrieved. Specific requirements on the data depend on the domain and the respective data types used in the field. To obtain a large enough dataset, data from different sources may need to be combined. This may come at the cost of data inconsistency between the sources. For instance, experimental toxicity data from different sources may have been obtained using slightly different protocols. Typical sources for toxicity data used to train QSAR models will be introduced in Chapter 4.

Before training a ML model, the retrieved data needs to be curated. This may include handling of incomplete data, transformations of feature or target values and feature selection. Chemical data as used for QSAR modelling requires specific curation steps. These will be described in a Chapter 4.

Handling of incomplete data: Retrieved datasets are often sparse in the sense that not all features are known for every data point. However, most ML algorithms require a complete dataset as input. In order to obtain a complete dataset, incomplete data points may be either discarded or the gaps may be filled using reasonable estimates. Estimates may be achieved using mean values of the dataset for a particular feature or a more sophisticated predictive model (Batista & Monard, 2003). A popular set of modelling techniques for filling data gaps are those based on matrix factorisation. A brief introduction to these methods will be given in a later section of this chapter, next to other ML algorithms.

Feature transformations: Input features typically need to be in a numeric form. Categorical data represented as string data may be transformed in different ways (Brownlee, 2020). One-hot encoding means that a separate feature is generated for each category of the original feature. A one indicates that an instance belongs to a particular category and a zero indicates that the instance does not belong to the category. For ordinal data (i.e. categories with a defined order), ordinal encoding may be selected where the information about the order is preserved, for instance by assigning ordered integer values to the different categories. Numerical features may comprise a wide range of different instance properties and hence be on different scales. Many ML methods preferentially detect patterns in numerical features of larger magnitude. To eliminate this bias, features need to be scaled to the same magnitude. Common methods are Z-score normalisation (the mean of the data points is scaled to 0 and the variance to 1) and min-max normalisation (the minimum and maximum values are scaled to a defined range, e.g. [-1,1]) (Shalabi et al., 2006).

Feature selection: A large number of features may be detrimental for a ML algorithm when there are few observations, because the feature space is sparsely populated, which makes it difficult for the algorithm to find meaningful patterns in the data (this is often referred to as the curse of dimensionality) (Friedman, 1997). Therefore, it may be necessary to reduce the number of features used as input to a ML model (Guyon & Elisseeff, 2003). Commonly, features with low variance are removed, as they are not useful to discriminate examples. Also, highly correlated features should be removed to discard redundant information. A further reduction of features can be obtained empirically by evaluating the performance of different combinations of features. For instance, recursive feature elimination is an iterative approach, where the least important feature is removed after each model training cycle (Gregorutti et al., 2017).

3.2.2 Model training and evaluation

The following paragraph will introduce common strategies for training and evaluation of ML models. Metrics used to evaluate classification and regression models will be presented in a separate section. Prevention of overfitting and handling of imbalanced datasets in classification tasks are specific issues related to training and evaluation of ML models. These will be briefly introduced later in this section.

For training and evaluating ML models the original data set is usually separated in three non-overlapping sets of data instances: a training set, a validation set and a test set (Géron, 2019). As the name suggests, the training set is used to train a number of ML model instances using different algorithms and hyperparameters. Each of the model instances is evaluated on the validation set in

order to find the best one. The performance of the selected model is finally evaluated on the test set. It is essential that the test set was not used to inform any stages of data curation, model training and model selection to ensure the measured performance on it is an unbiased estimation of model generalisation to new data instances. A technique enabling more insights on model performance during model validation is cross-validation. In k-fold cross-validation (shown in Figure 3-1), the dataset (after putting a test set aside) is separated into k folds of equal size. In an iterative manner each of the k folds is used once as validation set (orange boxes in Figure 3-1) for a model trained on all remaining folds. The performance may be averaged across the different iterations to get a more precise estimation of model performance.

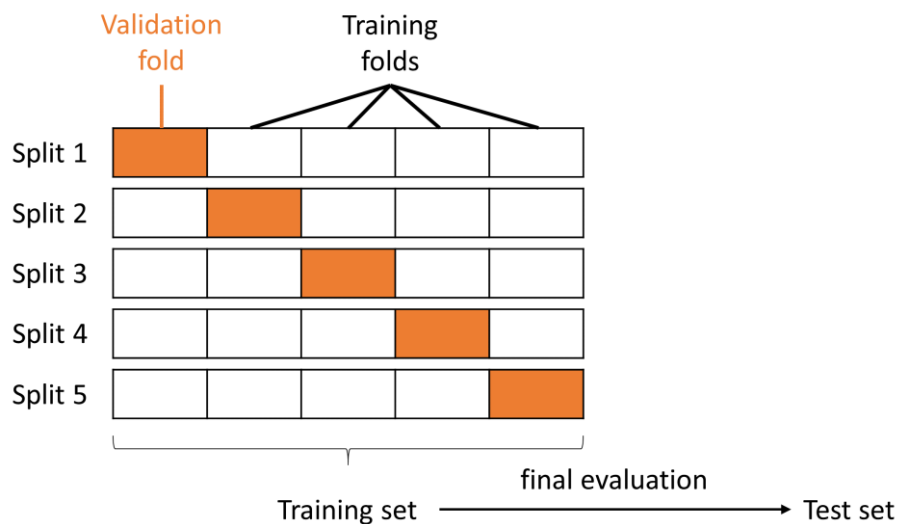


Figure 3-1 K-fold cross-validation. The training set is divided k (here k=5) times into 1 validation fold and k-1 training folds. In each iteration, a model instance is trained on the training folds and evaluated on the validation fold. This way, each data instance of the training set is used once in an evaluation fold. This enables a more robust validation of models with different parameters. Once the best set of parameters has been identified, the model is retrained on the full training set for final evaluation on the test set.

Various different ML algorithms with different benefits and limitations exist (see section 3.4). Complex models with many tuneable parameters may provide a benefit in performance over conceptually simpler modelling techniques, yet complex models may be more difficult to understand, and typically require more data for training. It is generally not known a priori which algorithm will give the best model on a given dataset and hence different ones are usually tested in a ML project. Moreover, most of the ML algorithms require hyperparameters to be set which determine specific aspects of the learning process. Different settings for hyperparameters need to be tested to optimise model performance. A popular choice for hyperparameter optimisation is grid search, in which all combinations of selected values for the different hyperparameters are evaluated on the validation set (or in a cross-validation) to find suitable hyperparameters (Bergstra et al., 2011).

A common challenge when training a ML model lies in the concept of bias-variance trade-off (Fortmann-Roe, 2012). High bias is where the model does not well reflect trends in the training data, for example, due to the model being overly simple. This is referred to as underfitting. In contrast, high variance means that the model reacts very sensitively to small variations in the training data, as it models noise in the data rather than general trends (i.e. overfitting). Prevention of overfitting is a common challenge that ML practitioners face. Overfitting is characterised by the poor capability of a model to generalise on unseen data, while showing minimal error on training data. Strategies to prevent overfitting will be introduced next to the ML algorithms they are applied to. These may be, for instance, to restrict the complexity of models (e.g. limited depth in tree-based models) or to limit the cycles of iterative model fitting (e.g. early stopping for neural network models).

A particular challenge for learning classification models may exist if the proportion of classes in the data set widely differs (i.e. class imbalance). When not accounting for class imbalance, ML models usually tend to make more accurate prediction for the majority class. Notably, caution is required when evaluating model performance, as some metrics may be dominated by the performance on the majority class hence obscuring poor model performance on the minority class (see section 3.3). Approaches to account for class imbalance include (I) resampling of training data to achieve a better balance (oversampling of the minority class or undersampling of the majority class) (Estabrooks et al., 2004), (II) increasing the importance of data instances from the minority class during training with a weighting scheme (Elkan, 2001) and (III) changing the classification threshold applied to the raw predicted probabilities for data instances (Sheng & Ling, 2006).

3.2.3 Model deployment

Once a final model has been selected, it can be deployed to be used in the real world. A ML project is typically not finished once the model has been deployed. The performance of the model must be monitored over time. For instance, changes in real world data may require changes to be made to the model (Sculley et al., 2015). In situations where new data is collected over time, ML models may be retrained to increase performance, for instance when more experimental data becomes available in pharmaceutical companies (Göller et al., 2020).

3.3 Evaluation metrics

3.3.1 Regression models

In regression tasks, the ML model aims to estimate the true numerical values of the data as accurately as possible and various metrics may be used to evaluate model predictions (Botchkarev, 2019). One way to measure the performance of a regression model is to compute the Root Mean Square Error (RMSE) according to the formula:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{i,true} - y_{i,pred.})^2}$$

where $y_{i,true}$ is the true value of the i^{th} data point in the test set of size n , $y_{i,pred.}$ is the corresponding predicted value. Another popular metric is the coefficient of determination (R^2) given by the equation:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_{i,true} - y_{i,pred.})^2}{\sum_{i=1}^n (y_{i,true} - \bar{y})^2}$$

where \bar{y} is the mean of the true values. Generally, R^2 is a measure of the correlation between true and predicted values and can be interpreted as the proportion of variance in the data accounted for by the model.

3.3.2 Classification models

As opposed to regression tasks, in a classification task the prediction for a given data point is either correct or incorrect. The result of a classification task can be represented by a confusion matrix, as shown in Table 3-1, which has the following form for binary classification tasks (positive vs. negative):

Table 3-1 Confusion matrix of a binary classification task.

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives (TP)	False Negatives (FN)
Actual Negatives	False Positives (FP)	True Negatives (TN)

Many classification metrics can be derived directly from the confusion matrix (Hossin & Sulaiman, 2015) and these are reported in Table 3-2.

Table 3-2 Classification model metrics.

Metric	Formula	Description
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	proportion of correctly predicted data points
Sensitivity (recall)	$\frac{TP}{TP + FN}$	proportion of actual positives that are predicted correctly
Specificity	$\frac{TN}{TN + FP}$	proportion of actual negatives that are predicted correctly
Precision	$\frac{TP}{TP + FP}$	proportion of predicted positives that are actual positives
Balanced accuracy	$\frac{\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)}{2}$	mean sensitivity across both classes
F1 score	$2 \times \frac{\textit{precision} \times \textit{sensitivity}}{\textit{precision} + \textit{sensitivity}}$	balanced form of sensitivity (recall) and precision
Matthews correlation coefficient (MCC) (Matthews, 1975)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$	complex metric accounting for all four classes of the confusion matrix

Another commonly used metric is the area under the Receiver Operating Characteristic curve (ROC-AUC). In the curve, sensitivity is plotted against 1-specificity for various decision thresholds (Figure 3-2). For example, in random forest (RF) models (see below) the proportion of trees voting for a given class can serve as a threshold for the classification decision. In this way, a sorted list of all data points can be obtained. Then, for each possible decision boundary in the ordered list, sensitivity and 1-specificity are calculated and plotted. The area under the obtained curve is used to measure the

performance of the classification model. While an AUC of 0.5 corresponds to random predictions, an AUC of 1 is achieved by a perfect classifier.

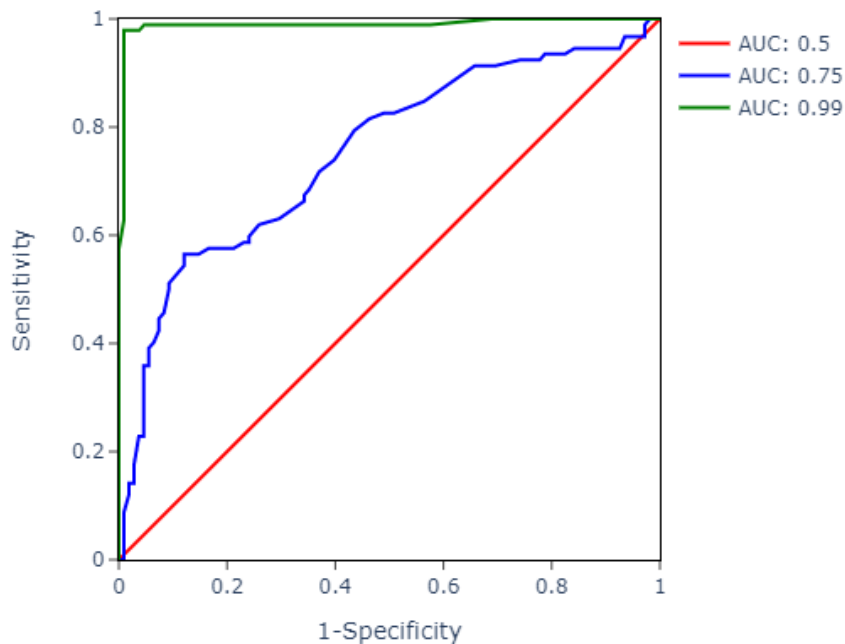


Figure 3-2 ROC curves. Shown are curves for a random classifier (AUC=0.5), a moderately performing classifier (AUC=0.75) and a nearly perfect classifier (AUC=0.99).

Different metrics have different advantages and disadvantages. For instance, accuracy is not well suited to evaluate performance on imbalanced datasets. For instance, when 99% of data points belong to the majority class, a (useless) classifier always predicting the majority class would achieve an accuracy of 0.99. Recall, precision and F1 score in contrast measure performance for a specific class and are more meaningful than accuracy when applied to the minority class. Balanced accuracy and MCC score are further metrics suitable for imbalanced datasets (Boughorbel et al., 2017).

3.4 Machine learning algorithms

3.4.1 k-Nearest neighbours

From a conceptual perspective, k-nearest neighbours (k-NN) (Dey, 2016) is perhaps the simplest ML algorithm for classification and regression tasks. It requires an integer number k (e.g. $k=5$) to be defined and a means to assess distance between instances in the data set. The k-NN algorithm considers the k nearest data points in the training set, according to the chosen distance measure, for its prediction. In the case of classification, the predicted label is based on a majority vote of the k

nearest labels. For regression, the average of the k nearest labels is computed. The principle is visualised in Figure 3-3. For both classification and regression tasks, the prediction may be adjusted by giving a higher weight to nearer neighbours.

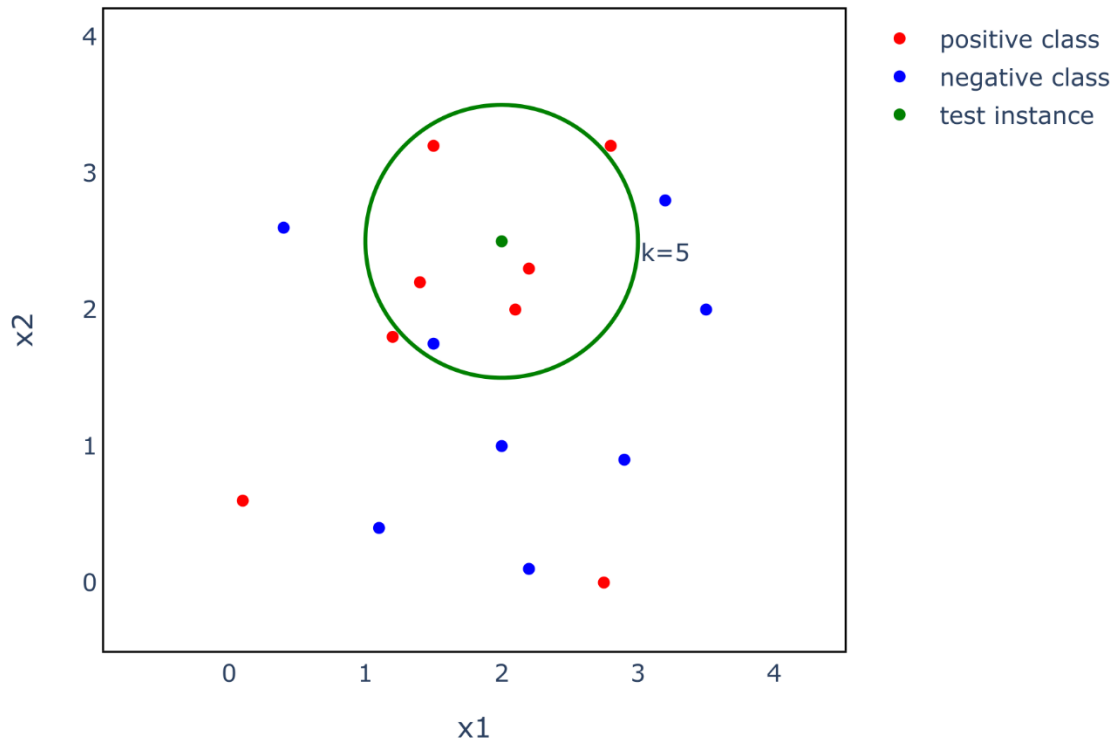


Figure 3-3 k-NN principle. Here k is selected to be five and hence the five instances closest to the test instance are considered for the prediction. The measure of distance is Euclidean distance in the 2-dimensional feature space (x_1, x_2). Using a simple majority vote for classification, the test instance is classified as belonging to the positive class.

3.4.2 Linear regression

Linear regression (Gujarati, 2019) models the relationship between a numerical label y and a d -dimensional feature vector x as a linear equation of the form

$$y = a \times x + b$$

where a and b are parameters which need to be optimised during training of the model. The training of a model is done by minimising the value of a loss function that estimates the performance of the model for a given set of parameters. In the case of linear regression, the loss function computes the sum of squared errors for all points i in the training set (size= n) as a function of the learnable parameters:

$$L(a, b) = \sum_{i=1}^n (y^{(i)} - (a \times x^{(i)} + b))^2$$

An example of linear regression with a single feature (i.e. univariate linear regression) is given in Figure 3-4. In practice, a much larger number of features may be used to obtain a linear regression model. Having a high number of features increases the risk of overfitting the training data.

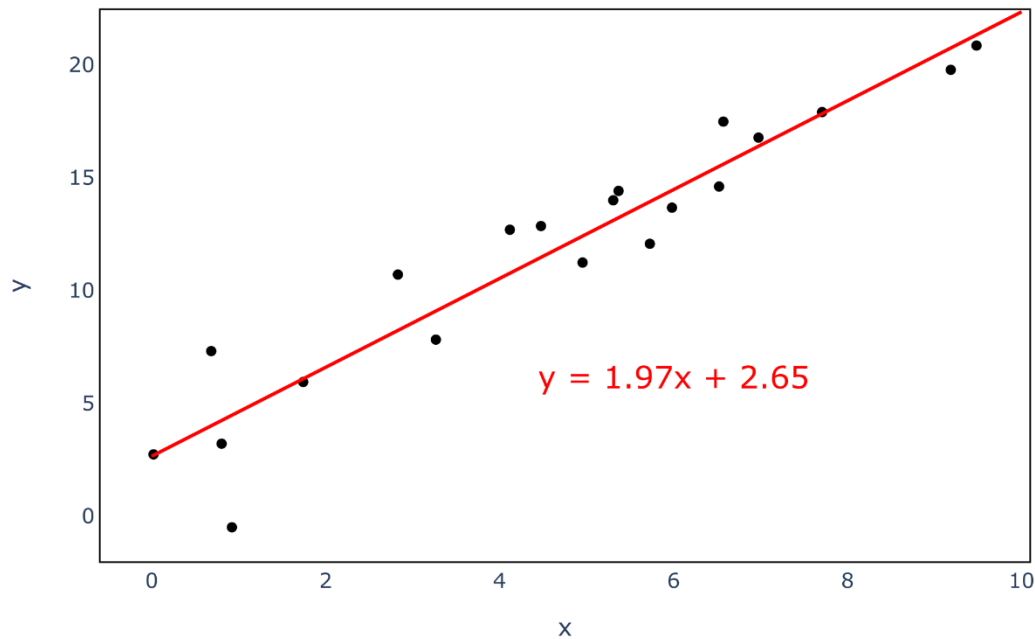


Figure 3-4 Example of linear regression. Data instances in this example consist of y as label and x as a single feature. To predict y from x , a straight line is fit using the least-square objective. Thereby $a=1.97$ and $b=2.65$ are obtained as model parameters.

Regularisation techniques can be applied to prevent overfitting of linear regression models. These techniques penalise the use of numerically large model parameters explicitly in the loss function, which leads to the training of less complex models. Different regularisation techniques can be distinguished according to the scheme applied for penalising model parameters:

- In ridge regression, the L2 norm (i.e. the magnitude of the summed squared model parameters) is penalised. (Hoerl & Kennard, 1970)
- In lasso regression, the L1 norm (i.e. the magnitude of the summed model parameters) is penalised. (Tibshirani, 1996).
- Elastic net regression penalises both the L1 norm and the L2 norm (Zou & Hastie, 2005).

In all cases, a hyperparameter λ is used to determine how strongly the different norms should be penalised relative to the least squares objective defined above. Ridge regression often yields models of better performance than lasso when the number of samples is much larger than the number of features. Lasso regression has the advantage of yielding parsimonious models, as its objective leads to irrelevant model parameters being zero. Elastic net regression was introduced to combine advantages of both approaches.

3.4.3 Logistic regression

Logistic regression (Walker & Duncan, 1967) provides a means to estimate the probability of a data point belonging to one of two possible classes. Thus, it can be used to solve binary classification problems. Conceptually, logistic regression is closely related to linear regression. In linear regression, a linear relationship between the target variable y and the features x is assumed. In contrast, in logistic regression, the log of the odds of event y being the positive class is assumed to linearly depend on x .

$$\log\left(\frac{p}{1-p}\right) = a \times x + b$$

To convert the log of the odds into a probability, the logistic (sigmoid) function is applied.

$$\Pr(y|x) = \frac{1}{1 + e^{-(a \times x + b)}}$$

The parameters a and b can be optimised in such a way that the product of computed probabilities for all training points is maximised. This approach is called maximum likelihood estimation.

$$\prod_{i=1}^n \Pr_{a,b}(y^{(i)}|x^{(i)})$$

3.4.4 Support Vector Machines

The Support Vector Machine (SVM) algorithm solves classification tasks by modelling a linear decision boundary that maximises the margin of data points next to the decision boundary, so-called support vectors (Boser et al., 1992). The principle is illustrated in Figure 3-5 for a 2-dimensional feature space where the decision boundary is a 1-dimensional line. In a d -dimensional feature space the decision boundary is a $(d-1)$ -dimensional hyperplane. SVM classifiers can be distinguished as maximal-margin classifiers (Figure 3-5A) and soft margin classifiers (Figure 3-5B) according to the method used for

determining the decision boundary. Maximal-margin classifiers strictly select a boundary by finding a hyperplane that perfectly separates the classes and maximises the margin. If the classes are not perfectly separable, soft margins are required. Soft margins allow misclassifications, but penalise them in the objective function. Even if two classes are perfectly separable, soft margins may be preferred over maximal-margin classifiers as soft margins often lead to better generalising models (i.e. not overfitting the training data).

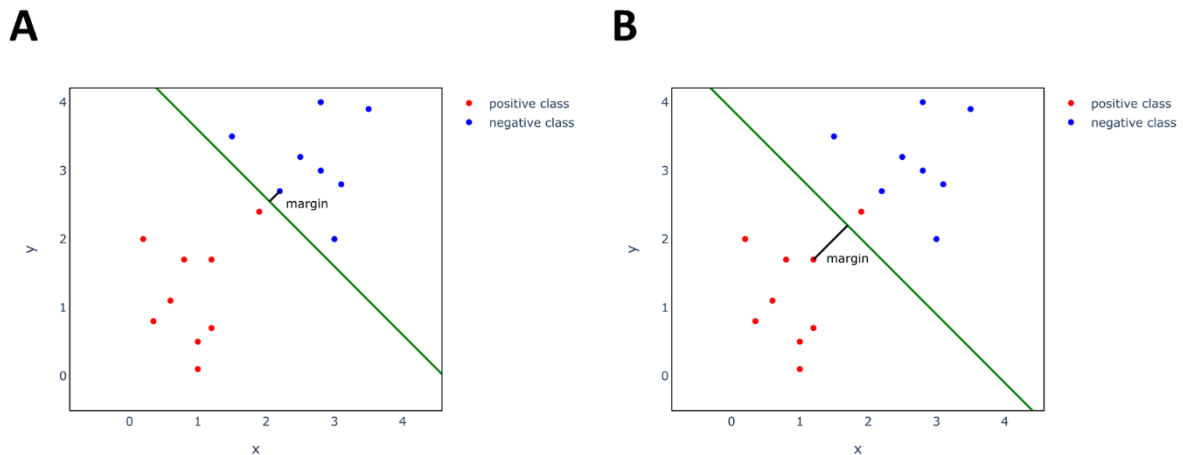


Figure 3-5 Example of SVM. **A** shows a maximal-margin classifier, whereas **B** shows a soft margin classifier. The soft margin leads to one training instance of the positive class being misclassified. The margin describes the distance of the support vectors to the decision boundary.

SVM is not limited to linear separation based on the original feature space. By applying kernel functions, the original features are mapped into a higher dimensional space. For instance, the polynomial kernel computes polynomials of the original features and then finds a linear solution in the projected (non-linear) feature space (Cortes & Vapnik, 1995).

While SVM was initially developed for classification tasks, its principle can also be adapted to solve regression tasks. Instead of fitting a decision boundary, a linear function (in the potentially expanded feature space) is fit to predict continuous values. This function is obtained by optimising simultaneously for small deviation of training data outside a pre-defined tolerated error range ϵ and for low complexity (similar to regularisation in linear regression) (Drucker et al., 1996).

3.4.5 Decision Trees

Various tree-based methods for classification and regression tasks exist. All these methods rely on the basic principles of decision trees (Loh, 2008; Quinlan, 1986).

Decision tree algorithms solve a classification or regression task by constructing a tree based on training data that separates the data according to the feature values. At each node, the data is separated based on a condition of a single feature. Terminal nodes (i.e. these are not further split) are called leaves. In Figure 3-6, a simple example of a decision tree is presented. The decision tree is used to predict whether passengers survived the sinking of a ship. To make a prediction for an instance (here a person), data is sent to a leaf node according to its attribute values. For instance, test data point P1 (sex=female, age=24) is predicted as to have survived, whereas P2 (sex=male, age=42) is predicted to have died.

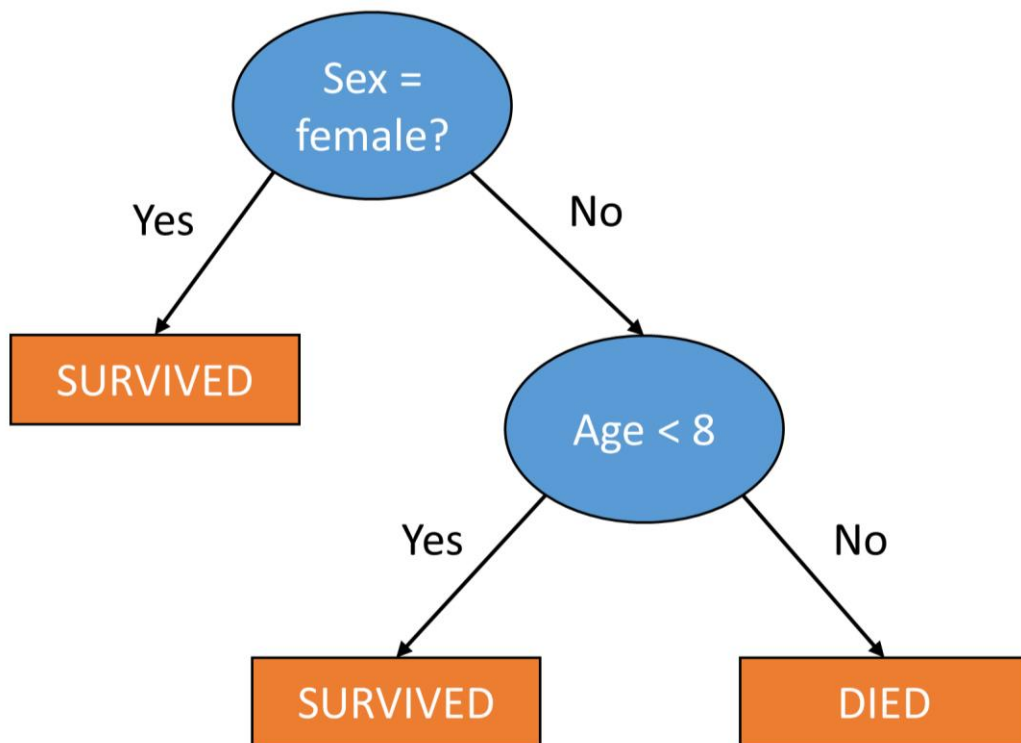


Figure 3-6 Decision tree predicting whether passengers survived the sinking of a ship. The top node is called the root. Blue, oval shapes indicate nodes that are further split according to the stated condition, whereas orange, rectangle shapes are leaf nodes.

During model training, splits are chosen according to the impurity of the original and resulting sets of data instances. For classification tasks, entropy derived from information theory (Shannon, 1948) can be used as an impurity criterion. In the binary case, the entropy H for a set of positive ($t=A$) and negative instances ($t=a$) can be computed according to:

$$H(t) = - (p(t = A) \times \log_2 p(t = A) + p(t = a) \times \log_2 p(t = a))$$

Where $p(t=A)$ is the probability of an instance of the set being positive (i.e. the proportion of positive instances in the set) and vice versa for instances being negative. Since base 2 is selected for the

formulation, an entropy of 1 corresponds to one bit of information. The information gain of a split can be computed as the difference in entropy between an original set and the sum of entropies of the resulting sets. Among potential splits, the split with the largest information gain is selected. Instead of entropy, impurity can also be measured using the Gini index according to:

$$Gini(t) = 1 - (p(t = A)^2 + p(t = a)^2)$$

For a continuous target t (i.e. a regression task), the variance of a set (of n instances) can be used as a measure of impurity, calculated according to:

$$var(t) = \frac{\sum_i^n (t_i - \bar{t})^2}{n - 1}$$

While the decision tree is grown, nodes are further separated until a stopping criterion is reached, which may be a minimum size of the leaf (number of training instances in the leaf) or a given class purity. Stopping criteria are necessary to reduce the risk of overfitting, as the algorithm otherwise would continue separating nodes until only perfectly pure leaf nodes are obtained. In this case, the noise of the data would be fit rather than trends that generalise beyond the training data. Each leaf corresponds to a certain prediction that is made, which is, for example, the majority of training labels found in this split in the case of classification and the mean of training labels in case of regression. An advantage of decision trees is their interpretability, as the predictions are based on sequential and explicitly stated decisions with respect to single features.

Other tree-based algorithms combine single decision trees to obtain an ensemble of models. Bagging and boosting techniques can be distinguished depending on how the ensemble is generated. In bagging, bootstrapping is used to create different subsets of training instances for which different decision trees are trained. Random forest is a technique based on bagging. In contrast, in boosting decision trees are trained sequentially which the aim to improve predictions for instances that were misclassified in previously learned decision trees.

3.4.6 Random forest

Random forest (RF) was developed as a more effective generalising alternative to decision trees (Breiman, 2001). The technique creates a set of decision trees and predictions are made according to a majority vote among all individual trees for classification tasks or the mean for regression tasks. The key to the success of RF is the diversity of individual trees. Diversity is achieved by two different mechanisms:

- Bootstrap sampling: each tree is trained on a subset of the training instances sampled with replacement.
- Only a randomly selected subset of features is available to find a split for each node.

The basic principle of RF is summarised in Figure 3-7.

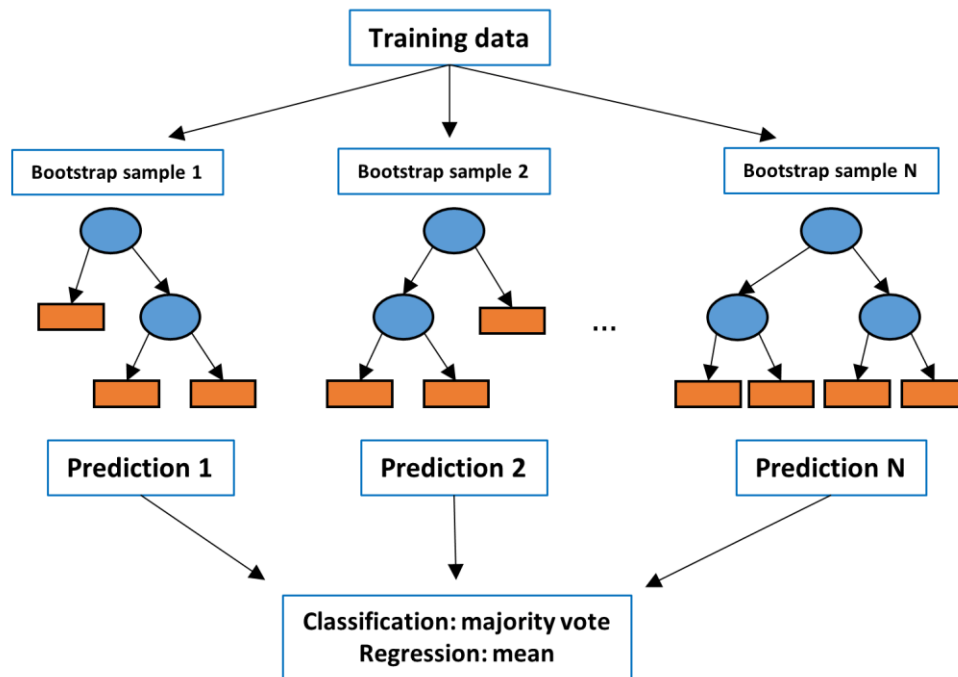


Figure 3-7 Random forest principle. The predictions of diverse trees, trained on different training data, are aggregated to obtain the final prediction.

Implementations of RF such as in scikit-learn (Pedregosa et al., 2011) contain various hyperparameters to determine the behaviour of the model. These include the selection of the impurity criterion, definition of stopping criteria for tree growing (maximum depth, minimum samples per leaf), the amount of randomly selected features used for each split, and assigning different weights to classes to counter imbalance in the dataset.

3.4.7 Gradient tree boosting

Boosting techniques combine several weak learners (which on their own perform just slightly better than random) into a powerful ensemble (Meir & Rätsch, 2003). The basic idea behind gradient tree boosting approaches is to make predictions by combining predictions of sequentially trained decision

trees, each trained to predict the residual of predictions obtained by all previously trained trees (Friedman, 2001).

For regression, the initial tree of the ensemble is a single leaf node so that the predicted value for all instances is the mean of the training data. Each following tree is then trained to predict the residual of instances (i.e. the difference between the true values and the aggregated predictions of all preceding trees). Specifically, these residuals are incorporated into differentiable loss functions and hence gradients are used to grow trees. The contribution of each newly trained tree is determined by the learning rate, a hyperparameter of the model. The principle is illustrated in Figure 3-8.

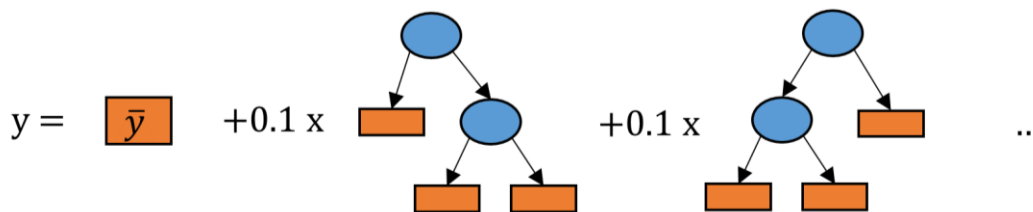


Figure 3-8 Gradient tree boosting for regression. The figure illustrates how predictions are obtained by aggregating predictions made by individual decision trees. The value 0.1 is the learning rate used to scale the contribution of each tree. Each tree is trained to predict residuals for instances given the prediction of all previous trees. The initial prediction (represented as a tree consisting of a single leaf) is the mean of training instances.

For classification tasks, the aggregated prediction is obtained as the log of the odds (similar to logistic regression), which eventually can be converted in predicted probabilities using the sigmoid function.

XGBoost was introduced as an implementation of gradient tree boosting that scales well to large scale datasets (Chen & Guestrin, 2016). In addition, it introduces some modifications to previous gradient tree boosting models. Regularisation is an effective technique to prevent overfitting of linear regression models (see above). In the XGBoost algorithm, regularisation is used to penalise both the magnitude of the output values in the leaves and the number of leaves. In addition, as in RF models, subsampling of available features for split finding is employed. Another feature of XGBoost is its ability to handle sparse data instances. In particular, for each split optimal default directions for missing data are learned to minimise the loss.

Further implementation details of XGBoost affect its scalability. For large scale datasets, XGBoost can obtain an approximate solution for growing trees instead of a slow exact solution. This is achieved by finding splits only for bins of data instead of trying all possible splits. Also, a block structure of data is used to enable parallelisation of model training.

3.4.8 Artificial neural networks and deep neural networks

Artificial Neural Networks (throughout the thesis referred to as 'neural networks') are ML models loosely reminiscent of a biological brain in the sense that they contain neurons that exchange signals. Neural networks consist of nodes (neurons) arranged in different layers, which include an input layer, one or more hidden layers and an output layer. Neural networks containing more than one hidden layer are referred to as Deep Neural Networks (DNNs) (Goodfellow et al., 2016). The input layer takes the feature vectors of data points as input. Thus, it consists of as many nodes as features used for the task. The output layer represents the given prediction by the neural network for a given data point, while the hidden layers are intermediate representations of a data point. Overall, a neural network is a complex function mapping an input vector to an output vector. With the exception of the input layer, every node receives input from every node in the previous layer. The activation of a single neuron is a linear combination of the activations of the neurons in the previous layer with the addition that a non-linear activation function is applied. Thus, the activation of neuron j in layer l is computed by

$$h_j^l = f(W \times H + b)$$

Where $f()$ is an activation function, W is the vector of weights incoming from the neurons in layer $l-1$, H is the activations of the neurons in layer $l-1$ and b is the bias of the neuron. By applying a non-linear activation function, such as the sigmoid function or the ReLU (rectified linear unit) function (which does not modify positive numbers and turns negative numbers to 0) (Agarap, 2019), the network is capable of expressing non-linear relationships between the input vector and the output vector. The design of the output layer mirrors the nature of the task the neural network aims to fulfil. The following cases shall be considered:

- Single task regression: The output is a single numeric value and therefore the output layer consists of a single neuron. Generally, no activation function is required for the neuron. If the outputs must not be negative, the ReLU function can be applied.
- Multi-task regression: If a neural network is trained for n regression tasks simultaneously, the output layer consists of n neurons, each representing the prediction for a distinct task. The outputs are treated the same way as for single task regression.
- Single task binary classification: The output is single binary value and therefore the output layer consists of a single neuron. A sigmoid function is applied to scale the output in the range from 0 to 1, which is interpreted as the probability of the positive class. To get a prediction from this value, numbers above 0.5 are considered positive, whereas numbers below 0.5 are considered negative.

- Multi-task binary classification: As for multi-task regression n neurons are required to make predictions for n distinct tasks. The outputs are treated the same way as for single task binary classification.
- Multi-class classification: The number of neurons is equal to the number of classes in the task. To achieve a probability distribution as output (the output of each neuron represents the probability for a class) the softmax function is applied. This function takes the whole set of neurons in the output layer as input, applies the standard exponential function to each value and then divides each exponential by the sum of the exponentials. As a result, each value will be in the range $(0,1)$ and the sum of all values will be 1.

The form of neural networks described above is also called fully-connected feedforward, as information flows only in one direction from input to output and no cyclic connections between neurons exist. A schematic depiction of the above described architecture is given in Figure 3-9.

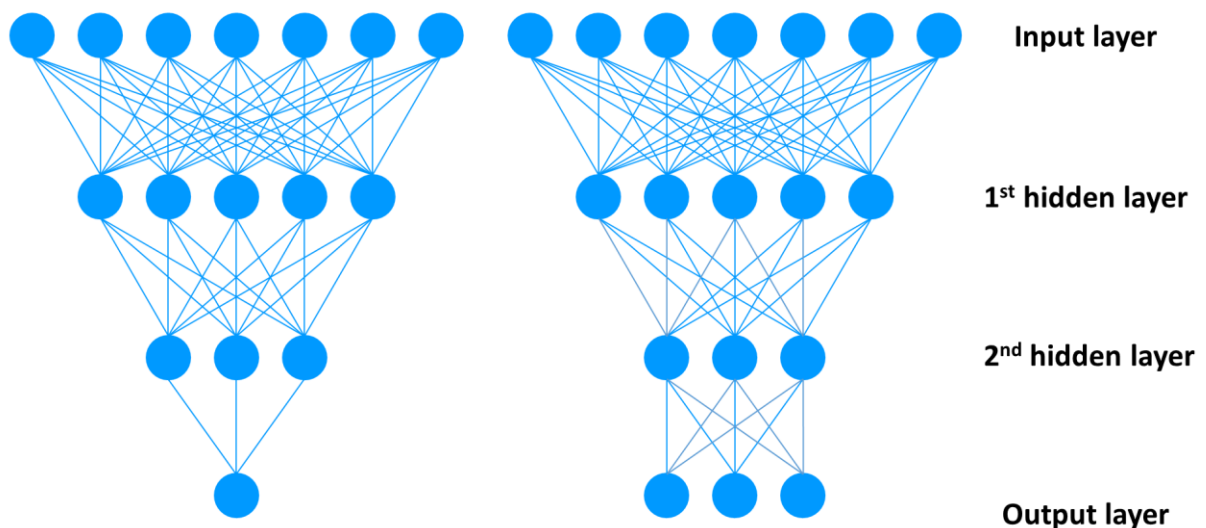


Figure 3-9 Architecture of feedforward DNN models. Shown are a network with a single output neuron (left) and a network with three output neurons (right). The network with a single neuron may be used for single task regression or binary classification tasks. The other network may be used for multi-class classification or multi-task settings. Neurons typically contain more neurons in input and hidden layers than shown in this simplified figure.

To train a neural network model, weights and biases are adapted to reduce loss between training labels and predictions made by the neural network. Various loss functions can be used, depending on the nature of the task (Brownlee, 2019). For regression problems, the mean squared error between predicted and true labels can be used (comparable to linear regression). A popular choice for binary classification tasks is binary cross entropy (BCE) loss, which for a single instance is given by:

$$BCE\ Loss = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

where y is the true label (0 or 1) and \hat{y} is the predicted probability of the instance. For multiple instances the BCE loss can be summed or averaged depending on method implementation. In essence, BCE loss evaluated how close a prediction is to a 'perfect' prediction (i.e. the predicted probability is identical to label). For multi-class classification tasks, the (general) cross entropy loss can be used where for each class the distance between predicted probability and true class is evaluated. For multi-task settings, the same loss functions can be used by summing or averaging the losses across the different tasks.

Training of neural networks is normally done using gradient descent or related methods. These techniques require the determination of the gradient of the loss function with respect to each tuneable parameter (i.e. weights and biases) using backpropagation (Rumelhart et al., 1986). Gradients are found by applying the chain rule for differentiation starting from parameters in the last hidden layer and moving backwards through the network. The gradient can be understood as the direction in which the respective parameter needs to be changed to minimise the loss. An optimisation step is performed by updating the parameters in the direction of the gradient. The learning rate, a hyperparameter set by the user, determines the size of the step.

Computing the gradients using the entire training set may be computationally expensive. This can be circumvented by using stochastic gradient descent (SGD). In SGD, instead of computing the gradients for the full data set in each optimisation step, gradients are iteratively computed for batches (i.e. subsets) of the training set. One training epoch is completed when each training instance has been used once for optimisation within a batch. The training of a neural network may consist of more than one epoch. Batch size and number of epochs are further hyperparameters of the training process. More sophisticated optimisation techniques like Adam (Kingma & Ba, 2014) have been developed. A key characteristic of Adam is that learning rates are individually adapted for different parameters using the history of updates. This means that update steps for a parameter can be increased if high gradients were found for this parameter in previous steps to accelerate learning. In analogy to the physical phenomenon, this principle has been named momentum.

Further techniques have been developed to prevent neural networks from overfitting training data and to improve generalisation capability. Similar to linear regression, L1 or L2 regularisation can be applied to penalise the magnitude of learned weights (Le Roux & Bengio, 2007). Another regularisation technique is dropout (Srivastava et al., 2014). Dropout means that randomly sampled hidden neurons or input neurons including all their connections are removed from the network during training. The motivation behind this technique is to prevent complex co-adaptations of different neurons. Put differently, hidden neurons should encode features that are meaningful by themselves. A third

technique is early stopping of neural network training (Yao et al., 2007). In this technique, the generalisation error of a model is monitored during training using a validation set and the training is terminated when no further improvement on the validation set is observed.

Fully-connected feedforward neural networks as described above can be used for supervised ML tasks where numerical features in the form of a d -dimensional vector for each instance are given. Different architectures are required for different types of data (LeCun et al., 2015). Convolutional architectures are employed for image (Convolutional Neural Networks (CNNs)) or graph data (Graph Convolutional Networks (GCNs)) where the spatial arrangement of pixels or nodes is meaningful. In Recurrent Neural Networks (RNNs), neurons can form cyclic connections and these architectures may be used on sequential data like strings of variable length.

3.4.9 Matrix factorisation

Matrix factorisation techniques can be used to fill gaps in a sparse (i.e. incomplete) matrix by making predictions (Koren et al., 2009). A sparse matrix may refer to situations when not all features for data instances are known or in a multi-task setting when the label for an instance is only known for some of the tasks. However, the distinction between features and labels might disappear in such situations as target labels may be used as features to predict other target labels.

The basic principle behind matrix factorisation is to factorise the data matrix into the product of two smaller matrices. A matrix X of dimension $m \times n$ (m rows and n columns) can be decomposed into the product of U and V :

$$X_{m \times n} = U_{m \times d} \times V_{d \times n}$$

The entry of matrix X for the i^{th} row and the j^{th} column is given by the dot product of the i^{th} row of U with the j^{th} column of V :

$$X_{i,j} = U_{i,1} \times V_{1,j} + U_{i,2} \times V_{2,j} + \dots + U_{i,d} \times V_{d,j}$$

Matrix factorisation methods gained interest during the Netflix Prize competition, which had the aim to improve systems that recommend movies to users based on their previous ratings. The task was to predict the ratings of movies by users using a matrix of given ratings. In the above equation for matrix factorisation, the matrices U and V can be considered representations of the users and movies, respectively, in a joint d -dimensional latent factor space.

To fit a matrix factorisation model, the matrices U and V have to be found. Objective functions seek to minimise the distance between observed entries in X and the corresponding prediction resulting from the product of U and V . Since the matrix X may be very sparse, overfitting to the few observed entries is a problem. Overfitting can be countered by including regularisation terms in the objective function, which penalise high values in U and V . An objective function of the following form is obtained:

$$\min_{u,v} \sum_{(i,j) \in I_X} (X_{ij} - u_i v_j^T)^2 + \lambda_u \|u\|_F^2 + \lambda_v \|v\|_F^2$$

where u_i and v_j are the latent vectors for the i^{th} row of U and the j^{th} column of V , I_X the set of filled cells in X , X_{ij} is the observed value in the i^{th} row and the j^{th} column of X , λ_u and λ_v are regularisation parameters and $\|\cdot\|_F$ is the Frobenius norm. The Frobenius norm of a matrix A is defined as:

$$\|A\|_F = \sqrt{\sum_{i,j} (A_{i,j})^2}$$

Another way to construct a matrix factorisation model is by using a probabilistic approach (Salakhutdinov & Mnih, 2007). Macau is an example for probabilistic matrix factorisation (Simm et al., 2015). A notable property of Macau is that, in addition to the sparse matrix, it may use features that describe the entities belonging to rows and columns of the matrix (also called side information). A formal mathematical description of probabilistic matrix factorisation in general and Macau in particular is provided in Appendix A.

3.5 Conclusion

The present chapter introduced ML techniques which are commonly used to construct QSAR models. Focus was put on relevant algorithms as well as strategies to evaluate the models. The next chapter (Chapter 4) will introduce the concept of QSAR modelling and describe aspects relevant to the use of QSAR models for the case of toxicity prediction.

Chapter 4 QSAR modelling for toxicity prediction

In the preceding chapters, an introduction to toxicity assessment of chemicals (Chapter 2) as well as ML (Chapter 3) was provided. QSAR modelling can be understood as ML applied to chemicals, namely to predict properties of chemicals (including bioactivity and toxicity) based on their chemical structure. The present chapter will introduce the theoretical foundations of QSAR models as well as their application to predict toxicity.

In particular, the chapter introduces chemical descriptors as representations suitable for QSAR modelling (4.1). Then, the concept of molecular similarity is introduced which implicitly forms the basis of QSAR modelling (4.2). In the next section (4.3), QSAR modelling is described as a ML model linking chemical structures to properties of chemicals. In section 4.4 curation of chemical structures is described as an important pre-processing step to QSAR modelling. An overview of toxicity data available for QSAR modelling is provided in section 4.5. Section 4.6 describes known determinants of the success of QSAR modelling and introduces the concept of an applicability domain. The current use of QSAR models in regulatory toxicology is summarised in section 4.7. Due to the importance of neural network models in this thesis, their use for toxicity prediction is briefly reviewed in section 4.8.

4.1 Representation of chemical structures in computers

Cheminformatics has been defined as “the application of informatics methods to the solution of chemical problems” (Gasteiger, 2006). The prerequisite for the numerous cheminformatics application are means to represent chemical structures computationally (Warr, 2011). The most common way among chemists to represent the structure of a molecule is as a diagram showing its atoms and their connections in two-dimensional space (see Table 4-1, first row).

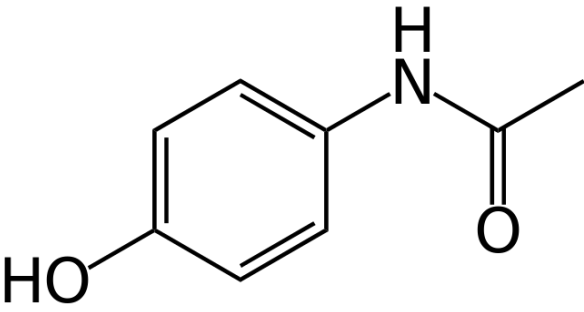
In such diagrams hydrogen atoms typically are not explicitly depicted (unless bound to a heteroatom), as their presence next to carbon atoms can be easily inferred by a chemist. This form of representation can be abstracted to a two-dimensional graph, a mathematical object consisting of nodes (representing atoms) which are connected by edges (representing chemical bonds).

Connection tables (Gluck, 1965) enumerate all atoms of a molecule and state the connection between the atoms. Thus, they can be considered as a tabular form of a molecular graph. Connection tables

form the basis of MOL files (Dalby et al., 1992). An example for acetaminophen retrieved from the DrugBank database (Wishart et al., 2018) is provided in Table 4-1 (second row). The upper block (atoms block) contains the atoms with their element symbols and 2D (or possibly 3D) coordinates. The bonds block below states which atoms are connected as well as the bond type (single, double, triple).

A more concise way to store chemical structures is using line notations (strings). In line representations, a molecule is represented as a string. A popular line representation is SMILES (Simplified Molecular Input Line Entry System) (Weininger, 1988). It describes the structure as a sequence of letters and special characters, capable of expressing different bond types, branching and rings (see Table 4-1, third row). The InChI (International Chemical Identifier) notation (see Table 4-1 fourth row) has been developed by the IUPAC as a standardised line representation of a molecule (Heller et al., 2015). Compared to SMILES, InChI is more difficult to read by a human, but in principle the chemical structure can be generated from an InChI. An InChI Key (see Table 4-1, fifth row) represents a condensed form of an InChI (obtained through hashing) which was designed to facilitate searching for a structure in a database.

Table 4-1 Various chemical representations of acetaminophen (paracetamol).

Chemical graph	
Connection table	<pre> 316 Mrv0541 02231214352D 11 11 0 0 0 0 999 V2000 2.3645 -2.1409 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.7934 1.1591 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.3645 1.1591 0.0000 N 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.3645 0.3341 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.0790 -0.0784 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.6500 -0.0784 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.0790 -0.9034 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.6500 -0.9034 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.3645 -1.3159 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.0790 1.5716 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.0790 2.3966 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 9 1 0 0 0 0 2 10 2 0 0 0 0 3 4 1 0 0 0 0 3 10 1 0 0 0 0 4 5 2 0 0 0 0 4 6 1 0 0 0 0 5 7 1 0 0 0 0 6 8 2 0 0 0 0 7 9 2 0 0 0 0 8 9 1 0 0 0 0 10 11 1 0 0 0 0 M END </pre>
SMILES	<chem>CC(=O)NC1=CC=C(O)C=C1</chem>
InChI	InChI=1S/C8H9NO2/c1-6(10)9-7-2-4-8(11)5-3-7/h2-5,11H,1H3,(H,9,10)
InChI Key	RZVAJINKPMORJF-UHFFFAOYSA-N

All the representations presented in the table capture 2D structures of the molecule. Since molecules in reality are 3D objects, there are also representations that capture the location of the atoms in 3D space. It must be stated, however, that due to rotation about single bonds many different potential conformations can be realised by a molecule. 3D representations generally seek to provide an energetically favoured conformation. The Cambridge Structural Database is a repository containing 3D structure data obtained using X-ray crystallography (Groom et al., 2016).

None of the representations introduced so far is amenable to the construction of QSAR models (at least in a traditional way), because these typically require numerical representations of fixed length. This can be achieved using molecular descriptors. Molecular descriptors represent a molecule by

describing its properties. One can distinguish between different types of molecular descriptors: constitutional, physicochemical, topological and 2D fingerprints, as described below.

Constitutional: Constitutional descriptors represent simple counts of features such as atoms, heteroatoms, functional groups with certain properties (e.g. hydrogen bond acceptors or donors), bond types or ring systems (Leach & Gillet, 2007). These can be easily obtained from a connection table representation.

Physicochemical: Physicochemical properties describe properties of molecules as a whole as opposed to describing their constituents. Examples are molecular weight, lipophilicity (typically represented by log P which is the octanol water partition coefficient) or TPSA (topological polar surface area) (Ertl et al., 2000). For properties that, in principle, are the result of a wet-lab experiment (like log P), methods to computationally estimate the property have been developed to enable computation for large sets of compounds (Mannhold et al., 2009). Physicochemical properties may be key determinants for bioactivity. For instance, the relationship between lipophilicity and anaesthetic activity is well established (Glave & Hansch, 1972).

Topological: Topological indices are derived from the 2D molecular graph and capture characteristics like size, shape and branching (Leach & Gillet, 2007). A prominent example is the Wiener Index which is the sum of all topological (through bond) distances between all atoms in the molecule (Wiener, 1947). Other descriptors capture the connectivity of atoms by considering the number of valence electrons not bonding to hydrogen (Kier & Hall, 1981).

2D Fingerprints: Structural fingerprints represent molecules in the form of bit vectors where a one indicates the presence of a certain structural feature and a zero indicates absence. Alternatively, fingerprints can be obtained as count-based where each position in the fingerprint indicates the number of occurrences of a certain feature. Fingerprints can be distinguished as dictionary-based fingerprints and hashed fingerprints. In a dictionary-based fingerprint each bit corresponds to an element of a pre-defined dictionary of fragments (e.g. MACCS keys) (Durant et al., 2002). Hashed fingerprints, in contrast, generate fragments from the structure according to defined rules and map the fragments to a vector of defined length. Different fragments may be mapped to the same bit.

A popular hashed fingerprint, specifically designed for QSAR modelling, is the extended connectivity fingerprint (ECFP) which determines all unique fragments up to a user-defined radius starting from each atom as a centre (Rogers & Hahn, 2010). If a radius of up to one bond is chosen, the obtained fingerprint is called ECFP2 (2 referring to the 'diameter' of up to two bonds for the obtained

fragments). Equivalently, if the chosen radius is 2, ECFP4 is obtained, and so on. The following properties of atoms constituting the fragments are considered:

- The number of direct heavy atom neighbours
- The valence of the atom (ignoring hydrogen atoms)
- Atomic number (chemical element type)
- Atomic mass (to distinguish different isotopes)
- Atomic charge
- Number of attached hydrogen atoms
- Whether the atom is part of at least one ring

The generation of ECFPs is illustrated in Figure 4-1. The open-source cheminformatics toolkit RDKit (*RDKit: Open-Source Cheminformatics*, n.d.) implements Morgan fingerprints which are roughly equivalent to ECFPs.

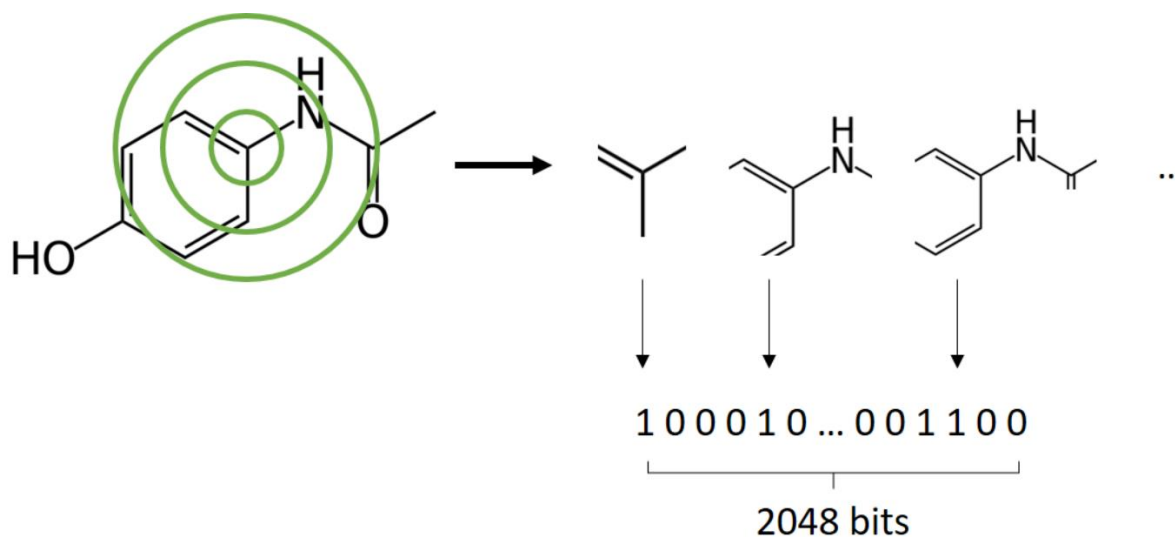


Figure 4-1 Representation of molecules using ECFPs. To encode a chemical structure as an ECFP4, the environments of all atoms up to a radius of 2 bonds are determined (the green circles encapsulate the environments of radii 0, 1 and 2 bonds for the central carbon atom). Then, all unique environments included in the molecule are subjected to a hash function and folded to a binary fingerprint of fixed length (here 2048 bits), where 1s indicates the presence of a certain atom environment and 0s their absence. This provides a numerical representation of a molecule which is amenable to standard QSAR modelling.

4.2 Molecular similarity

The similarity principle states that structurally similar molecules tend to have similar properties (M. Johnson et al., 1988). Consequently, a set of molecules with similar properties can be found by searching for molecules with similar structures. In toxicity assessments, this concept is explicitly

exploited in the earlier described read-across method to predict the toxicity of untested substances from known activities of similar substances. The application of this concept requires a means to evaluate the similarity of molecules. One possibility is to check if two molecules contain a certain common 2D substructure or 3D pharmacophore (i.e. a spatial arrangement of molecular features). However, this approach merely splits a given library of molecules into those which contain the considered substructure or pharmacophore and those which do not. More sophisticated methods are required to quantify the similarity between molecules. A popular way to quantify the similarity of two molecules is by calculating a similarity coefficient based on binary chemical fingerprints (Willett, 2006). Generally, these methods quantify the similarity in the range from 0 to 1, where 0 means the absence of any common features and 1 means identical representations (but not necessarily identical molecules). The most widely used similarity coefficient is the Tanimoto coefficient which is defined as

$$Tc(A, B) = \frac{c}{a + b - c}$$

where a is the number of bits set to 1 in the fingerprint of molecule A, b is the number of bits set to 1 in molecule B and c is the number of bits set to 1 in both fingerprints. Other similarity coefficients are the Dice coefficient and the Cosine coefficient defined by the following equations:

$$Dice(A, B) = \frac{2c}{a + b}$$

$$Cosine(A, B) = \frac{c}{\sqrt{ab}}$$

where a , b , and c are defined in the same way as in the Tanimoto coefficient. More generally, the similarity between two objects can be considered as the counterpart to the distance between two objects. The introduced similarity coefficients can be converted to a measure of distance using the formula:

$$\text{distance} = 1 - \text{similarity}$$

A widely used distance coefficient is Euclidean distance (Jochum et al., 1980) defined by

$$D_{\text{eucl}}(A, B) = \left[\sum_{i=1}^N (x_{iA} - x_{iB})^2 \right]^{1/2}$$

Where A and B are objects with N features x_{iA} and x_{iB} are the values of the i^{th} feature of A and B. Such a general distance metric can measure the distance between two molecules represented as either binary fingerprints or as a set of arbitrary molecular descriptors of continuous nature. The Euclidean distance between two objects is 0 if their representations are identical, however, it is

unbounded since infinitely high values of distance can occur theoretically. In conclusion, similarity and distance between two molecules have no universal character. Instead, they depend on the context in terms of the molecular representation and the chosen coefficient.

4.3 Basic principles of QSAR modelling

A QSAR model can be considered to be any function that uses some representation of the chemical structure as input and predicts a biological activity relying on mathematical or statistical relationships (Cherkasov et al., 2014). Thus, QSAR models generally can be constructed for any property or bioactivity of a molecule. Corwin Hansch is considered as the pioneer of QSAR modelling. Using linear regression, he expressed biological responses of compounds within a chemical series as a function of the molecules' lipophilicity ($\log P$) as shown in the following equation (Hansch et al., 1963):

$$\log\left(\frac{1}{C}\right) = k_1 \log P + k_2 \sigma + k_3$$

where C is the concentration of the compound responsible for the defined biological activity and σ is the Hammett parameter corresponding to the particular substitution pattern of a benzene derivative (Hammett, 1937). This equation is limited to a small series of similar molecules. More sophisticated methods are required to model relationships of large heterogeneous datasets and to account for non-linear relationships. Consequently, the field has expanded extensively since then in terms of chemical descriptors, functions/algorithms that are employed and properties that are modelled. Molecular descriptors were introduced above. The range of algorithms used for QSAR modelling includes simple approaches like linear regression and k-NN, tree-based methods (decision tree, RF, gradient tree boosting) and neural networks (shallow or deep). An overview of these algorithms was provided in Chapter 3. Generally, the construction of a QSAR model is an example of a ML workflow as introduced in Chapter 3. First, the molecules in the dataset are expressed as molecular descriptors. Then, a QSAR model is trained on training data. As described for ML in general, the QSAR model needs to be evaluated on compounds that were not used during training to demonstrate predictivity on new compounds.

4.4 Preparing chemical structures for QSAR modelling

In order to train a QSAR model, compounds need to be represented using some chemical descriptor. However, a crucial step is to curate chemical data beforehand. One problem is that a dataset may contain incorrect structures (i.e. the given structure does not correspond to the tested chemical). A

study investigated public and private databases and found error rates in the range from 0.1 to 3.4% (Young et al., 2008). Another problem is that chemical structures in a dataset may be represented inconsistently (e.g. certain functional groups in different tautomeric form) or such that it cannot be properly represented by traditional chemical descriptors (e.g. complex bonds or compound mixtures). Several chemical data curation steps have been suggested to remedy these problems (Fourches et al., 2010).

Where possible, inorganic and organometallic compounds should be removed because commonly used molecular descriptors are in many cases not suitable to represent them. Also, mixtures of different chemicals should be removed, unless the observed activity can be attributed with confidence to a single component of the mixture, which then can be retained. Moreover, the descriptors often are not capable of representing salts and thus inorganic counterions need to be removed. Then, the molecules should be neutralised (removal of charges) or the charges adjusted to the experimental pH. In the next step, certain chemotypes with multiple possible structural representations like nitro groups or tautomers need to be converted to a standardised manner to avoid inconsistencies within the data set. It is also imperative that a specific structure is only contained once in a dataset. Duplicates may be in the dataset due to experimental replicates or introduced by the previously performed standardisation steps. Duplicates can be handled by averaging experimental values or, in the case of conflicting evidence, by removing all instances to avoid potential errors. As a final step, manual checking of the structures in the data set is recommended.

4.5 Toxicity data for QSAR modelling

A prerequisite for a successful QSAR model is the availability of sufficient data of good quality. In the area of toxicity prediction, modelled events may be the experimentally measured outcomes of in vitro toxicity assays or in vivo toxicity studies. A brief overview of available in vitro and in vivo data is provided in this section.

4.5.1 In vitro toxicity data

The two largest databases (>10⁶ compounds) for in vitro data are PubChem (Wang et al., 2017) and ChEMBL (Gaulton et al., 2017), which are provided by the National Center for Biotechnology Information (NCBI) and the European Bioinformatics Institute (EBI), respectively. These databases are depositories for all kinds of in vitro bioactivity data, rather than being specific for toxicity effects. In

contrast, Tox21 (Thomas et al., 2018) and ToxCast (Richard et al., 2016) are projects designed to generate in vitro data presumed to be highly relevant for toxicological events in vivo. As mentioned earlier, Tox21 is a federal inter-agency collaboration in the US. A cornerstone of the project was performing HTS experiments on approximately 10,000 chemicals in approximately 70 assays. ToxCast on the other hand is a project carried by the EPA. While the ToxCast data overlaps with Tox21 to some extent, it focusses on a broader set of assays with a smaller set of compounds. More specific is the ISSTOX (Istituto Superiore di Sanità Toxicity) database, which contains data relevant for chemical carcinogenicity including mutagenicity data in bacteria (Ames test) (Benigni et al., 2008).

4.5.2 In vivo toxicity data

As described in Chapter 2, in vivo toxicity data are generated in animal studies according to guidelines. A public database containing the results of various in vivo animal studies for over 1,000 compounds is EPA's ToxRefDB (Watford et al., 2019). A similar resource, though not publicly available, is the eTox database (Sanz et al., 2017). It was generated in a cooperation of pharmaceutical companies, academic institutions and small and medium-sized enterprises and contains in-house data of the companies for almost 2,000 compounds. The objective of the project is to facilitate data exchange among pharmaceutical companies and to exploit the data for predictive modelling of toxicity. The CEBS (chemical effects in biological systems) database contains toxicity studies generated by the National Toxicology Program (NTP) (Lea et al., 2017), which includes studies on genotoxicity, carcinogenicity, reproductive toxicity and immunotoxicity. As a repository dedicated to chemical carcinogenicity, the above mentioned ISSTOX database also contains in vivo carcinogenicity studies. A comprehensive dataset on acute rodent toxicity has been assembled by the NICEATM (National Toxicology Program Interagency Center for the Evaluation of Alternative Toxicological Methods) and the EPA (Kleinstreuer et al., 2018).

4.6 Determinants of successful QSAR modelling

The usefulness of a QSAR model depends on its capability to make accurate predictions for novel compounds. Therefore, understanding under which circumstances a QSAR model will be most successful has been, and still is, an area of active research. This section discusses certain factors whose impact on QSAR models has been studied. Subsequently, the concept of an applicability domain is

introduced, which aims to determine whether or not the prediction of an individual compound is reliable.

One challenge that may occur for classification models is data imbalance which has been mentioned in the previous chapter and is not specific to QSAR models. In particular, data imbalance may result in models making accurate predictions for majority class compound, but inaccurate predictions for minority class compounds (Zakharov et al., 2014). Approaches to account for data imbalance in QSAR models include increasing the cost of misclassifying minority class compounds (Guha & Jurs, 2005), resampling techniques (Bae et al., 2021) and adapting the classification threshold (Esposito et al., 2021).

Another aspect is the amount of available data to train a QSAR model. For a given modelled endpoint and identical test sets, performance tends to increase if more training data becomes available over time (Sheridan, 2022). Experiments on very small datasets (~100 compounds) have shown that the effect of reducing a dataset's size depends on the particular dataset (Roy et al., 2008). For instance, a large decrease in performance was found for a local dataset on anti-HIV activity of thiocarbamates, while no significant impact of dataset size was found for a diverse dataset on bioaccumulation. In contrast, for very large datasets (ten thousands of compounds), adding more data to the training set tends to have very little impact on overall model quality, as the additional data might not lead to a larger coverage of chemical space (Aleksić et al., 2022). When comparing different datasets, training set size does not seem to be a good indicator of success. QSAR models with small datasets may be successful (at least in a local context), while models with large dataset size may still fail for other reasons (see following paragraphs).

QSAR models are based on the principle that similar compounds tend to have similar properties. Therefore, it is not surprising that chemical similarity between training and test compounds is a good indicator for the success of QSAR models (Sheridan et al., 2004). Similarity may be evaluated for instance as the (average) similarity value to the nearest neighbour(s) in the training set or the number of neighbours in the training set above a selected similarity threshold. The relation may still hold true if the chemical descriptors used to assess chemical similarity are not identical to those used for model training.

Another challenge for QSAR models is the presence of activity cliffs (Maggiore, 2006). The term is derived from the concept of activity landscapes, in which the x- and y-coordinates encode chemical space while the z-coordinate represents bioactivity on a certain target. Activity cliffs refer to compounds that are chemically similar, yet possess greatly different activities which in activity landscapes resemble topographical cliffs. The presence of activity cliffs in a dataset is a strong

determinant for QSAR model performance (Golbraikh et al., 2014), as these are very challenging to predict.

Another aspect impacting on the model performance is experimental uncertainty of generated bioactivity data. It was shown that when random noise is artificially added to a dataset, the performance of otherwise predictive QSAR models drops depending on the amount of added noise (Sheridan et al., 2020). A commonly stated assumption is that the model cannot be more accurate than the experimental method used to generate its training data. In other words, this would mean that the experimental uncertainty represents a hard limit on model performance. A recent study challenged this view by analysing model performances when random noise was added to the training data but not the test data (Kolmar & Grulke, 2021). Their findings suggest that a QSAR model may indeed be more accurate than the training data, yet this may in practice not be measurable due to the experimental uncertainty present in the test data.

Notably, the different factors impacting the performance of QSAR models are not completely independent. For instance, decreasing the training set size may remove chemically similar compounds or the presence of apparent activity cliffs in the dataset may be due to experimental uncertainty rather than being real discontinuities in the activity landscape.

Users of a QSAR model may not be interested in the global performance of a QSAR model. Instead, they might wonder how reliable the prediction for an individual compound of interest is. The concept of an applicability domain (AD) was developed to address this need. The AD of a QSAR model defines the area of chemical space and response space where predictions can be made with a given reliability.

Several ways to define an AD have been proposed over the years. Mathea et al. identified two distinct strategies that are in use to determine the AD of a QSAR model (Mathea et al., 2016). Approaches known as novelty detection define the AD solely in terms of the space of molecular descriptors. This is done by either considering the range of the molecular descriptors (or of a projection), the distance of a molecule to its nearest neighbours in the training set, or by analysis of the local density in the area of the molecule to be predicted. All these approaches assume that reliability of predictions decreases as the remoteness to the molecules in the training set increases. In contrast, confidence estimation also takes the activity labels of training instances into account. The activity of a molecule may be difficult to predict even though it possesses a low distance to the training set in descriptor space (see activity cliffs). For a classification model (distinguishing between active and inactive molecules), confidence estimation can be achieved by considering the distance of a new molecule to the decision boundary, or for ensemble models by considering the agreement of single predictors.

Novelty detection and confidence estimation can be considered as complementary strategies covering different aspects of the concept of ADs.

Hanser et al. suggested a three-step framework (applicability, reliability, decidability) to evaluate if a model is suitable to predict the bioactivity of a given substance in the context of human safety assessment, where a chemical falsely predicted as harmless might have serious consequences (Hanser et al., 2016). In the first step, the model is evaluated to determine if it is generally applicable to the query substance. This means that the query substance must be of a substance class included in the model and its descriptors must lie within the descriptor range of the model. In the next step, model reliability is checked by investigating the density of data around the query substance and how well the model performs in this area, e.g. in a cross-validation. Put differently, reliability considers the quantity and quality of information in the area of descriptor space around the query. Finally, it is determined if the evidence is sufficient to make a (potentially high-stake) decision. Methods suggested for this step are essentially confidence estimation methods such as those described above. Notably, the framework demands these steps are considered in the described order, as each stage assumes that the previous one is valid and they cannot compensate for each other. The suggested framework is of fairly general nature and thus can be implemented using various algorithms. It represents a reasonable strategy to fulfil the specifications implied in the concept of ADs for QSAR models in an appropriate manner.

4.7 Use of QSAR models in regulatory toxicology

Among many potential applications of QSAR models, predicting toxicity is associated with some special challenges. Generally, one can distinguish between the following stages in which a QSAR model might be used for toxicity prediction. Either it is used at an early screening stage aiming at filtering out compounds (e.g. drug candidates) with undesired toxic properties, or it is used to demonstrate safety in the context of regulatory decision making (e.g. to support market approval of a drug). It is apparent that in the latter case mistakes (especially chemicals falsely predicted as safe) may result in serious consequences if the population is exposed to hazardous chemicals. Thus, predictions made in these cases need to be of high confidence. In contrast, for the screening task, overlooked toxicological liabilities can be detected in later testing stages. To address these special requirements, the OECD has defined five principles that need to be fulfilled by a QSAR model to be considered for a regulatory purpose (OECD, 2004).

A defined endpoint: The endpoint being modelled and the experimental protocol should be stated. This is important to demonstrate the relevance of the modelled property for the respective toxicological endpoint. For instance, a QSAR model trained to predict the outcome of the Direct Peptide Reactivity Assay (DPRA) (Lalko et al., 2012) can be considered as relevant for toxicity associated with skin sensitisation as it measures a key event in the pathology of skin sensitisation (C. Johnson et al., 2020).

An unambiguous algorithm: The rationale behind this principle is to provide transparency and reproducibility. Therefore, it needs to be stated which data were used, in which way were the molecules represented (molecular descriptors) and what algorithm was used to construct the model.

A defined domain of applicability: As introduced above, an AD refers to the part of chemical space in which the model is capable of making reliable predictions. For a prediction to be applicable in a regulatory context, the test compound needs to be within the AD of the QSAR model.

Appropriate measures of goodness-of-fit, robustness and predictivity: An estimate of the predictive performance of the QSAR model needs to be provided. Goodness-of-fit refers to the capability of a model to fit training data. Robustness refers to the sensitivity of model parameters and predictions to small changes in the training data. A common way to evaluate robustness is by evaluating model performance in a cross-validation scheme (which corresponds to varied training data). Predictivity refers to an external validation. That is, evaluating how well the model predicts activities for compounds not used for model training or internal model validation.

A mechanistic interpretation, if possible: It should be attempted to find a mechanistic association of the descriptors used in the model and the predicted toxicity. A reasonable mechanistic interpretation will increase the credibility of the model.

While traditionally toxicity has been evaluated using animal studies, the use of *in silico* approaches including QSAR models has started to be recognised in regulatory contexts. A currently accepted application of QSAR models for regulatory decision making is the assessment of mutagenic impurities in pharmaceuticals. In this case, the combination of an expert system (introduced in Chapter 2) and a QSAR model may be used to replace an Ames test (Amberg et al., 2016). *In silico* approaches are considered as relevant to support toxicity assessment (at least as additional evidence to *in vivo* and *in vitro* studies) in various domains including industrial chemicals (ECHA, 2017), pesticides (for metabolites and degradates) (JRC, 2010), cosmetics (Gellatly & Sewell, 2019) and food (Hardy et al., 2017). Current efforts aim to further increase the acceptability of *in silico* models (including QSAR model) to evaluate toxicity as part of integrated strategies which combine experimental and *in silico*

findings (Bassan et al., 2021; Hasselgren et al., 2019; C. Johnson et al., 2020; Myatt et al., 2018). For some toxicity endpoints such as hepatotoxicity, the biological complexity of involved processes poses a big challenge on predictivity (Sistare et al., 2016). It can be stated that QSAR models have the potential to play a bigger role in regulatory toxicology moving forward, yet more work is needed to demonstrate their validity.

4.8 Use of neural networks in QSAR modelling

Neural networks, as introduced in Chapter 3, have long been used in chemistry applications including QSAR modelling (Gasteiger & Zupan, 1993). Due to less computational power being available, early approaches were restricted to a low number of input features (chemical descriptors) and a single hidden layer with a low number of neurons. Such shallow neural networks were, for instance, used to predict neurotoxicity of insecticides (Zakarya et al., 1997) or acute aquatic toxicity (Basak et al., 2000). With computational power increasing over time, deep learning emerged as a novel tool with applications in many domains such as image classification, speech recognition and language translation (LeCun et al., 2015). In deep learning, DNNs consisting of multiple hidden layers and potentially large numbers of neurons per layer are trained. DNNs have also been used for QSAR modelling and gained large attention due to their successful use in QSAR modelling competitions using bioactivity and toxicity assay data (Ma et al., 2015; Mayr et al., 2016). The success of DNNs was, at least partly, attributed to their capability of being used as multi-task models. Multi-task approaches for QSAR modelling are separately reviewed in Chapter 5. Compared to shallow neural networks, DNNs possess a much higher number of tuneable parameters. Hence, it can be expected that much larger amounts of data will be required to train them without overfitting.

The DNNs used in the studies described above all have in common that they used traditional chemical descriptors (e.g. chemical fingerprints) as input to the model (feedforward neural networks). A key characteristic of DNNs is their capability to learn suitable representations from raw input data (e.g. image pixels). This means that QSAR models may be learned without the use of traditional chemical descriptors. Instead compounds may be represented as SMILES strings (Gini et al., 2019), chemical graphs (Yang et al., 2019) or images of their 2D structure (Fernandez et al., 2018) and the obtained QSAR models may achieve performances comparable or, in some cases, even superior the models trained on traditional chemical descriptors. This is possible as the respective architectures (see below) are composed of differentiable operations which enables end-to-end learning (i.e. from raw input to toxicity).

DNN architectures utilised to learn from chemical images are CNNs (Goh et al., 2017). These contain convolutional layers in which meaningful features are extracted by compressing signals of spatially close pixels of an image. Then, the feature maps obtained after one or multiple convolutional layers are flattened to vectors which are ultimately used to predict the toxicity as done in a feedforward neural network.

DNNs used to learn from chemical graphs use GCNs (Rittig et al., 2022). Atoms are represented as vertices and bonds as edges of graphs. Vertices and edges are initially featurised using simple information about atoms (such as atom type, part of ring, hybridisation, charge) and bonds (bond type, conjugated or not), respectively. In a graph convolutional layer, the initial representations of atoms and/or bonds are updated by combining them with information about neighbouring atoms and bonds. Atom representations may be influenced by distant atoms and bonds if multiple graph-convolutional layers are applied. Similar to CNNs, the representations for individual atoms and bonds are finally combined into a vector representing the whole compound before predicting the toxicity.

DNNs based on SMILES strings as input typically use RNNs as the architecture. These architectures embed SMILES strings of variable length into a vector representation of fixed length before using that vector representation to predict toxicity (Gini et al., 2019). RNNs operate on sequential data by using the hidden representation of the previous SMILES character as additional input to the current SMILES character. Simple RNNs are not well capable of remembering input across long sequences and more sophisticated architectures such as Long Short-Term Memory (LSTM) have been introduced in order to overcome this limitation. LSTM introduce gates for input ('how to update the hidden state using the input?'), output ('how to use the hidden state to obtain an output?') and forget ('what part of the hidden state should be removed?') in order to control the behaviour of the hidden state at each step of a sequence (Olah, 2015). The hidden state obtained after the final character is used as vector to represent a compound in order to make a prediction.

DNNs are now used extensively for QSAR modelling and they generally perform very well compared to other ML approaches. However, their complex structure (i.e. composition of numerous non-linear functions) makes it difficult to understand how these models make predictions. This is why DNNs have been referred to as black box models (Loyola-Gonzalez, 2019). Interpretability of QSAR models is an important issue, especially when they are used to make high-stakes decisions. Being able to understand why a model made a particular decision increases the confidence in the predictions and thus the acceptance of the model tremendously. This point is also reflected in the fifth OECD principle for the use of QSAR models in context of regulatory decision making which states that models should be interpretable (see above). Furthermore, model interpretability may support a chemist to optimise

chemical structure by modifying the parts of the structure responsible for toxicity. Attempts to interpret DNN models for QSAR modelling are summarised in Chapter 7.

4.9 Conclusion

In this chapter, QSAR models have been introduced as a method to predict the toxicity of chemicals. To train a QSAR model, a suitable toxicity dataset linking measured toxicities to chemical compounds needs to be found. Next, chemical structures need to be curated and chemical descriptors need to be calculated. QSAR models are then trained and validated as for other ML models (see Chapter 3). For a QSAR model to be useful, it should be predictive and predictions should be interpretable. Both aspects are addressed in this thesis. In Chapters 5 and 6, multi-task and imputation models are compared to single task QSAR models with respect to their performance. Moreover, attempts are made to rationalise observed differences in performance. In the Chapters 8 to 11, a novel method to interpret neural network models for QSAR modelling is developed and tested.

Chapter 5 Multi-task and imputation modelling for toxicity prediction

5.1 Introduction

This chapter investigates the use of single task, multi-task and imputation models to predict in vitro toxicity data. The first section of the introduction will describe the differences between these modelling approaches. Subsequently, the use of these techniques for toxicity prediction in the literature will be summarised and the objectives of this study will be presented.

5.1.1 Multi-task and imputation models

Figure 5-1 provides a visual comparison of single task, multi-task and imputation models.

Single task modelling conceptually represents the simplest case of QSAR modelling. A separate model is trained for each task (i.e. toxicity assay), as indicated by an individual arrow for the predictions for each toxicity assay. Each learned model represents a mapping of a set of chemical descriptors for a compound to a predicted outcome for a certain toxicity assay. To make predictions for new compounds, a vector of chemical descriptors is used as input to the model.

In multi-task modelling, a single model is trained to predict several tasks at the same time. Hence, the model represents a mapping of a set of chemical descriptors for a compound to a predicted outcome for each of the modelled toxicity assays. This mapping is learned by using the training compounds' chemical descriptors and toxicity labels for each assay as input. As for single task models, the predictions for new compounds are made by inputting the chemical descriptors characterising the compounds.

In general, imputation means techniques that fill gaps in a sparse dataset (Horton & Kleinmann, 2007). A sparse dataset in the context of toxicity data is obtained if not every compound was tested in each of the toxicity assays in the dataset. Imputation may be done in a very simple manner such as taking the mean value for a given assay or single task models as described above may be used. More sophisticated approaches may adopt a multi-task approach to imputation and hence make predictions for several assays at the same time. In contrast to traditional multi-task models, such imputation

models can make use of both chemical descriptors and an (incomplete) set of experimentally measured data to predict the outcome of remaining assays for test compounds.

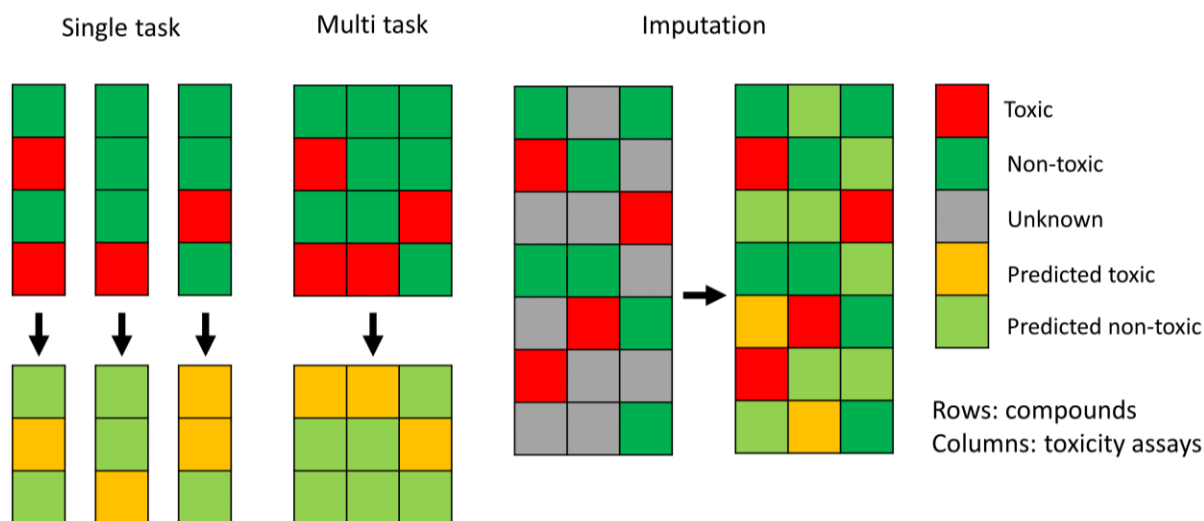


Figure 5-1 Schematic comparison of single task, multi-task and imputation models. Single task: A separate model is trained for each toxicity assay, represented by separated columns and arrows. Multi-task: A joint model is trained for all toxicity assays, indicated by the grouped columns and a single arrow. (Multi-task) Imputation: The model predicts the gaps in the sparse dataset. Unlike for traditional single task and multi-task models, there is no distinction between training and test compounds. All of the known labels may be used to train the model. Hence, when the model predicts the label for a certain cell of the matrix, it may have been trained on the labels of other toxicity assays for this compound.

5.1.2 Multi-task models in QSAR modelling

An early study on the application of multi-task models for QSAR modelling was conducted by Varnek et al (Varnek et al., 2009). They investigated two different approaches, which are multi-task neural networks and Feature Nets composed of neural networks. In Feature Nets, to predict property A, the prediction of property B is used explicitly as an additional feature. The tasks considered in this study were tissue-air partition coefficients for different tissues. They found that both multi-task neural networks and Feature Nets improved the performance over single task neural networks in most of the cases. In another study an approach comparable to Feature Nets (i.e. using predicted bioactivity profiles as additional inputs to a QSAR model) was tested in conjunction with conformal prediction (a method to estimate uncertainty of predictions) on several bioactivity and cytotoxicity datasets and they also found improved performance compared to traditional single task models (Norinder et al., 2020).

Multi-task modelling gained a lot of attention in the context of Deep Learning for QSAR modelling. In both the Tox21 challenge and the Kaggle challenge organised by Merck, the winning approaches were based on multi-task DNNs (Ma et al., 2015; Mayr et al., 2016), highlighting the potential of multi-task approaches. The Merck Challenge consisted of predicting numerical bioactivities for 15 datasets of

various sizes. In the work by Ma et al., a direct comparison of single task and multi-task DNNs was made across a range of different hyperparameter settings using R^2 as the metric. Multi-task DNNs were found to outperform single task DNNs overall, but not for the two largest of the 15 datasets. The Tox21 Challenge used 12 in vitro toxicity datasets containing around 10,000 compounds and the task was to classify the compounds as active or inactive. Mayr et al. compared single task with multi-task DNNs and found a higher ROC-AUC for multi-task DNNs for nine out of 12 datasets. However, neither of the two studies provided an explanation for the superiority of multi-task approaches.

An investigation to further understand the success of multi-task models was conducted on the Merck dataset in a publication by a Merck team (Xu et al., 2017). They trained pairwise DNNs and compared their performance to single task DNNs and multi-task DNNs including all tasks. In addition, they trained each model with 20 different random initialisations to enable a robust comparison between models. They reported that a given task (target task) in a multi-task model borrows signal from structurally similar molecules in other tasks (assistant task). If the activities of similar molecules are correlated (either positively or negatively) with those in the target task, the performance is increased. If the activities of similar molecules are uncorrelated, the performance may be decreased. If no data for structurally similar tasks is available in assistant tasks, then the performance is not influenced. This explanation is only satisfactory for sparse datasets with little overlap of compounds between different assays, as found in the Merck dataset.

In another study, the effect of sparsity on the performance of two different multi-task techniques (multi-task DNNs and Macau) was investigated (de la Vega de León et al., 2018). The datasets under study were a kinase dataset and a diverse dataset extracted from PubChem and the amount of sparsity was controlled by progressively removing data labels from the complete datasets. As expected, the performance decreased monotonically with increasing sparsity in the datasets. However, the decrease was quite small at first and accelerated as more labels were removed. The observed trends were remarkably similar for both multi-task techniques.

Since the arrival of Deep Learning, multi-task modelling has become a popular method for QSAR modelling, that seems to provide a benefit over single task modelling in many, but not all cases. The previously mentioned studies generated some insight into how the relations between tasks and the sparsity of the dataset affect the performance of multi-task models. However, more work is needed for a more complete understanding of the potential and limitations of multi-task QSAR models.

5.1.3 Imputation models in QSAR modelling

The profile-QSAR (pQSAR) model developed by Novartis represents an imputation approach suitable for bioactivity prediction on a large scale (Martin et al., 2019). The latest version of this approach is called All-Assay-Max2 pQSAR, but for simplicity the approach will be referred to as pQSAR. The pQSAR model consists of a two-step procedure. In the first step, a RF model is trained for each assay based on Morgan fingerprints as features and the available assay labels. These models are used to fill the gaps in the sparse dataset. In the second step, a PLS (partial least squares) model is trained for each assay on the profile of the remaining assays as obtained in the first step. To reduce the number of features, only assays related to the target assay (Pearson correlation >0.2 or >0.05) are included. The pQSAR model clearly outperformed RF models (a single model for each assay) both on a proprietary Novartis dataset (11805 assays) and a public ChEMBL dataset (4276 assays). For the Novartis dataset, pQSAR achieved a median R^2 (across all the assays) of 0.53 compared to 0.05 achieved by RF.

Alchemite is an imputation method based on neural networks developed in a collaboration by Optibrium and Intellegens (Irwin, Levell, et al., 2020; Whitehead et al., 2019). The neural network uses chemical descriptors as well as the bioactivities of remaining assays as input, followed by a single hidden layer to predict each assay. Missing input data is initially replaced by the mean value for the respective assay and predictions made by the network are used to update the missing values iteratively. This procedure is repeated until no further improvement in the predictions is observed. On a kinase dataset, the Alchemite method clearly outperformed RF models, a collective matrix factorisation method and a multi-task DNN. It achieved approximately the same performance as pQSAR 2.0, a precursor of the above described version of the pQSAR method. A notable feature of Alchemite is its capability to express confidence in single predictions. This is achieved by training the network with several random initialisations and use the standard deviation across different runs of the model as a measure of uncertainty. The confidence in the predictions is correlated with the accuracy and the performance can be increased by only keeping the most confident predictions.

Both of these recent imputation models clearly outperformed single task and standard multi-task models. The benefit in performance seems to come from the relations between different assays and the models' capability to leverage these patterns. However, it is unclear what characteristics of a dataset cause this behaviour and hence under which circumstances imputation approaches will be particularly effective.

5.1.4 Objectives

QSAR modelling is a well-established approach to predict the toxicity of chemicals. A range of different toxicity endpoints must be considered to evaluate the safety of a chemical. Predictions for the different endpoints can be made by training single task QSAR models for each endpoint or by training a joint multi-task model for all endpoints. Imputation modelling is possible if predictions are to be made for compounds that were already tested for some of the endpoints of relevance. In the present study, a rigorous comparison of single task, multi-task and imputation models is conducted based on performance measured on two multi target in vitro toxicity datasets and the results are compared to findings of previous studies. Furthermore, attempts are made to understand and explain differences in performance that occur between the different methods. These insights might be useful to anticipate for which types of datasets (relation between assays, size, sparsity) certain approaches might be particularly suitable. After careful analysis of the findings, recommendations are given on strategies to predict toxicities of multiple toxicity endpoints.

5.2 Methodology

5.2.1 Datasets

Two different in vitro toxicity datasets were used for the studies regarding single task, multi-task and imputation modelling: firstly, the ISSSTY (Istituto Superiore di Sanità Salmonella Typhimurium) database containing data for the Ames test for mutagenicity, generated by the Italian National Institute of Health; and secondly, the Tox21 dataset, generated by the Tox21 consortium.

The ISSSTY database (*ISSTOX Chemical Toxicity Databases*, 2021) (Table 5-1) contains data for six different bacteria strains, designed to detect different mechanisms of mutagenicity, such as substitutions of DNA bases or deletions and insertions leading to a frameshift in the triplet code of the DNA. The Ames test is a well-established in vitro test for mutagenicity used in regulatory contexts (OECD, 1997). Each of the six strains was tested with or without the addition of S9 mix to mimic metabolism of higher organisms, which leads to a dataset of 12 different toxicity assays. The outcome for a compound in each of the assays is either 'active', 'inactive' or 'equivocal' (Benigni et al., 2013), due to the fact that the database may contain repetitions of the same experiment. A compound was considered 'active' if it was active in more than 60% of repeated experiments and 'inactive' if it was

active in less than 40% of the experiments. Otherwise, the results were considered ‘equivocal’. To obtain a binary dataset for this study, ‘equivocal’ entries were removed from the data table.

The Tox21 dataset (*Tox21 Challenge Dataset*, 2014) (Table 5-2) consists of the three separate datasets ‘training’, ‘testing’ and ‘final evaluation’, as used in the Tox21 QSAR modelling challenge. For this study the three datasets were combined to a single dataset. The dataset contains binary data (active or inactive) for 12 toxicity assays. Seven of them measure the activation of various nuclear receptors related to toxic effects, while the remaining five measure the activation of cellular stress pathways. Assays like these are typically conducted in a preclinical toxicity screening of chemicals (Huang et al., 2016).

In both datasets, not every compound was measured in every assay resulting in sparse (i.e. incomplete) datasets. The Ames dataset is 40.5% complete and the Tox21 dataset is 83.4% complete. The number of compounds measured in each assay can be seen in Table 5-1 and Table 5-2. The numbers refer to the datasets after the steps described in the section 5.2.2 (Data Processing) were performed.

Table 5-1 The Ames dataset. Assay function describes which types of DNA mutations are detected in the respective assay. The numbers are those after the data processing steps described in ‘Data Processing’. Each row contains the total number of unique molecules and the proportion of active labels for the given assay. The ‘overall’ row describes the total number of compounds and the proportion of active labels among all the labels.

Assay name	Assay function (Hamel et al., 2016)	Number molecules	Proportion actives
TA100	Substitutions	4627	0.259
TA100_S9	same as TA100 for metabolites	4350	0.318
TA102	substitution, small deletions, cross-linking and oxidations	880	0.173
TA102_S9	same as TA102 for metabolites	763	0.213
TA1535	Substitutions	2489	0.114
TA1535_S9	same as TA1535 for metabolites	2347	0.119
TA1537	frameshifts, intercalation	2081	0.118
TA1537_S9	same as TA1537 for metabolites	1998	0.120
TA97	frameshift, intercalation	1049	0.140
TA97_S9	same as TA97 for metabolites	1010	0.168
TA98	Frameshifts	4345	0.235
TA98_S9	same as TA98 for metabolites	4055	0.285
overall	-	6168	0.213

Table 5-2 The Tox21 dataset. Assay function describes the major functions of the tested receptor or pathway. The numbers are those after the data processing steps described in 'Data Processing'. Each row contains the total number of unique molecules and the proportion of active labels for the given assay. The 'overall' row describes the total number of compounds and the proportion of active labels among all the labels. NR: nuclear receptor, SR: stress response, AhR: aryl hydrocarbon receptor, AR: androgen receptor, AR-LBD: androgen receptor ligand-binding domain, ARE: antioxidant response element, ATAD5: ATPase family AAA domain containing 5, ER: estrogen receptor, ER-LBD: estrogen receptor ligand-binding domain, PPAR-gamma: peroxisome proliferator-activated receptor gamma, HSE: heat shock element, MMP: mitochondrial membrane potential

Assay name	Assay function	Number molecules	Proportion actives
NR-AhR	Regulation of xenobiotic metabolism, immunity, cell differentiation (Kawajiri & Fujii-Kuriyama, 2017)	6810	0.115
NR-AR	Development of the male reproductive system (Matsumoto et al., 2013)	7460	0.034
NR-AR-LBD	Same as NR-AR, different test system	6991	0.030
NR-Aromatase	Enzyme required for estrogen synthesis (Simpson et al., 2002)	6009	0.049
NR-ER	Development of the female reproductive system (Muramatsu & Inoue, 2000)	6367	0.105
NR-ER-LBD	Same as NR-ER, different test system	7199	0.041
NR-PPAR-gamma	Regulation of glucose and lipid metabolism (Janani & Kumari, 2015)	6752	0.029
SR-ARE	Response to cellular oxidative stress (T. Nguyen et al., 2009)	6121	0.156
SR-ATAD5	Genome replication (Park et al., 2019)	7326	0.040
SR-HSE	Prevention of protein misfolding (Balchin et al., 2016)	6794	0.047
SR-MMP	Parameter for mitochondrial toxicity (Sakamuru et al., 2012)	6074	0.151
SR-p53	Recognition of DNA damage, regulation of DNA repair (May & May, 1999)	7049	0.062
overall	-	8090	0.069

5.2.2 Data processing

Chemical structures were standardised using the Python packages RDKit (*RDKit: Open-Source Cheminformatics*, n.d.) (2019.09.03) and MolVS (Swain, 2016) (0.1.1). Firstly, SMILES strings from which no valid molecules could be generated were discarded. Then, bonds to metal atoms were

disconnected and the charge was removed where possible. In the next step, inorganic fragments and solvents were removed. Next, certain chemotypes (e.g. nitro groups) and tautomers were transformed into a canonical form. Subsequently, SMILES were converted to standard InChI (Heller et al., 2015) and back to SMILES so that two molecules, which share the same InChI, are guaranteed to be represented by the same SMILES. Finally, mixtures of different organic components were discarded.

At this point, the datasets contained molecules represented by identical SMILES strings. The results for identical SMILES were aggregated to obtain a single set of data labels for each unique SMILES in the dataset. This was done by keeping the majority label (active or inactive) for each assay for identical SMILES. No data label (i.e. a data gap) was assigned to a SMILES-assay pair if active and inactive labels were of equal number. As molecular descriptors, RDKit's Morgan fingerprints (radius=2, hashed to 2048 bits) were calculated, which are comparable to Extended Connectivity Fingerprints (ECFP4) (Rogers & Hahn, 2010).

5.2.3 Data splitting

Each dataset was split into a training set and a test set. The training sets were used to optimise hyperparameters and train the final models, and the test sets were used to evaluate the performance of the final models. Two different splitting methods were employed: compound-based and assay-based splits.

The compound-based splits (Figure 5-2: left) were used to compare the performance of traditional multi-task models with single task models. 20% of the compounds in each dataset (Ames or Tox21) were selected at random for the test set and the remaining 80% formed the training set. This means that the training and test sets are the same for all the assays in each dataset.

Compound-based splits represent the established way to assess the performance of QSAR models and is the approach used, for example, in the Tox21 QSAR modelling challenge (Mayr et al., 2016). However, the imputation techniques described above, namely Alchemite and pQSAR, have been evaluated primarily in a different scenario, appropriate for sparse activity matrices. In this scenario, the imputation model is created using training data and applied to fill the missing values in the activity matrix. We implemented this testing scenario with what we call the assay-based splits (Figure 5-2: right). Each assay was considered independently and 20% of the compounds for which the toxicity labels are known were selected at random and placed in the test set with the remaining compounds being added to the training set. Some compounds therefore appear in both the training data and the

test data, however, the assay labels are split so that none of the same compound/assay label pairs appear in both the training and the test data. Thus, a compound may be in the test set for assay A, but in the training set for assay B with the toxicity label of the compound from assay B given as input to the imputation model.

Compound-based split

	A1	A2	A3	A4
C1	1	-	-	0
C2	-	1	-	1
C3	0	1	0	1
C4	0	0	-	0
C5	1	-	1	0
C6	1	-	0	0
C7	1	0	0	1
C8	0	1	-	0
C9	-	0	-	0
C10	0	1	1	1

Assay-based split

	A1	A2	A3	A4
C1	1	-	-	0
C2	-	1	-	1
C3	0	1	0	1
C4	0	0	-	0
C5	1	-	1	0
C6	1	-	0	0
C7	1	0	0	1
C8	0	1	-	0
C9	-	0	-	0
C10	0	1	1	1



	Train label	0: non-toxic
	Test label	1: toxic

Figure 5-2 Comparison of splitting strategies. For the compound-based split 20% of the compounds are randomly put in the test set. These compounds are used for testing across all the assays. For the assay-based split, 20% of the compounds per assay are randomly put in the test set. This leads to a different set of test compounds for each assay. A: assay. C: compound

5.2.4 Modelling

In the following, the different techniques used for single task, multi-task and imputation modelling are briefly described, followed by a description of the model training.

5.2.4.1 Single task models

A: Random forest

Random forest (RF) is a well-established algorithm for QSAR modelling that has achieved good performance in a variety of tasks (Svetnik et al., 2003). RF is an ensemble method, where different decision trees are trained on different bootstrapped subsets of the training data and different random subsets of features. For this work, the implementation of RF in the Python scikit-learn (Pedregosa et al., 2011) package (0.22.1) was used. The hyperparameters considered for optimisation in the grid search are given in Table 5-3.

Table 5-3 Random forest hyperparameters considered in optimisation.

Hyperparameter	Values	Description
<i>n_estimators</i>	100, 500	number of trees in the Random forest model
<i>class_weight</i>	<i>None, balanced, balanced_subsample</i>	assigns weights to instances of different classes <i>Balanced</i> : instances of the minority class receive a weight equal to the inverse of the class frequency in the whole input data <i>Balanced subsample</i> : the frequencies of classes for each respective tree (with its bootstrap sample) are considered to assign weights
<i>max_features</i>	log2, 0.25	controls the number of randomly selected features that are considered to find the best split <u>log2</u> : logarithm of base 2 of all input features <u>0.25</u> : fraction of all input features

B: XGBoost

Another ensemble technique is so-called boosting, which generally refers to the sequential combination of several weak (i.e. perform slightly better than random) learners in order to learn from the mistakes of previous learners (Meir & Rätsch, 2003). In gradient tree boosting, decision trees are learned sequentially to predict the residuals (i.e. mistakes) of the previous tree, and eventually the predictions of all single trees are combined (Friedman, 2001). XGBoost (XGB) is a popular open-source implementation of gradient tree boosting which scales well to very large datasets (Chen & Guestrin, 2016). The Python package XGBoost (1.0.1) provides an implementation of gradient tree boosting optimised for scalability. The hyperparameters considered for optimisation in the grid search are given in Table 5-4.

Table 5-4 XGBoost hyperparameters considered in optimisation.

Hyperparameter	Values	Description
<i>Num_round</i>	300, 700	sets the number of sequential trees that are trained
<i>eta</i>	0.1, 0.3, 0.5	learning rate that shrinks the size of the update of predictions after each boosting step
<i>Colsample_bytree</i>	0.5, 0.7	randomly selected proportion of features used within each individual tree
<i>Alpha</i>	0, 1	strength of L1 regularisation of the weights per leaf
<i>lambda</i>	1, 10	strength of L2 regularisation of the weights per leaf
<i>Scale_pos_weight</i>	1, weighted	assigns weights to instances of different classes <u>weighted</u> : instances of the minority class receive a weight equal to the inverse of the class frequency in the whole input data

C: Deep neural network

Deep neural networks (DNN) are combinations of artificial neurons organised in layers (LeCun et al., 2015). Each layer can be considered a simple non-linear machine learning model, but they can become increasingly complex by adding additional layers. The DNNs used in this project are feedforward neural networks obtained using the Python package Tensorflow (Abadi et al., 2016) (2.1.0) with the Keras API (Chollet & others, 2015). 2048 nodes were used in the input layer (one node for each bit of the chemical fingerprint) and the output layer consisted of a single node to which the sigmoid function was applied to obtain a binary output for classification (using 0.5 as the classification threshold). The DNNs for different assays contain between one and three hidden layers (number of hidden layers is a hyperparameter considered for optimisation). The ReLU activation function was applied to all nodes in the hidden layers. Binary cross-entropy was used as the loss function and the Adam algorithm was used for fitting the network's weights and biases. The hyperparameters considered for optimisation are given in Table 5-5.

Table 5-5 Deep neural network hyperparameters considered in optimisation.

Hyperparameter	Values	Description
Hidden layers	1, 2, 3	Number of hidden layers in the network
Nodes per hidden layer	1024, 2048	Number of nodes in each hidden layer
Learning rate	0.0003, 0.001, 0.003	size of the updates to the network parameters per training step
Dropout	0, 0.2	Amount of randomly selected nodes that are ignored in each training step
<i>L2 regularisation</i>	0.0001, 0.001	Strength of L2 regularisation on the weights per node
Batch size	10, 50	Size of batches used for each update of weights with the Adam optimiser
Number of epochs	3, 5, 10	Number of training runs through every data point.
Class weight	1, weighted	assigns weights to instances of different classes <u>weighted</u> : instances of the minority class receive a weight equal to the inverse of the class frequency in the whole input data

5.2.4.2 Multi-task and imputation models

For both traditional multi-task and multi-task imputation models the same algorithms (Feature Net, multi-task DNN and Macau) were used. As described above, the imputation models were used with the assay-based splits, while the traditional multi-task models were used with the compound-based splits. This leads to the fundamental difference that imputation models may use experimentally determined toxicity labels of test compounds to make predictions.

A: Feature Net

The Feature Net (FN) (Davis & Stentz, 1995) technique combines multiple single task predictors into a net-like structure and can be used with any supervised machine learning algorithm. In the first step, a single task model was trained for each assay separately using the chemical features and the models were used to predict the unknown toxicity labels for the whole dataset. In the second step, the final model for each assay was obtained by retraining the single task model using the labels of the other assays explicitly as features, in addition to the chemical features. In this setting, each modelled assay is called the target assay and the other assays, used as features, are called auxiliary assays. For a given target assay, the second model is trained using compounds with experimental labels for this assay (i.e. no predicted labels as ground truth), however, the labels of the auxiliary assays are either experimentally determined labels, or the predicted labels from the models in the first step, when no experimentally determined label is available. A schematic depiction of the FN models is given in Figure

5-3. The FN models were trained using all of the single task methods described above using the same hyperparameters as for the single task models, that is, no additional optimisation was performed.

When a FN model is used for prediction, each assay value is predicted sequentially with the test compound input as chemical features and assay labels for all other assays, that is, the auxiliary labels. The auxiliary assay labels may be predicted values or experimental values. Two different scenarios that mimic how QSAR is used in practice were investigated. One common application is making predictions for virtual compounds, that is, compounds which have not yet been synthesized. In this case, any experimental values were discarded so that all the input labels for the test compounds were predicted values. This application was tested in the compound-based splits. The second application is the prediction of assay outcomes for compounds that have already been characterized for some toxicity endpoints. To simulate this case, we used experimental values for the auxiliary assays in the test set, when these were available, otherwise we used predicted values as determined in the first step of the FN method. This application was tested both in the compound-based splits (with experimental test labels) and in the assay-based splits.

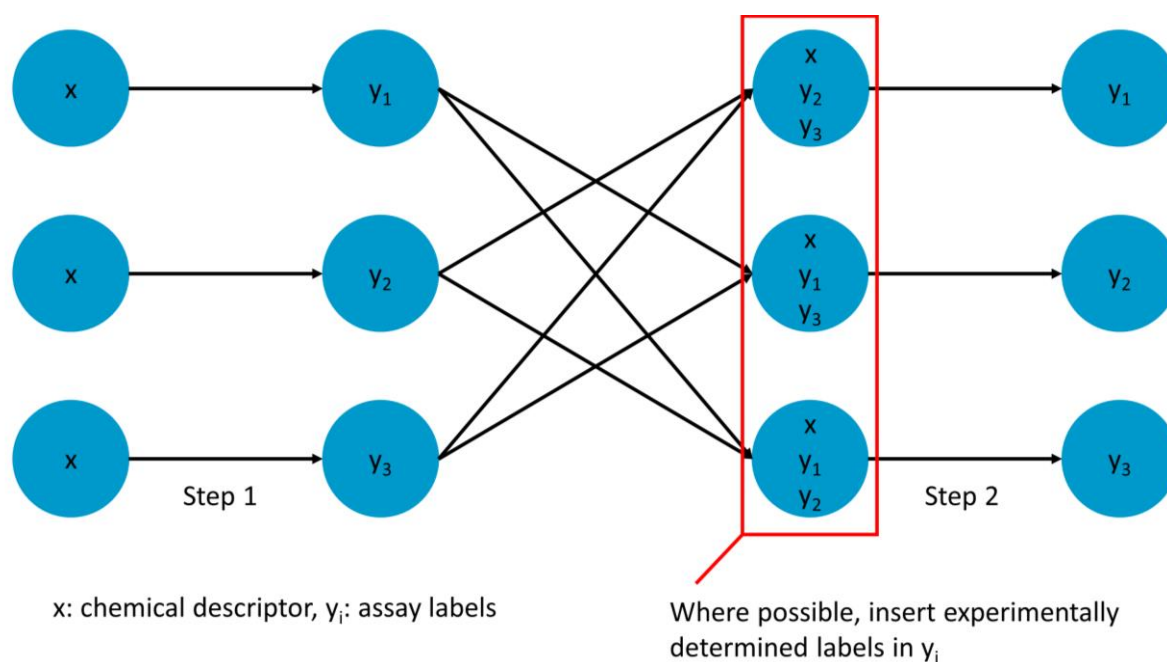


Figure 5-3 Schematic depiction of Feature Net models. In Step 1, a single task QSAR model is trained to predict the labels of test compounds for each toxicity assay in turn (y₁, y₂, y₃). In Step 2, the models are retrained using both chemical descriptors (x) and assay labels as features. The labels for the auxiliary assay are either experimentally determined assay labels or the predictions from Step 1 where experimental values are not available.

B: Multi-task Deep Neural Network

As for the single task DNN models, the multi-task DNN models were trained using the Python package Tensorflow with the Keras API. A multi-task DNN is trained to predict several assays at the same time

with the output layer of the network consisting of one node for each task. The sigmoid function was applied to each node in the output layer and the obtained output was binarised (using 0.5 as the classification threshold) to obtain a predicted class for each assay. The loss for a single data point (a single compound with up to 12 assay labels for Ames and Tox21 dataset) during training is its binary cross-entropy averaged across all assays. For partially complete data points (i.e. molecules for which the label is known only for some of the assays), the assays without available labels were excluded when the loss was computed. This enables training on sparse datasets. The DNN was trained using the 2048 chemical features as inputs, as for the single task models. When the multi-task model was used for prediction, a test compound was input as chemical features and the output was a vector of predicted values, one for each assay.

For the assay-based split, the multi-task DNN was trained on all compounds assigned to the training set (all of which had at least one assay label as shown in the assay-split in Figure 5-2) and, for a given compound, only the labels that were assigned to the training set (green cells of the matrix in Figure 5-2) were used to compute the loss during training.

The same hyperparameters and values were considered for optimisation as for the single task DNNs. The only exception is the parameter *class weight*. The considered options to weight the minority class across all tasks were: 1, 3, 5, 15. The best set of hyperparameters was selected by averaging the MCC scores for each set of parameters across all the assays.

C: Macau

Macau is a Bayesian probabilistic matrix factorization technique that is able to analyse sparse matrices. Probabilistic matrix factorization gained attention for recommender systems, following the 2009 Netflix competition where it was used to make predictions using the data matrix only, that is, the ratings for viewer-movie pairs. The Macau method has recently been used for multi-task modelling in QSAR (Simm et al., 2015) and is also able to use descriptors of the entities being analysed, which are referred to as side information (de la Vega de León et al., 2018). In the Macau models developed here, the sparse data matrix consists of compound-assay label pairs and Morgan fingerprints are used as side information for the compounds, with no side information being used for the assays. The hyperparameters considered for optimisation are given in Table 5-6. The best set of hyperparameters was selected by averaging the MCC scores across all assays for each set of parameters. The Python package Macau (version 0.5.2) was used in this study.

Table 5-6 Macau hyperparameters considered in optimisation.

Hyperparameter	Values	Description
<i>Num_latent</i>	16, 32, 64	Number of dimensions used for the latent space representation of entities
<i>nsamples</i>	800, 1600, 3200	Number of samples drawn from the Gibbs sampler
<i>burnin</i>	200, 400, 800	Number of burn-in samples from the Gibbs sampler that are discarded

Naturally, matrix factorisation techniques are used for imputation modelling. Since Macau can include side information about compounds, it can be used for conventional multi-task modelling. In that case, the latent representation of test compounds is fully determined by the fingerprint, as no toxicity labels for the test set are available to the model.

5.2.4.3 Model training

For each technique, a 5-fold cross validation grid search was performed on the training set for hyperparameter optimisation, except for Macau, where the grid search was performed using a single validation set containing 20% of the labels in the training set for each assay. This difference is due to the assay-based splits better reflecting the typical use of Macau and the complexity of setting up cross validation in this scenario. Furthermore, preliminary experimentation suggested that Macau is relatively insensitive to the specific hyperparameter settings. All of the models used Morgan fingerprints hashed to 2048 bits as chemical features, generated using the Python package RDKit.

5.2.5 Model evaluation

All the modelling techniques used in this study involve stochastic processes, e.g., the random initialization of weights in the neural networks or the generation of bootstrap samples in the RF models. The random behaviour of these methods can be made reproducible by setting a random seed, however, the result will represent only one from a distribution of potential results. For a rigorous comparison of the different techniques, all of the final models for the Ames and Tox21 datasets were trained using 20 different random seeds and the range of scores for the test set was examined. This follows the approach of Xu et al. in their comparison of multi-task DNNs with single task DNNs (Xu et al., 2017).

Matthews Correlation Coefficient (MCC) was chosen as the primary metric to evaluate model performance, due to its suitability for imbalanced datasets. In addition, the F1 scores and ROC-AUCs are reported in the Appendix (Tables A5-3 to A5-14). When reporting the performance of a technique across different assays and different random seeds, the mean of all assays for each seed was computed first, and the median across all seeds is reported.

5.2.6 Rationalisation of the imputation models' performances

After evaluating overall performances, the characteristics of the datasets were investigated in order to gain insights on when multi-task imputation is likely to lead to the most benefit. The different evaluation measures are described below.

5.2.6.1 Roles of chemical similarity

It is well established that the success of QSAR models depends on the chemical similarity between compounds for which predictions are made and those used to train the model (Sheridan et al., 2004). In particular, performance of the model tends to be high for very similar compounds and relatively poor for chemically dissimilar compounds. In fact, the concept of an applicability domain is used to determine regions of chemical space where a model makes reliable predictions (Mathea et al., 2016). An investigation of the extent to which the performance of imputation models is affected by chemical similarity is carried out to determine if these models may be particularly useful to overcome poor performances for single task QSAR models on chemically dissimilar compounds.

For this experiment the test sets as obtained in the assay-based splits were divided into three bins (0-0.4, 0.4-0.6, 0.6-1), depending on the chemical similarity of each test compound to the training compounds of the respective assay. Chemical similarity was evaluated as the average Tanimoto similarity of a test set compound to the five nearest neighbours in the training set based on Morgan fingerprints. Results were only obtained for assays with at least 100 test compounds in each of the three bins. The performance of each model was evaluated on each of the bins independently and, in each case, the results were averaged over the 20 models obtained using different random seeds.

5.2.6.2 Role of data sparsity

A key characteristic of imputation models is that toxicity data of related endpoints may be used for the prediction of a given toxicity endpoint. It was investigated how many labels of related toxicity endpoints are required for an imputation model to outperform single task QSAR models.

For this experiment, the test sets as obtained in the assay-based splits were divided into three bins, depending on the number of experimentally determined toxicity labels of the remaining assays that are in the training set for a given test compound. The bins were 0-1 labels, 2-3 labels and >3 labels. Results were only obtained for assays with at least 100 test compounds in each of the three bins. The performance of each model was evaluated on each of the bins independently and, in each case, the results were averaged over the 20 models obtained using different random seeds. This analysis was not possible for the Tox21 dataset, as this dataset contains very few compounds that were tested in only a few of the assays.

5.2.6.3 Pairwise and leave-one-assay-out Feature Net models

Two different approaches were taken to examine the importance of single assays for the overall success of FN models: pairwise FN models and leave-one-assay-out (LOAO) FN models.

For pairwise FN models, just one auxiliary assay was used as an additional feature. For each target assay, the remaining auxiliary assays were tested in turn in pairwise FN models and the scores of the models were compared to the score of a single task QSAR model for the target assay. A large improvement in performance would suggest a high importance of the respective auxiliary assay for the full FN model.

LOAO FN models, on the other hand, are FN models missing exactly one of the auxiliary assays, hence all but one of the remaining assays were used at a time as additional features and the performances were compared to those of the full FN models. In this approach, a high importance of an auxiliary assay would be indicated by a large decrease in performance, when this particular assay was left out.

As for the full FN models and the single task QSAR models, 20 different random seeds were used for each pairwise FN model and LOAO FN model and the median MCC score across the 20 models was used to represent each particular setting.

The importance of a single auxiliary assay to the overall success of the full FN model was also examined using information theory (Shannon, 1948). The entropy of an assay A is computed as:

$$H(A) = -(P_A \times \log_2 P_A + P_a \times \log_2 P_a)$$

where P_A is the probability of a randomly selected compound being active in the assay, computed by the ratio of compounds with the label 'toxic' to all labelled compounds, and P_a the proportion of non-toxic compounds, given by $1 - P_A$. The mutual information of two assays is computed by taking the sum of the entropies of each assay and subtracting the entropy of the joint distribution:

$$MI(A, B) = H(A) + H(B) - H(A, B)$$

The entropy of the joint distribution is computed as:

$$H(A, B) = -(P_{AB} \times \log_2 P_{AB} + P_{Ab} \times \log_2 P_{Ab} + P_{aB} \times \log_2 P_{aB} + P_{ab} \times \log_2 P_{ab})$$

Where P_{AB} is the proportion of compounds labelled as toxic in both assay A and B , P_{ab} is the proportion of compounds that are non-toxic in both assay A and assay B , and P_{Ab} and P_{aB} are the proportions of compounds toxic only in assay A or only in assay B , respectively. The proportions were only computed for compounds that are labelled for both assay A and assay B .

The relatedness between a target assay and an auxiliary assay was measured by the ratio of mutual information (MI) between two assays to the entropy of the target assay. This measure of relatedness, called here MI-entropy ratio, describes how much of the total entropy (or amount of information) of the target assay (here assay A) is contained in the auxiliary assay.

$$MI - entropy\ ratio(A, B) = \frac{MI(A, B)}{H(A)}$$

Observed differences of pairwise and LOAO FN models to the respective single task models were analysed with respect to the MI-entropy ratio of the appropriate assay pair.

5.2.7 Exploration of various classification thresholds for Macau models

Different classification thresholds for the XGB-FN and the Macau model on the p53 assay in the Tox21 dataset were investigated. The assay was selected as an example, as the Macau model performs particularly poorly according to MCC score (worse than single task models) on this assay. For simplicity, the analysis is based on predictions obtained from a single run (one random seed) for each model. For each model, the range of predicted probability scores was plotted and analysed. In particular, the number of actual toxic and non-toxic compounds that fell in different ranges of predicted scores were analysed as well as how different thresholds on classifications would affect the MCC score on the test set.

5.3 Results

Initially, traditional single task and multi-task QSAR models were compared using compound-based splits, i.e. the same compounds were placed into training and test set for all the assays.

5.3.1 Single task models

Single task QSAR models for the two datasets under study were generated using three different ML algorithms: RF, XGB, DNN. The performances of these models on the test set are reported in Figure 5-4 using the MCC score as a metric. As described in the Methodology, 20 independent runs using different random seeds were conducted for each model, which are summarized as box plots.

The performance between assays varied considerably. For instance, the MCC scores for TA98 were substantially higher than the ones for TA97, across all the algorithms (Figure 5-4A). This suggests that, for the available data points, some of the assays are inherently more difficult to predict than others. When comparing the scores of the models on a per assay basis, there were a few cases where one model clearly outperformed the other two, such as XGB for TA1535 (Figure 5-4A), or RF for NR-Aromatase (Figure 5-4B). However, in most cases the different ML algorithms yielded models of comparable performance. XGB provided the best median MCC score across the different random seeds across both datasets, albeit the differences were quite small (Table 5-7).

The range of scores obtained by an algorithm on a single assay was quite large in some cases (e.g. DNN for TA102_S9, Figure 5-4A) and quite small for others (e.g. RF for NR-ER-LBD, Figure 5-4B). The results demonstrate that the performance of the models is very sensitive to the random seed in some cases. This variance tended to be larger for the DNN models compared to the other ML algorithms. Additionally, for some assays the variance seemed to be consistently larger than for others. The affected assays seemed either to have a comparatively small number of data points (e.g. TA102, TA102_S9, TA97) or to be particularly imbalanced (e.g. NR-PPAR-gamma, NR-AR-LBD). By considering stochastic effects for an algorithm on a given assay, a more robust comparison between models is possible. In particular, it becomes less likely that differences observed between models are the result of chance.

The results of the single task models will be used as a benchmark for the multi-task and imputation models.

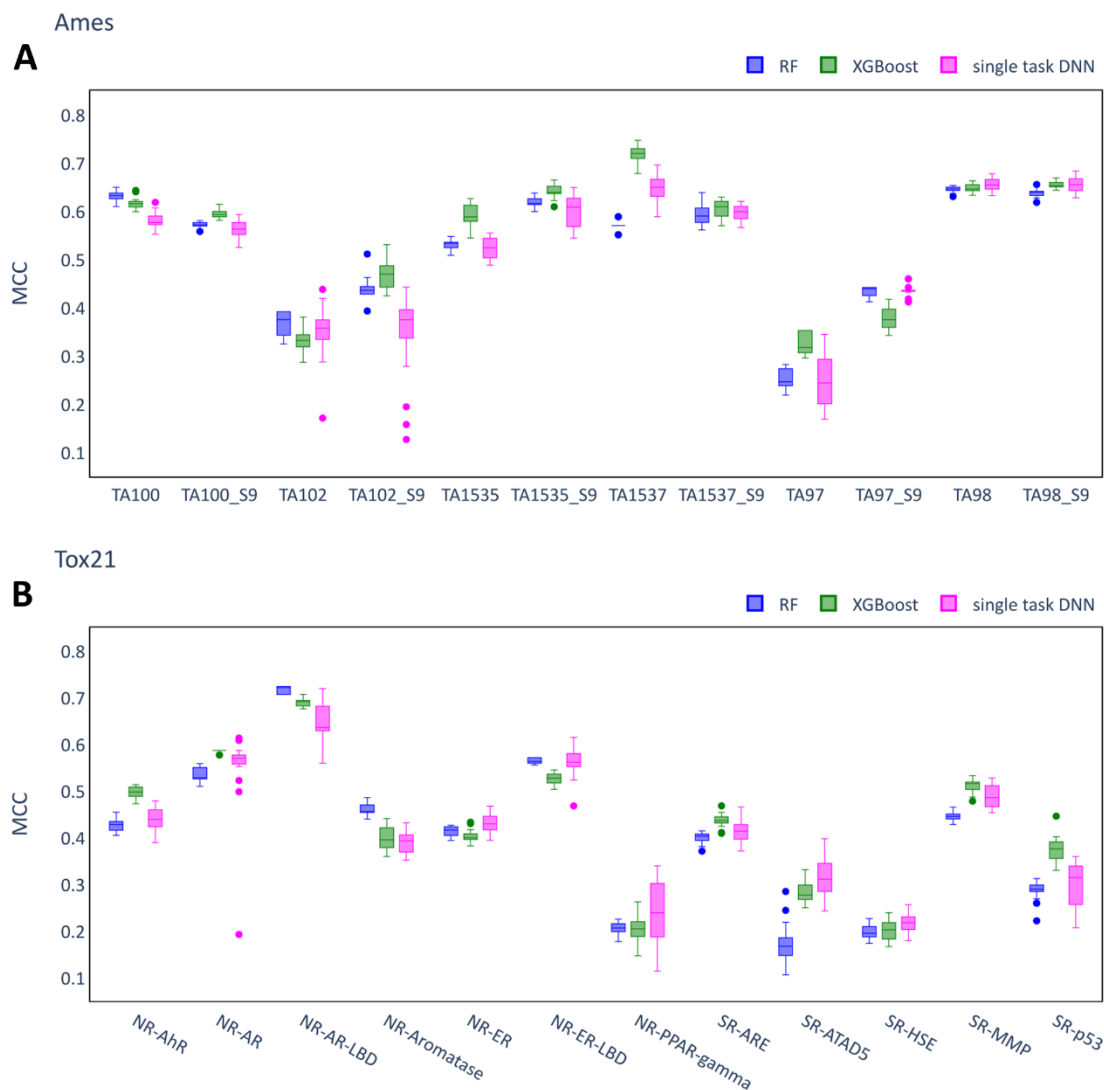


Figure 5-4 Performance of single task QSAR models. A: Ames dataset. **B:** Tox21 dataset. Each box summarises the MCC scores of 20 independent runs of the model on the test set with identical hyperparameters but different random seeds.

Table 5-7 Median MCC scores of single task QSAR models. Median MCC scores and interquartile range for each technique and dataset on the test set across the 20 random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model for each dataset is shown in bold.

	Ames	Tox21
RF	0.526 (0.523-0.527)	0.402 (0.400-0.405)
XGB	0.547 (0.545-0.550)	0.427 (0.422-0.430)
DNN	0.519 (0.508-0.527)	0.417 (0.410-0.426)

5.3.2 Traditional multi-task models

Multi-task models are trained on the labels of several tasks, in this case toxicity assays. Three different multi-task techniques were investigated: FN models (based on either RF, XGB or DNN models), multi-

task DNN and Macau. Figure 5-5 compares the performances of multi-task QSAR models to XGB as the best single task model. Among the FN models, only the best performing model on the test set (the one based on XGB) was included in Figure 5-5.

In a few cases, some of the multi-task models outperformed the XGB models (e.g. TA97 in Figure 5-5A or NR-ER-LBD in Figure 5-5B), but in others the multi-task models were outperformed by the XGB models (e.g. TA1537 in Figure 5-5A or SR-p53 in Figure 5-5B). Therefore, multi-task models were beneficial for some assays, but they did not outperform single task models consistently in these datasets. The mean MCC scores for the different techniques are reported in Table 5-8. MCC, F1 and ROC-AUC scores for all assays are reported in Appendix B (Tables B3 to B8)

None of the multi-task techniques achieved a higher median MCC score than the single task XGB models on the Ames dataset. In contrast, both the XGB-FN and the multi-task DNN yielded higher median MCC scores than XGB on the Tox21 dataset, albeit only by small margins. The Macau technique was outperformed by the other methods on both datasets. The difference in performance between the Macau model and the other techniques was comparatively small on the Ames dataset, but quite large on the Tox21 dataset. In fact, for two of the assays of the Tox21 dataset Macau achieved median MCC scores of zero, which indicates a performance not better than random guessing. A MCC score of zero may be achieved if no instances in the test set are predicted as positive or none of the predicted positives are actual positives (technically the MCC score is not defined in these cases due to a division by zero, however the scikit-learn implementation returns a score of zero).

Similar to the single task model results, the variance between different runs (with different random seeds) was quite large in some cases. Among the multi-task techniques included in Figure 5-5, multi-task DNN showed the largest variances. In addition, it appears that the same assays as for the single task models were prone to large variances across the different algorithms, such as TA102 (Figure 5-8A) and NR-PPAR-gamma (Figure 5-5B).

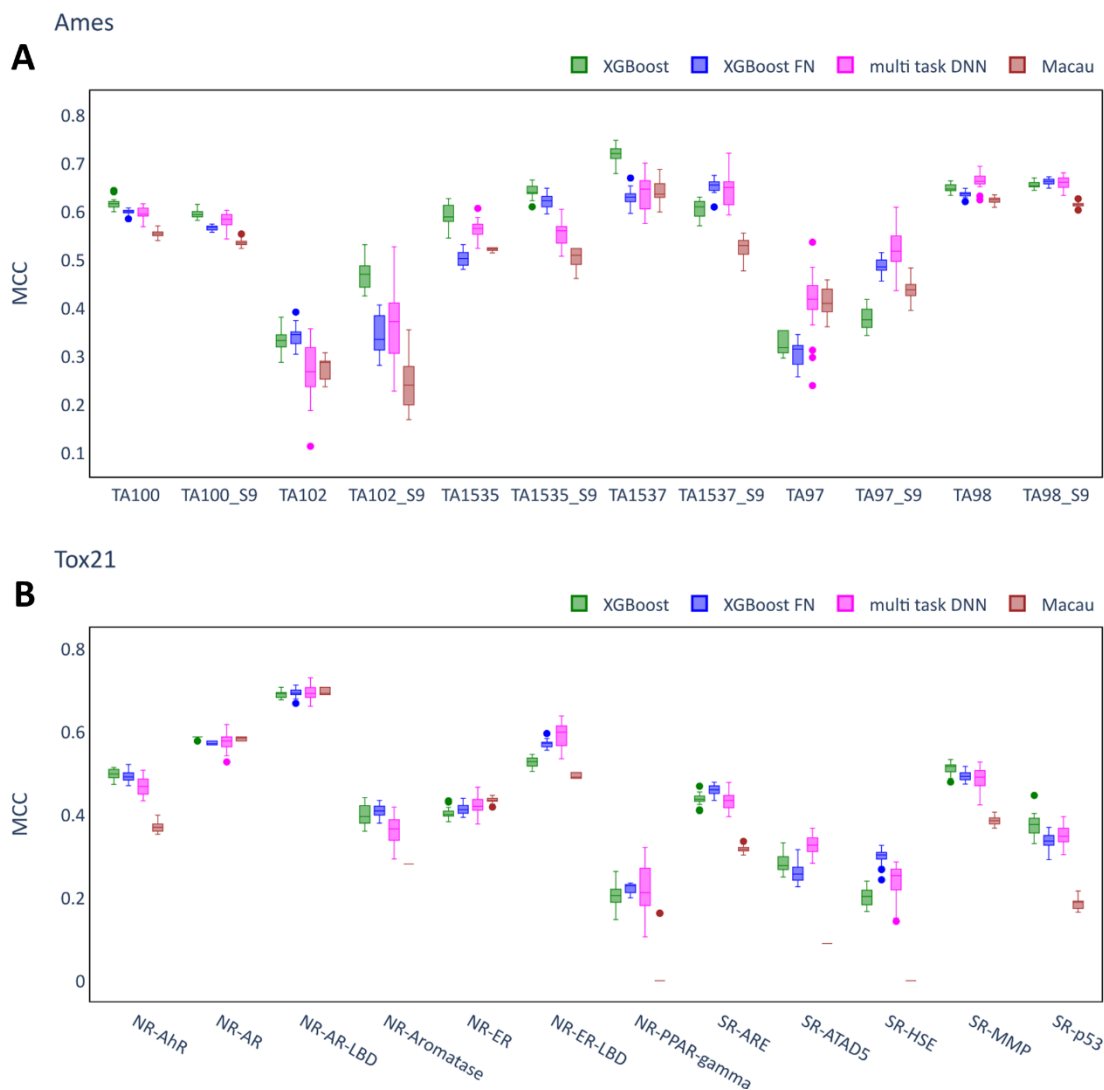


Figure 5-5 Performance of multi-task QSAR models. **A:** Ames dataset. **B:** Tox21 dataset. Each box summarises the MCC scores of 20 independent runs of the models on the test set with identical hyperparameters but differing random seeds. The XGB models (best performing single task QSAR model) are shown as a baseline. Only the best performing Feature Net model (XGB FN) is included in this plot.

Table 5-8 Median MCC scores of multi-task QSAR models. Median MCC scores and interquartile range for each technique and dataset on the test set across 20 random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model for each dataset in bold.

	Ames	Tox21
XGB	0.547 (0.545-0.550)	0.427 (0.422-0.430)
XGB-FN	0.529 (0.527-0.531)	0.435 (0.432-0.438)
Multi-task DNN	0.540 (0.527-0.549)	0.430 (0.423-0.437)
Macau	0.490 (0.484-0.499)	0.321 (0.319-0.323)

The FN models provided only a slight benefit over single task QSAR models and in some cases gave worse performances. This was the case for situations where no experimentally determined toxicity assay labels for the compounds in the test set were given as input to the model. However, there may

be situations where for a set of compounds experiments have been conducted for related assays and the aim is to predict labels for the not tested assays. FN models provide an obvious means to incorporate this type of information. This is by replacing predicted assay values by experimentally determined assay values in the feature vector of compounds in the test set. Figure 5-6 and Figure 5-7 report the MCC scores of these type of FN models compared to single task models and the FN not including the additional test labels for the Ames and the Tox21 dataset, respectively.

FN models with experimental test labels outperformed both the single task and the FN models without the additional test labels consistently across the different assays and algorithms, in many cases by a wide margin, for instance for TA97 (Figures 5-6A, 5-6B and 5-6C). The only two exceptions are NR-AR (Figures 5-7A and 5-7C) and NR-Aromatase (Figures 5-7A, 5-7B and 5-7C), where these models provided no benefit. Generally, the increases in performance were larger for the Ames dataset compared to the Tox21 dataset. This observation can be confirmed by considering the mean scores across the assays in Table 5-9. For instance, the RF-FN models provided an average MCC score exceeding the single task RF by nearly 0.2 (0.723 vs. 0.526), whereas the corresponding difference for the Tox21 dataset was smaller than 0.1 (0.490 vs. 0.402). Overall, RF-FN performed best on the Ames dataset, while XGB-FN achieved the highest average score on the Tox21 dataset.

The ranges of MCC scores observed in the different runs of the FN models were of comparable magnitude to those in the respective single task models and FN models without the additional assay labels. Like in the previous sections, the variances were higher for DNN based models (compared to RF and XGB based models) and were particular high for the same assays (e.g. TA102, NR-PPAR-gamma).

In summary, the FN models using test labels as input clearly outperformed both single task and conventional multi-task QSAR models. In fact, these models can be considered as multi-task imputation models, characterised by the partial input of experimentally determined assay labels for compounds in the test set. In the following sections, the improvements that imputation models provide over single task models will be investigated in more detail.

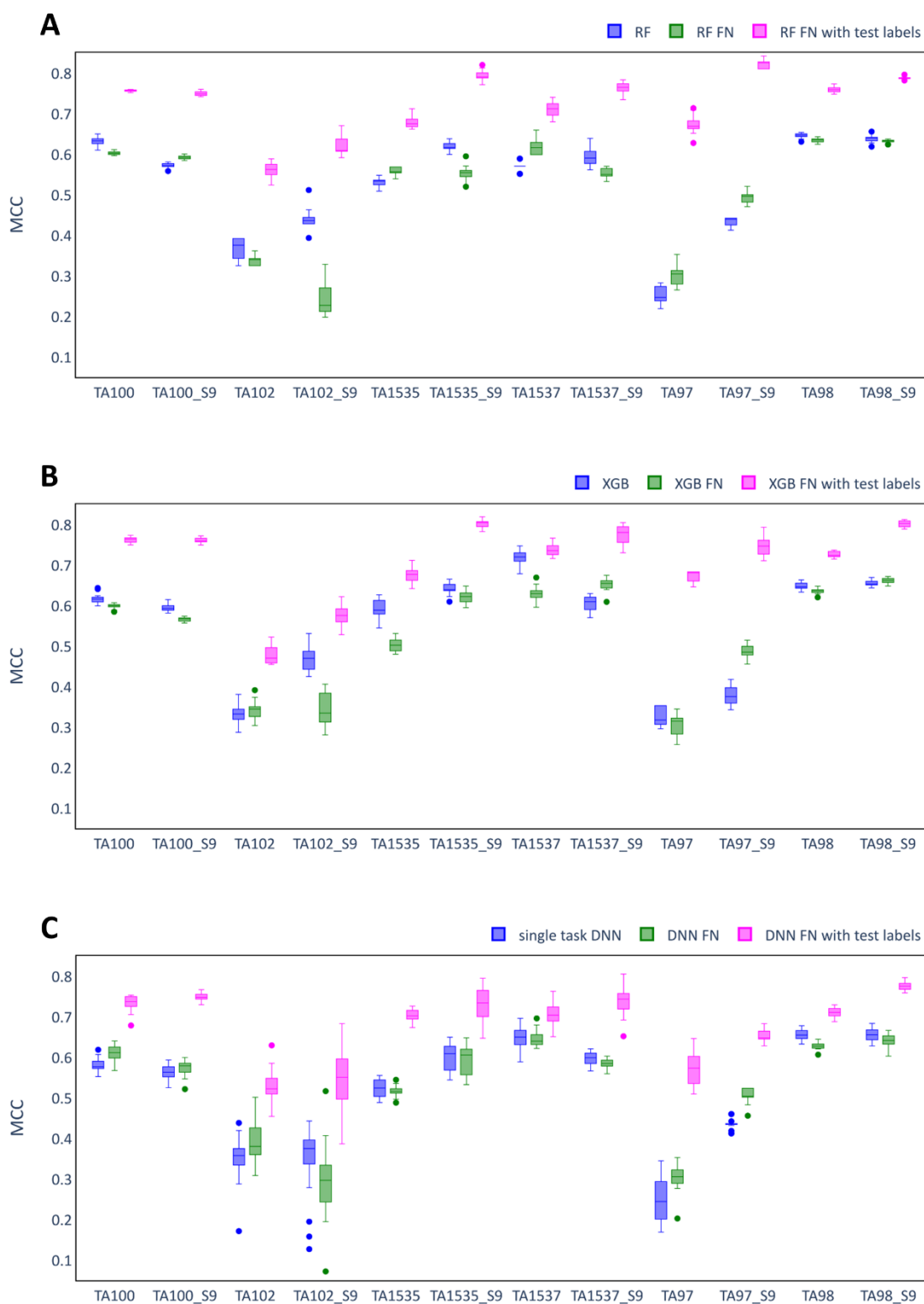


Figure 5-6 Performance of Feature Net models on the Ames dataset. A: RF models. **B:** XGB models. **C:** DNN models. The plots compare the MCC scores of Feature Net models to single task models obtained with the respective algorithm. The first Feature Net model in each plot (green colour) corresponds to the situation where no labels of the test set are used as input to the model, whereas in the second case (pink colour) all available labels of the test set (except for the predicted assay) are used as input to the model. Each box summarises the MCC scores of 20 independent runs of the model on the test set with identical hyperparameters but differing random seeds.

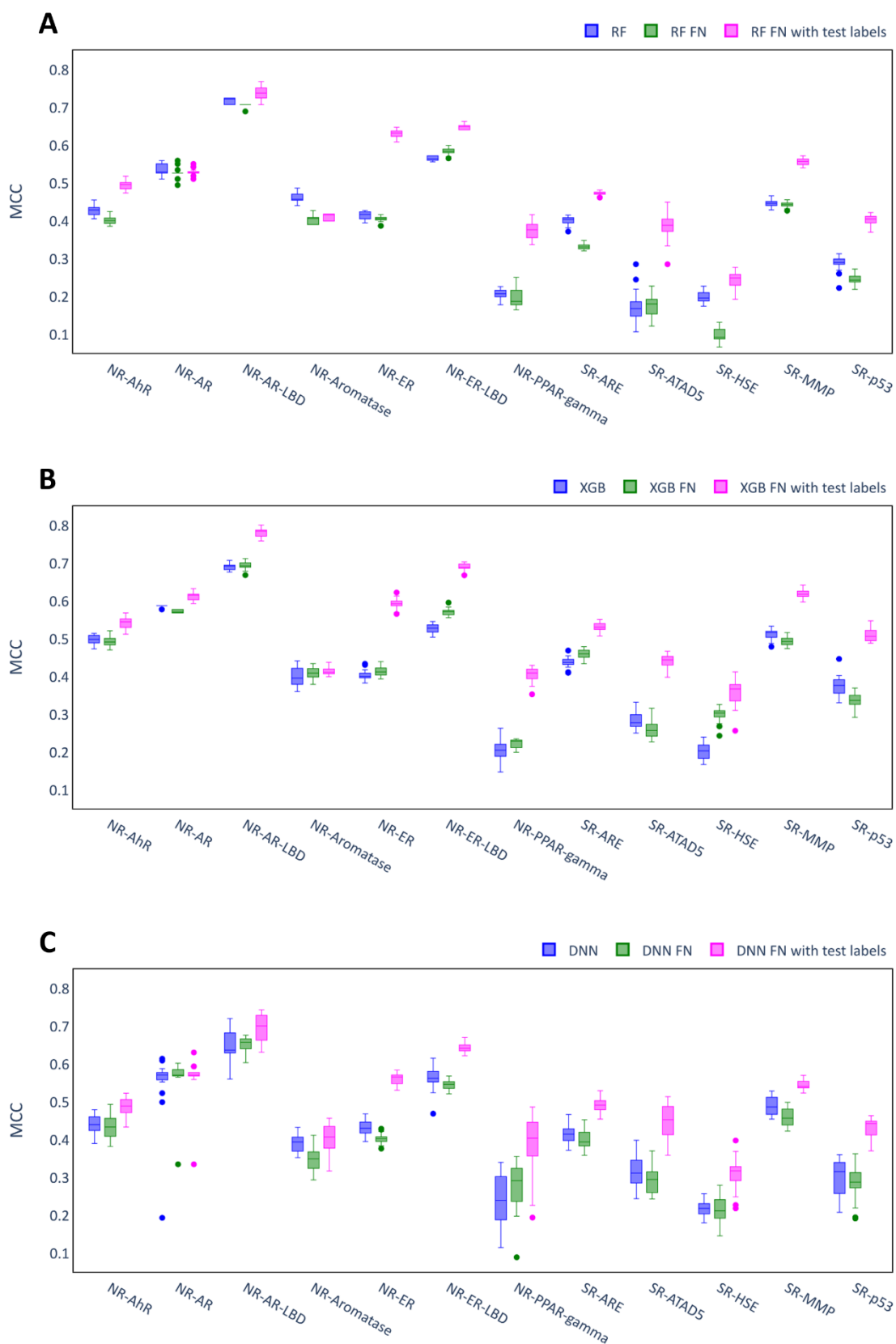


Figure 5-7 Performance of Feature Net models on the Tox21 dataset. A: RF models. **B:** XGB models. **C:** DNN models. The plots compare the MCC scores of Feature Net models to single task models obtained with the respective algorithm. The first Feature Net model in each plot (green colour) corresponds to the situation where no labels of the test set are used as input to the model, whereas in the second case (pink colour) all available labels of the test set (except for the predicted assay) are used as input to the model. Each box summarises the MCC scores of 20 independent runs of the model on the test set with identical hyperparameters but differing random seeds.

Table 5-9 Median MCC scores of Feature Net QSAR models. Median MCC scores and interquartile range for each technique and dataset on the test set across 20 random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model of each base algorithm (RF, XGB and DNN) for each dataset in bold.

		Ames	Tox21
RF	Single task	0.526 (0.523-0.527)	0.402 (0.400-0.405)
	FN without exp. test labels	0.509 (0.507-0.513)	0.376 (0.374-0.378)
	FN with exp. test labels	0.723 (0.721-0.726)	0.490 (0.487-0.494)
XGB	Single task	0.547 (0.545-0.550)	0.427 (0.422-0.430)
	FN without exp. test labels	0.529 (0.527-0.531)	0.435 (0.432-0.438)
	FN with exp. test labels	0.710 (0.704-0.713)	0.541 (0.538-0.543)
DNN	Single task	0.519 (0.509-0.527)	0.417 (0.410-0.426)
	FN without exp. test labels	0.526 (0.516-0.529)	0.408 (0.397-0.413)
	FN with exp. test labels	0.677 (0.675-0.682)	0.500 (0.488-0.504)

5.3.3 Imputation models

The results in Figure 5-6 and Figure 5-7 clearly hint at potential benefits of imputation models compared to conventional single task and multi-task QSAR models. To investigate imputation models further, a different scheme for splitting the available data in training and test set was employed. Specifically, compounds of the original dataset were randomly assigned to the training and test set on a per assay basis (assay-based splits). This means that a compound may be in training set for one assay, but in the test set for others. When the imputation model predicts the label of a particular assay-compound pair, other assay labels for this compound may be used as input to the model.

The multi-task imputation models trained for this study were FN models (based on RF, XGB and DNN models), multi-task DNN models and Macau models. Figure 5-8 compares the MCC scores of the different imputation techniques on both the Ames and the Tox21 dataset. Only the best performing FN model (XGB-FN) was included. The single task (imputation) models were re-trained for the assay-based splits and the best performing one (XGB) was included as a baseline model.

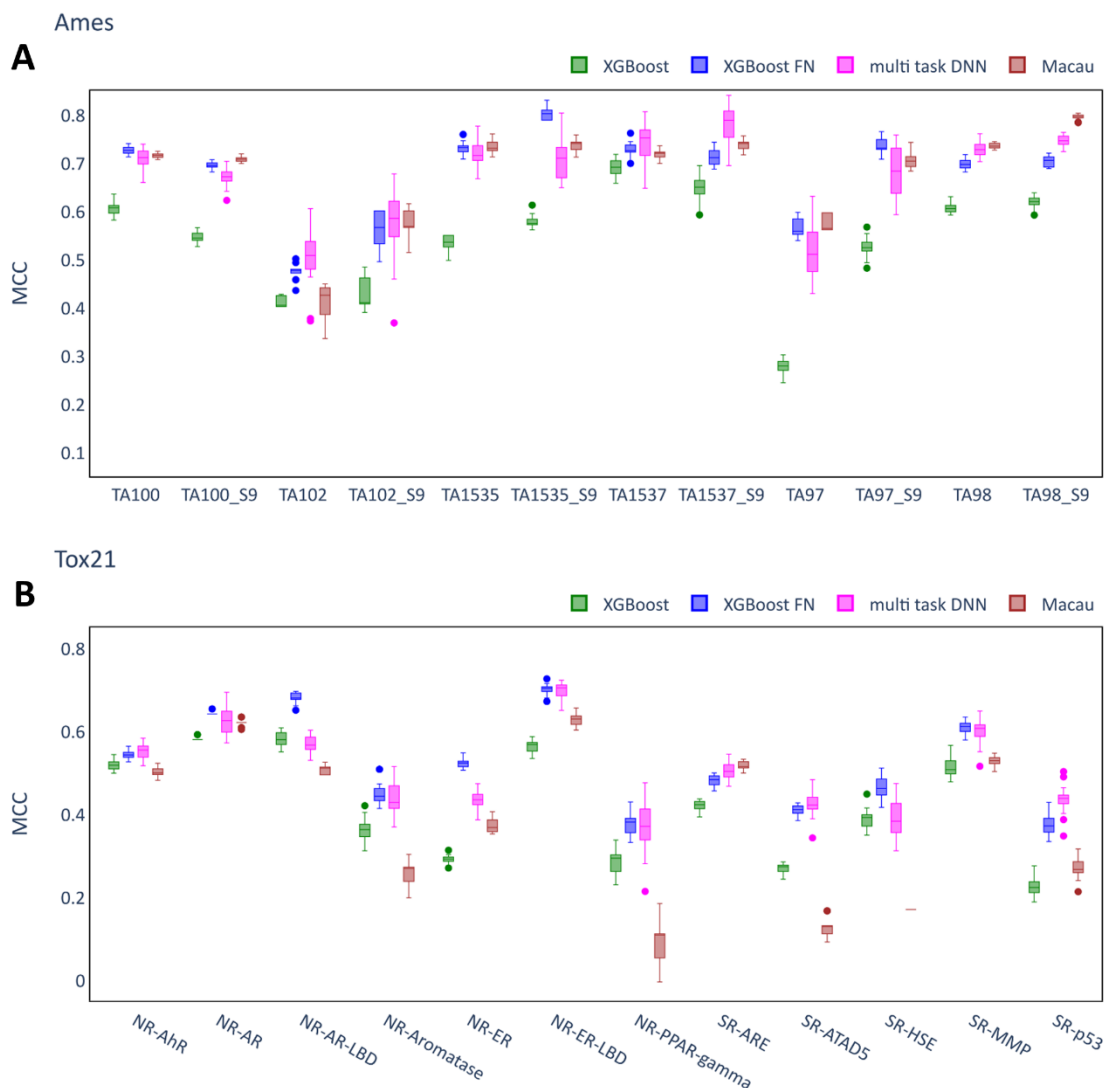


Figure 5-8 Performance of imputation models. A: Ames dataset. **B:** Tox21 dataset. Each box summarises the MCC scores of 20 independent runs of the model on the test set with identical hyperparameters but different random seeds. The XGB models (best performing single task QSAR model) are inserted as a benchmark. Only the best performing Feature Net model (XGB FN) is included in this plot.

All of the multi-task imputation models outperformed the XGB models on the Ames dataset (Figure 5-8A). In fact, the margin between the XGB models and all of the multi-task imputation models was remarkably large (more than 0.1 difference in median MCC score) for many of the assays (e.g. TA97, TA1535). Exceptions were TA102 and TA1537 where the benefit of the multi-task imputation models was comparatively small. The XGB models were also outperformed by the multi-task imputation models for the Tox21 dataset. However, this was not the case for the Macau models, which for many of the assays yielded a lower MCC score than the XGB models. Generally, the benefits of imputation models were smaller on the Tox21 dataset compared to the Ames dataset. There were few cases where the difference in median MCC score between the imputation model and the XGB

models was above 0.1 (e.g. XGB-FN for NR-ER). Consistent with the findings in the previous sections, multi-task DNN (like single task DNN and DNN-FN) models showed a comparatively large variability in performance across different runs of a particular model.

Table 5-10 reports the median MCC (MCC, F1 and ROC-AUC scores for individual assays in Tables B9 to B14 in Appendix B) scores across the assays for both datasets. Macau achieved the highest average score on the Ames dataset (0.679), yet the differences to XGB-FN (0.677) and multi-task DNN (0.676) are marginal. Conversely, Macau performed worse than the single task models on the Tox21 dataset, whereas XGB-FN (0.521) and multi-task DNN (0.503) achieved the highest average MCC scores.

Table 5-10 Median MCC scores of imputation models. Median scores and interquartile ranges for each technique and dataset on the test set across 20 different random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model for each dataset in bold.

		Ames	Tox21
Single task	RF	0.520 (0.517-0.523)	0.406 (0.404-0.412)
	XGB	0.540 (0.537-0.543)	0.415 (0.412-0.421)
	ST-DNN	0.500 (0.495-0.507)	0.415 (0.406-0.419)
Multi-task	XGB-FN	0.677 (0.675-0.682)	0.521 (0.516-0.525)
	MT-DNN	0.676 (0.667-0.688)	0.503 (0.493-0.512)
	Macau	0.679 (0.677-0.681)	0.385 (0.379-0.388)

The results conclusively show that multi-task imputation models can achieve higher predictive performance than single task QSAR models on various toxicity assays. In the following, the aim was to understand and explain which characteristics of the datasets were responsible for this effect. This included investigations regarding the chemical similarity between compounds in the datasets, the sparsity of the datasets and the relatedness between particular toxicity assays. Moreover, attempts were made to rationalise the stark contrast in performance between the two datasets for the Macau algorithm.

5.3.4 Roll of chemical similarity in imputation models

This section investigates the impact of chemical similarity on the effectiveness of imputation models. The test set was split into bins based on chemical similarity to compounds in the training set for each respective assay and performance of the models was evaluated independently on the individual bins. Chemical similarity was evaluated as average Tanimoto similarity to the five nearest neighbours in the training set and the ranges of the bins were: 0-0.4, 0.4-0.6, 0.6-1. It is well established that conventional QSAR models tend to perform poorly on compounds that are chemically dissimilar to

compounds in the training set (Sheridan et al., 2004) and the aim was to see whether a similar trend can be found for imputation models.

Figure 5-9 shows the MCC scores for the different bins for the assays TA100 (5-9A), TA100_S9 (5-9B), TA98 (5-9C), TA98_S9 (5-9D) and the average across those four assays (5-9E). The remaining assays were excluded from this analysis, as they had fewer than 100 compounds in each bin, which resulted in high variance. As expected, single task XGB performance was higher for more similar compounds (Figure 5-9A-D). The XGB models performed quite badly on the most dissimilar compounds (0-0.4), especially on the assays TA98 and TA98_S9 with median MCC scores between 0.2 and 0.3 (Figure 5-9C and 5-9D). All imputation techniques achieved much higher scores for this bin. Most notably, Macau achieved MCC scores in the range 0.65 to 0.7 for these two assays, but also multi-task DNNs (0.6 to 0.65) and XGB-FN (0.5 to 0.55) models outperformed the XGB models by a wide margin. Similar trends were observed for the TA100 (5-9A) and the TA100_S9 (5-9B) assays, although the benefit of imputation models seemed more moderate for the former assay.

The multi-task models also consistently outperformed the XGB models on the bins of more similar compounds (similarity values between 0.4-0.6 and between 0.6-1) and, similarly to the XGB models, the performance of the multi-task models tended to increase for more similar compounds. However, the margins between the single task XGB model and the multi-task models were much smaller.

These observations on the relative performance of the single task and the multi-task methods at low similarity values are supported by considering the averages across the assays in Figure 5-9E. Clearly, the largest numerical benefit of the multi-task imputation techniques was found for the bin containing the most dissimilar compounds, where XGB as a conventional QSAR model performed relatively badly. Generally, the different multi-task imputation methods achieved comparable MCC scores on the different bins, with the exception that XGB-FN performed somewhat worse on the dissimilar compounds than the other techniques. These results show that the multi-task imputation models were less beneficial on the test compounds which are highly similar to compounds in the training set, but this is likely due to the higher scores overall.

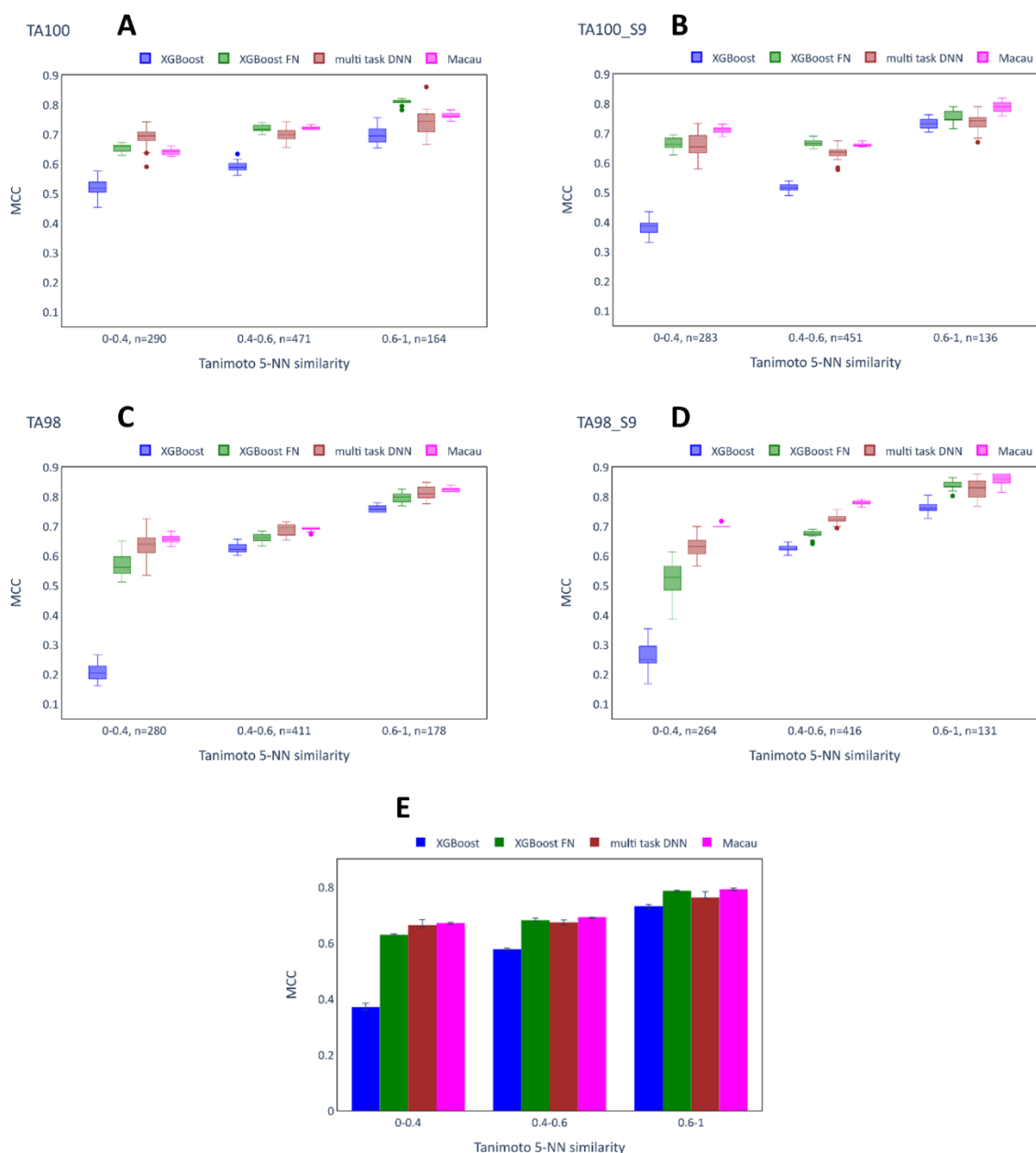


Figure 5-9 Performance of imputation models according to test compound chemical similarity: Ames. A-D show the MCC scores obtained in 20 independent runs (same hyperparameters, different random seeds) of the models on the different bins for TA100, TA100_S9, TA98 and TA98_S9. The number (n=x) written next to each bin indicates the number of compounds placed in this bin. E shows the average MCC scores across the assays (by computing the mean MCC score for each random seed across the assays and then taking the median across different random seeds).

Figure 5-10 shows the MCC scores of 4 representative assays (A: NR-Aromatase, B: NR-ER, C: SR-ARE, D: SR-p53) of the Tox21 dataset for different test set similarity bins, as well as the average across all assays (E).

As for the Ames dataset, the MCC scores of the XGB single task models were consistently higher for bins of higher chemical similarity. This effect was particularly strong for the NR-ER and the NR-p53

datasets. Likewise, the MCC scores of the multi-task models tended to increase for more similar compounds. An exception is the SR-p53 assay, where Macau achieved a MCC score of zero in the bin of highest chemical similarity. Generally, the results on this bin seem out of line, as the variability for all of the other models was very high (in the most extreme case ranging from -0.01 to 0.864 for multi-task DNN). However, this is explained by the very low number of actual toxic compounds in this bin (4 out of 195 (2.1%), whereas the 0.4-0.6 bin contains 34 toxic compounds (4.9%) and the 0-0.4 bin contains 48 toxic compounds (9.3%)), such that small changes in predictions made by the model have a very large effect on the MCC score.

Overall, the multi-task methods (except Macau) achieved higher scores than the XGB models on the Tox21 dataset. For the assays NR-ER and SR-ARE, the largest benefit was found for the bin representing the chemically most dissimilar compounds, as was the case for the Ames dataset. For this bin and these assays, the XGB model performed particularly poorly with an MCC of around 0.1, while some multi-task models achieved MCC values up to 0.5. In the NR-Aromatase assay, both the XGB-FN (0.863) and the multi-task DNN (0.703) model achieved remarkably high median scores compared to the XGB models (0.495). However, similarly to SR-p53, the variance for the multi-task DNN was extremely large, which complicates the interpretation. When considering the average values across all assays, both XGB-FN and multi-task DNN outperformed the XGB models on all of the bins. On average, the highest benefit of the multi-task models was found for the most dissimilar compounds, but this trend was less clear compared to the Ames dataset.

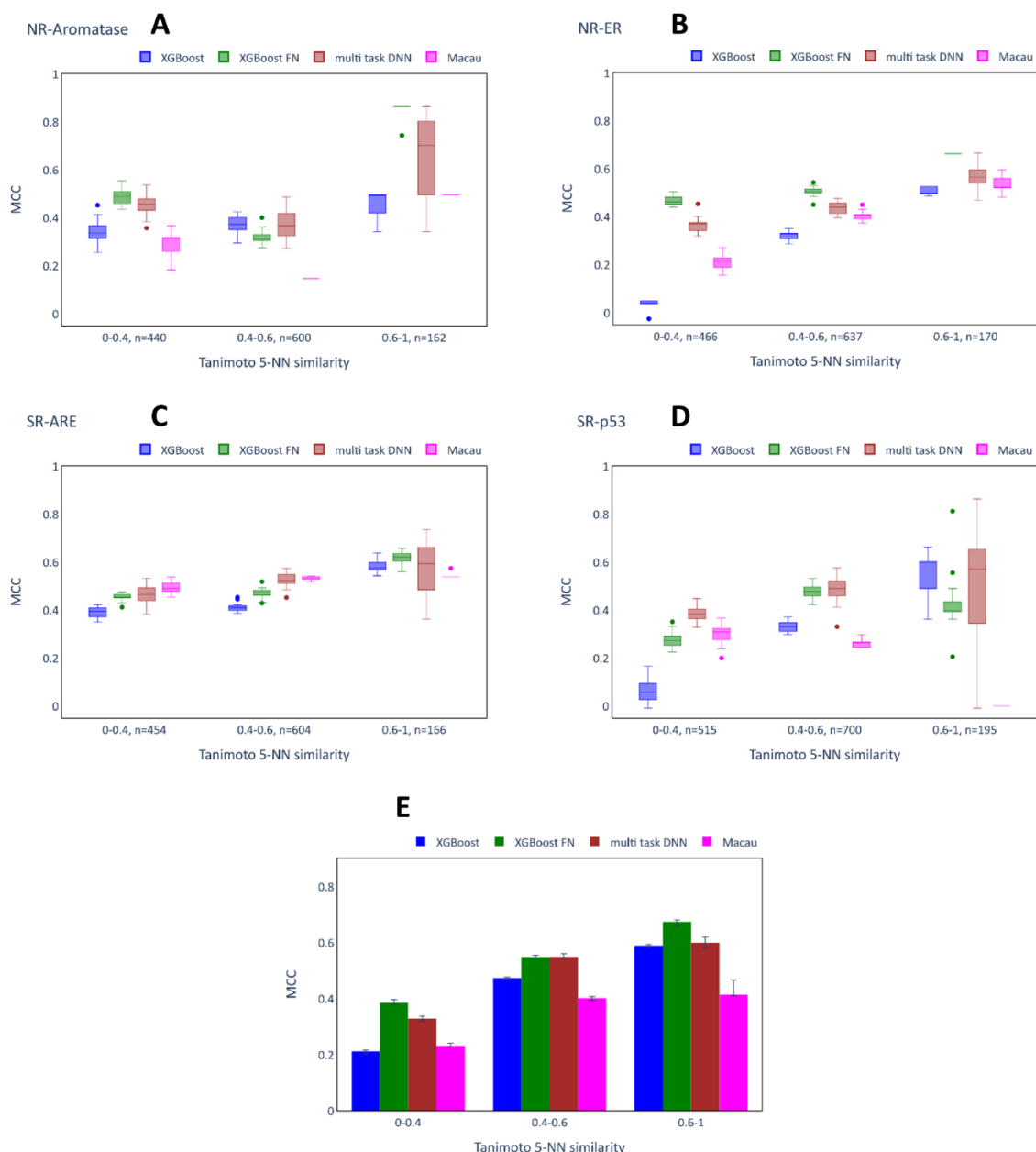


Figure 5-10 Performance of imputation models according to test compound chemical similarity: Tox21. A-D show the MCC scores obtained in 20 independent runs (same hyperparameters, different random seeds) of the models on the different bins for NR-Aromatase, NR-ER, SR-ARE, SR-p53 (representative assays). The number (n=x) written next to each bin indicates the number of compounds placed in this bin. E shows the average MCC scores across the assays (by computing the mean MCC score for each random seed across the assays and then taking the median across different random seeds).

5.3.5 Role of data sparsity in imputation models

The role of data availability on the effectiveness of single task and multi-task imputation models was investigated. As for the analysis on chemical similarity, the analysis was done on predicted values from the previous evaluation of single task and multi-task imputation models. The test set for each assay was divided into three bins according to the number of experimentally determined data labels (0-1,

2-3, and >3 available labels) each compound has for the 11 remaining assays in the training set. The multi-task imputation models incorporate information about the remaining assays whereas the XGB models (single task models) do not. This analysis was only performed for the Ames dataset, as the lower sparsity of the Tox21 dataset was such that the bins of low data availability were not sufficiently populated. The analysis for the Ames dataset was limited to assays for which at least 100 compounds could be placed in each of the bins, and these are the same assays as considered for the chemical similarity studies. The MCC scores for these assays (TA100, TA100_S9, TA98, TA98_S9) for the different bins and the average scores across these assays are reported in Figure 5-11.

For the first bin (0-1 available labels), the MCC scores of the multi-task models tended to be only slightly higher than those for the XGB models. The multi-task DNN models were the only imputation models with higher median MCC score than the XGB model across all the assays. The other multi-task models achieved lower median scores than the XGB models for this bin in one of the assays (Macau for TA100: 0.515 vs. 0.541 and XGB-FN for TA98-S9: 0.380 vs. 0.512). For the remaining two bins, which represent a higher number of available toxicity labels for the test compounds (2-3 and >3 available labels), the XGB models were outperformed by all multi-task techniques for all of the assays. The differences in MCC score between the multi-task models and the XGB models were largest for the third bin (>3 available labels), with the highest uplift occurring for the Macau model on the TA98_S9 assay (0.829 vs. 0.520). Generally, all of the multi-task imputation models achieved similar scores for the second and third bin, but Macau performed better than the other imputation techniques for the third bin.

The observations for the single assays are supported when considering the averages across the assays as depicted in Figure 5-11E. For the first bin, the XGB model was outperformed by all the multi-task models, albeit by a comparatively small margin. The margin between single task and multi-task models increased with more available data labels for test compounds. Notably, Macau outperformed the other multi-task approaches on the bin with >3 test data labels.

This analysis shows that the number of available experimentally determined assay labels for test compounds strongly affected the performance of the imputation models. For the Ames dataset, the multi-task models clearly performed better on compounds with a high number of available assay labels.

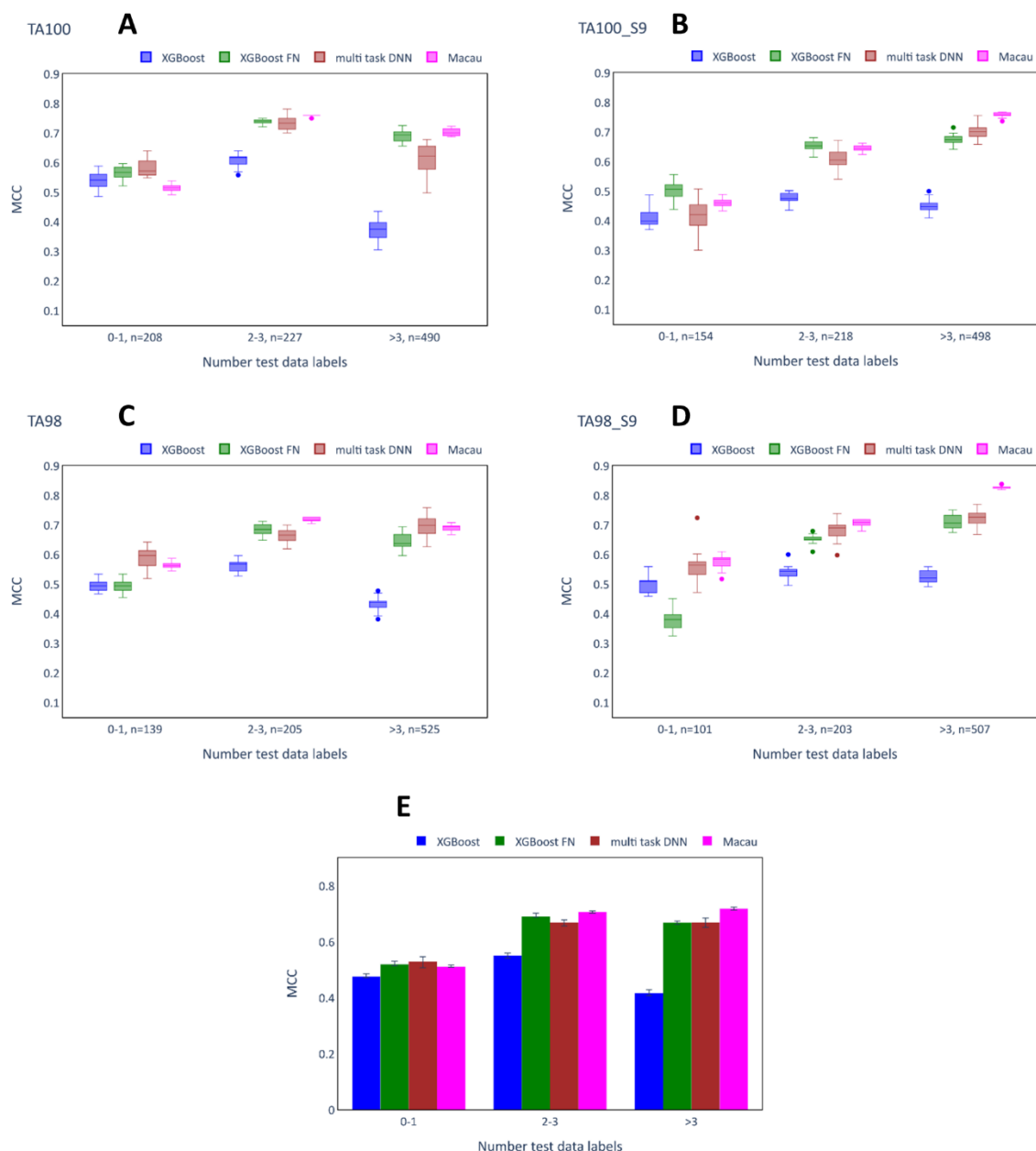


Figure 5-11 Performance of imputation models according to test compound data label availability. A-D show the MCC scores obtained in 20 independent runs (same hyperparameters, different random seeds) of the models on the different bins for TA100, TA100_S9, TA98 and TA98_S9. The number (n=x) written next to each bin indicates the number of compounds placed in this bin. E shows the average MCC scores across the assays (by computing the mean MCC score for each random seed across the assays and then taking the median across different random seeds).

5.3.6 Analysis on the impact of assay relatedness on imputation models

The contributions of single assays to the overall success of multi-task imputation models were investigated focussing on XGB-FN models. Two different approaches were used: (i) pairwise FN models, where for a given target assay a separate FN model is trained with each of the remaining assays as auxiliary one in turn (called pairwise FN models) and (ii) leave-one-assay-out (LOAO) FN

models, where starting from the full FN model for a given target assay, each of the auxiliary assays is left out one at a time. Figure 5-12 reports the performance of the pairwise FN models. The heatmaps report the performances of each FN model compared to the respective single task. In Figure 5-13, the scores of the LOAO FN models are reported, compared to the scores of the full FN models.

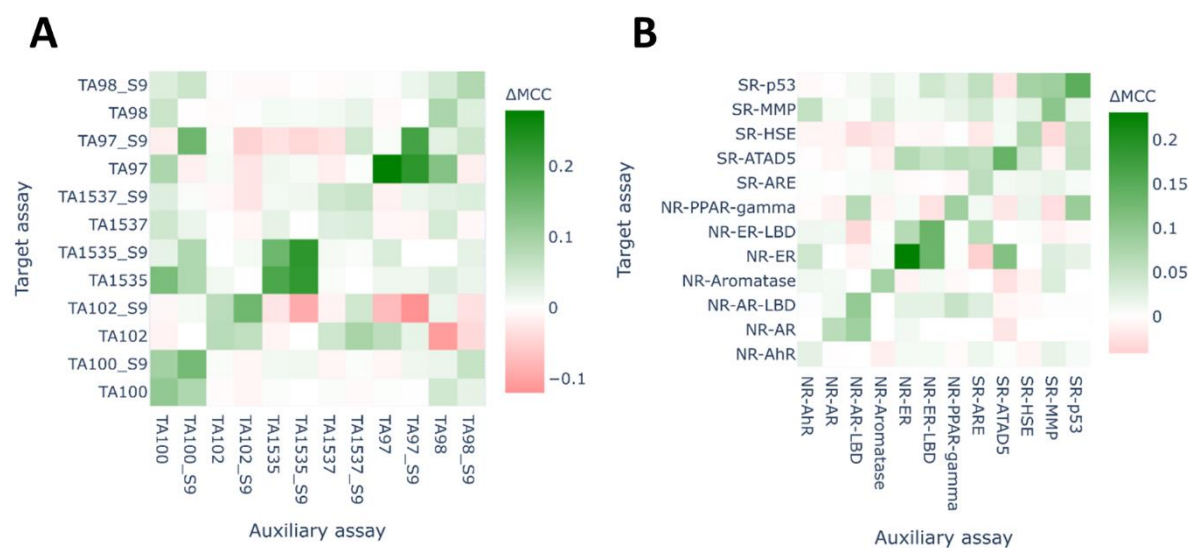


Figure 5-12 Performance of the pairwise FN models. A: Ames, B: Tox21. For each assay pairwise FN models were trained with each of the remaining assays and for each pair 20 independent runs of the model were conducted using different random seeds. To obtain the heatmap, the median MCC score was computed for each pair and the median MCC of the single task model for respective target assay was subtracted. The diagonals represent the differences in MCC score between the full FN model and the single task XGB model as a reference.

For the Ames dataset, the pairwise XGB-FN models achieved a higher MCC score than the single task XGB models in many cases (as shown by the green cells in the heatmap outside the diagonal, Figure 5-12A). The diagonal of the heatmap shows the difference between the MCC scores for the full FN models compared to single task models. Generally, the pairwise models did not achieve as high scores as the full FN models, with the exception of TA1535 with TA1535_S9. However, in a few cases, the pairwise FN model approximated the performance of the full FN model quite well (e.g. the MCC of the TA97 full model was 0.279 higher than the MCC of the single task model, whereas, the improvement of the pairwise model comprising TA97 and TA97_S9 was 0.230). In many cases, the pairwise FN model provided a substantial benefit (improvement over 0.05) compared to the single task model, even if this was smaller than that achieved for the full FN model. There were also many cases where the pairwise FN model showed very small differences compared to the single task models (shown by the white and very pale cells in the heatmap). Red cells indicate reduced performance of the pairwise models compared to the single task models. These cases were relatively rare overall, but occurred frequently in the assays with the fewest data points which also showed a high variance between different runs of a model (TA102, TA102_S9, TA97, TA97_S9). Unsurprisingly, the Ames strain results

with and without S9 are highly correlated and this is reflected in the consistent increase in performance compared to the single task models for the pairs of the same bacteria strain, represented by the accumulation of green cells adjacent to the diagonal. Another key finding is that the four assays with the most data points (TA100, TA100_S9, TA98, TA98_S9) as auxiliary assays resulted in at least a moderate increase in performance in most cases, suggesting that the number of available experimentally determined data points impacts on the performance of the pairwise FN models.

The performances of the pairwise XGB-FN models on the Tox21 dataset are shown in Figure 5-12B. Similar to the Ames dataset, the pairwise XGB-FN models achieved a higher MCC score than the single task XGB models in many cases. In two of the cases (NR-AR with NR-AR-LBD and with NR-PPAR-gamma) the pairwise model achieved a higher score than the full FN model. However, for most of the target assays the full FN model clearly performed better than any pairwise FN model. The Tox21 dataset contains two pairs of assays that measure the same target in a different test system (NR-AR/NR-AR-LBD and NR-ER/NR-ER-LBD) and it was therefore expected that these pairs would yield the best performing pairwise FN models. For NR-ER and NR-ER-LBD this was indeed the case, although other auxiliary assays yielded models of comparable performance (SR-ATAD5 for NR-ER and SR-ARE for NR-ER-LBD). NR-AR-LBD was the best auxiliary assay for NR-AR, but the same was not true for the opposite case (NR-PPAR-gamma was the best auxiliary assay for NR-AR-LBD). Some of the pairwise models performed worse than the respective single task models (red cells in Figure 11B). For the target assays NR-PPAR-gamma and SR-HSE this occurred for many of the pairs, yet the full FN model performed better than the single task models, which can be attributed to the few auxiliary assays that resulted in improved models (SR-p53 and NR-AR-LBD for NR-PPAR-gamma, and SR-p53 for SR-HSE).

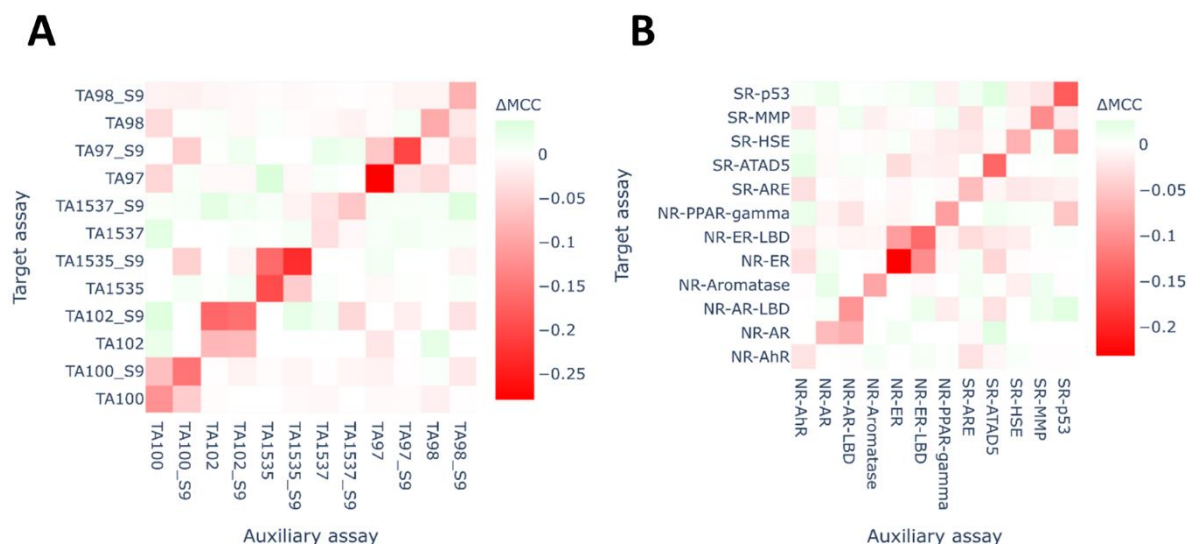


Figure 5-13 Performance of the leave-one-assay-out FN models. A: Ames, B: Tox21. For each assay FN models were trained with one of the remaining assays left out, one at a time and like for the pairwise FN models 20 runs with different random seeds were performed. To obtain the heatmap, the median MCC score was computed for each leave-one-assay-out FN model and the median MCC of the full FN model was subtracted. The diagonals represent the differences in MCC score between the single task XGB model and the full XGB FN model as a reference.

The heatmaps in Figure 5-13 indicate by how much the MCC score of the LOAO FN model decreased compared to the full FN model, with the diagonal showing the decrease for the single task models as a reference. In nearly all of the cases the decrease of performance when using a single task model was larger than the decrease of LOAO FN models, indicating that the success of the full FN model cannot be fully attributed to the presence of a single auxiliary assay. Exceptions were TA102_S9 for the Ames dataset and SR-HSE for the Tox21 dataset, where the LOAO model missing TA102 or SR-p53, respectively, performed worse than the single task models. In several cases, the removal of a single assay led to a model that performed better than the full FN (green cells in the matrix), yet those improvements were quite small (maximal for TA97 with TA1535 removed: +0.039). For the assays TA1537 and TA1537_S9, the removal of most of the different assays actually resulted in an increase of performance, albeit small. Generally, the majority of removals overall had only small effects on the performance (either positive or negative) compared to the full FN models. Similar to the pairwise FN models, the removal of assays of the same bacteria strain in the Ames dataset resulted in a comparatively high decrease in performance. For the pairs of assays in the Tox21 dataset measuring the same targets (NR-AR/NRAR-LBD and NR-ER/NR-ER-LBD) the findings were consistent with the findings for the pairwise FN models. NR-ER and NR-ER-LBD were the most useful auxiliary assays for each other, but the benefits of the full FN models cannot be fully explained by the presence of these closely related assays. In contrast, it seemed that the benefit of the full FN model for NR-AR can be mainly attributed to NR-AR-LBD as auxiliary assay. On the other hand, NR-AR-LBD was not found to be

a useful auxiliary assay for NR-AR, the respective LOAO FN model performed marginally better than the respective full FN model.

The pairwise FN models and the LOAO FN models are two different approaches to estimate the contribution of a single assay to the full FN models. Figure 5-14 compares the findings of the two approaches by plotting the benefit of the pairwise FN model on the x-axis against the cost of removing the auxiliary assay in a LOAO FN model on the y-axis, for each pair of target assay and auxiliary assay. For both datasets, most points are located around (0,0), meaning that this particular auxiliary assay had neither a large beneficial effect for the target assay in a pairwise FN model, nor was there a large decrease when this auxiliary assay was left out. Overall, there seemed to be a trend that large increases for pairwise FN models were associated with large decreases for leave-one-assay-out FN models, but the correlation was not very strong with Pearson correlation coefficients of -0.53 and -0.62 for the Ames and Tox21 dataset, respectively. In a few cases, there is a deviation from this trend in the Ames dataset. For the pair TA97-TA97_S9 (target-auxiliary), a large increase for the pairwise FN model (+0.23) was found together with a small decrease for the LOAO FN model (-0.026). In such cases, the auxiliary assay was obviously useful in the pairwise model and the lack of the assay in the LOAO approach could apparently be compensated for by the information provided by the remaining assays. In the opposite case, for instance for TA97_S9-TA97, a low benefit for the pairwise FN model (+0.007) is associated with a larger drop for the leave-one-assay-out FN model (-0.06), which could mean that the information provided by the auxiliary assay was only meaningful in combination with other assays. These extreme cases suggest that neither the pairwise FN model approach nor the LOAO FN approach alone can fully describe the contribution of an auxiliary assay to the success of the full FN model, and they can be considered as complementary approaches. Despite the insights gained using these approaches, the improvements of full FN models cannot be fully explained by additive effects of single assays. Instead, it seems that combinatorial effects between the auxiliary assays are at play, which are difficult to disentangle.

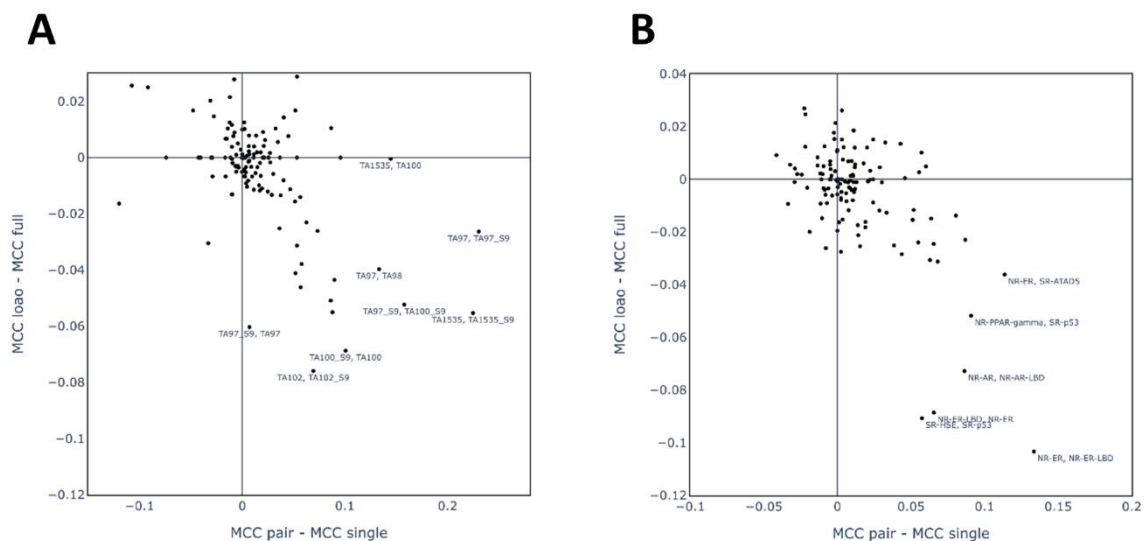


Figure 5-14 Concordance between the pairwise and LOAO FN models. A: Ames, B: Tox21. The scatter plots show for each target assay-auxiliary assay pair the difference in MCC score between the pairwise FN model and the single task model on the x-axis and the difference between the LOAO FN models and the full FN models on the y-axis. In the labels of the dots, the first assay is the target assay and the second the auxiliary assay.

The previous results showed that a single assay can have large influence on the success of FN models (either as improvement in pairwise FN models or as decrease in LOAO FN models). However, it is unclear whether this influence can be explained by the relationships in the data between the assays. The relationship between a target assay and an auxiliary assay was measured by computing their mutual information and dividing by the entropy of the target assay. This metric (MI-entropy ratio) estimates the proportion of the total information in the target assay that is included in the auxiliary assay. Figure 5-15 and Figure 5-16 show how this metric relates to the performances found in the pairwise FN models and the LOAO FN models for the Ames and Tox21 dataset, respectively.

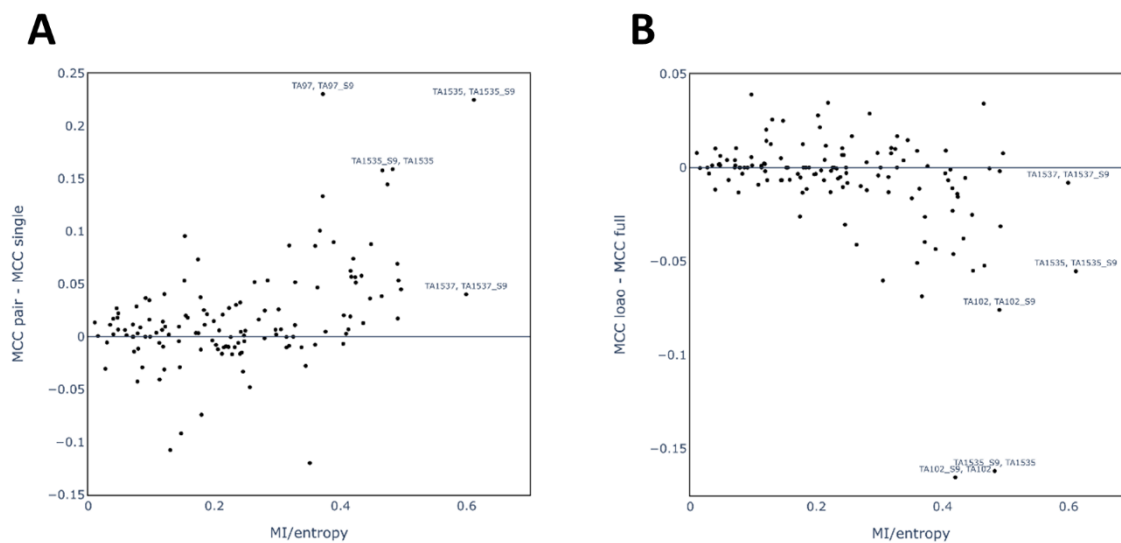


Figure 5-15 Effect of assay relatedness on FN models: Ames. For each target assay-auxiliary assay pair, the difference between MCC of pairwise FN model and single task model (A) or the difference between LOAO FN model and full FN model (B) are plotted vs. the ratio of mutual information (MI) between the two assays and the entropy of the target assay.

For the Ames dataset, neither the improvements of pairwise FN models over single task models, nor the decrease in performance for LOAO FN models compared to full FN models are very strongly correlated to the metric for assay relatedness (Pearson correlation coefficients: 0.48 and -0.43). Nonetheless, strong increases for pairwise models ($>+0.1$) or strong decreases for LOAO FN models (<-0.05) only occurred for pairs where the MI-entropy ratio is above 0.3. Hence, it seems that a close relatedness between the target assay and the auxiliary assay was necessary but not sufficient for a strong effect of that auxiliary assay on the FN model. The pair TA1537 with TA1537_S9 represents a case where a strong relatedness of the assays resulted in an apparently small effect. However, the performance of single task XGB on TA1537 was very high (median MCC: 0.691, Figure 5-8) and a much larger MCC score on this dataset may not be possible due the uncertainty in the toxicity labels.

As for the Ames dataset, an increase in the MI-entropy ratio tends to lead to the improvements of pairwise FN models over single task models and the decrease in performance compared to the full FN models, albeit the correlation is not very strong (0.50 and -0.52). A striking exception from these trends represents the pair NR-AR-LBD with NR-AR. However, the other pairs with a high MI-entropy ratio (e.g. NR-ER with NR-ER-LBD) are amongst the pairs where the auxiliary assay has the strongest effects on the FN model. Overall, the values for the MI-entropy ratio are lower on the Tox21 dataset, which might explain why the imputation models provide a larger numerical benefit on performance for the Ames dataset. The findings on both datasets suggest that the MI-entropy ratio might be a

useful metric to estimate which auxiliary assays could provide the strongest benefit in a FN model. However, clearly a high value does not guarantee a strong benefit.

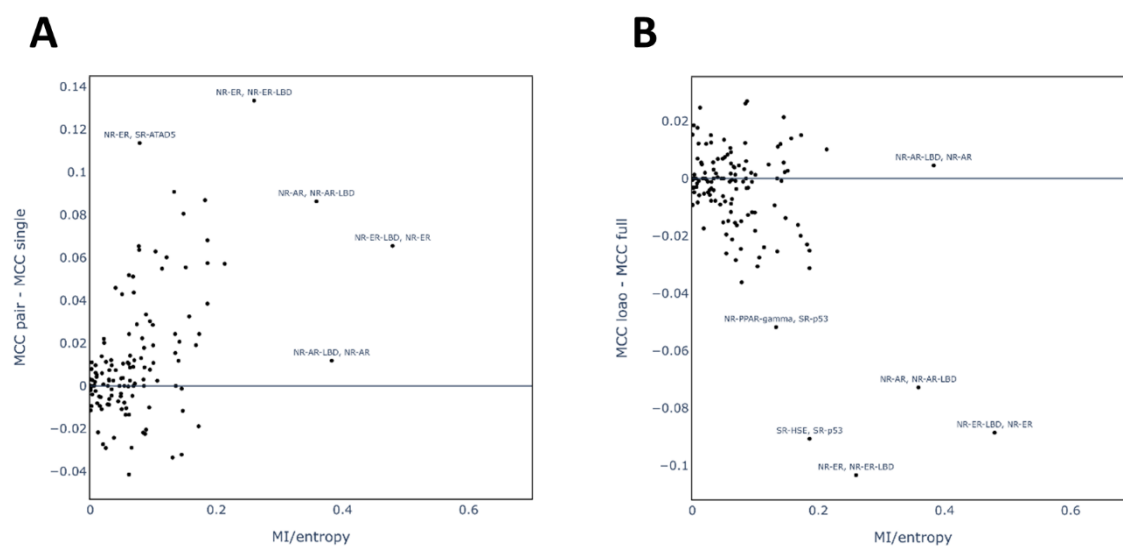


Figure 5-16 Effect of assay relatedness on FN models: Tox21. For each target assay-auxiliary assay pair, the difference between MCC of pairwise FN model and single task model (A) or the difference between LOAO FN model and full FN model (B) are plotted vs. the ratio of mutual information (MI) between the two assays and the entropy of the target assay.

5.3.7 Analysis of the classification threshold used in Macau models

Macau models were found to perform poorly on most assays in the Tox21 dataset. This section aims to understand this behaviour by focussing on the p53 assay. The MCC scores for Macau were clearly worse (median: 0.267) than those for the XGB-FN (median: 0.372) and multi-task DNN (median: 0.438) models. However, the ROC-AUC score on this assay for Macau (median: 0.899) was higher than for both multi-task DNN (median: 0.873) and XGB-FN (median: 0.862). To understand the stark contrast between those two metrics, the predictions for XGB-FN and Macau were carefully inspected. Figure 5-17 plots the predicted probability scores (for the active class) for each compound in the test set for both XGB-FN (Figure 5-17A) and Macau (Figure 5-17B), where the dots representing experimentally toxic compounds are coloured red and those for non-toxic compounds are coloured blue. It can be observed that for both XGB-FN and Macau a large proportion of test compounds received a very low prediction score (>80% smaller than 0.1) and most of these compounds are in fact non-toxic. As expected, most of the compounds with high predicted scores were found to be toxic (increasingly red dots towards the right). While the XGB-FN model provided scores in the full range from 0 to 1, the Macau model did not produce scores higher than 0.74. Table 5-11 lists the proportion of

experimentally toxic compounds for 10 bins of predicted probability scores. For both XGB-FN and Macau, the proportion of actual toxic compounds tends to increase with higher scores.

The MCC scores for all the models were computed by using the default value of 0.5 as threshold for the binary classification of toxic and non-toxic compounds. For the Macau model this threshold means that only 9 of the 1409 compounds in the test set were predicted as toxic, resulting in a poor MCC score. Choosing a lower classification threshold would lead to more compounds being predicted as active. The second column for XGB-FN and Macau in Table 5-11 lists the MCC scores obtained for various classification thresholds. For Macau, the MCC score can be increased from 0.247 to 0.487 if 0.2 is used as threshold instead of 0.5. In that case, 107 compounds would be predicted as toxic instead of nine test compounds. A higher MCC score is also obtained for XGB-FN when choosing a decision threshold different from 0.5, but the difference is comparatively small.

These results demonstrate that the apparently poor performance of Macau for the p53 assay can be attributed to an inappropriate classification threshold. This analysis was only conducted for the p53 assay, but similar observations are expected for other assays in the Tox21 dataset. Overall, the performance of Macau on the Tox21 dataset may be competitive to the other imputation techniques, if the classification threshold is carefully selected. Of course, this selection must be made based on findings in the training set, otherwise information from the test set would influence the model design, which must not happen.

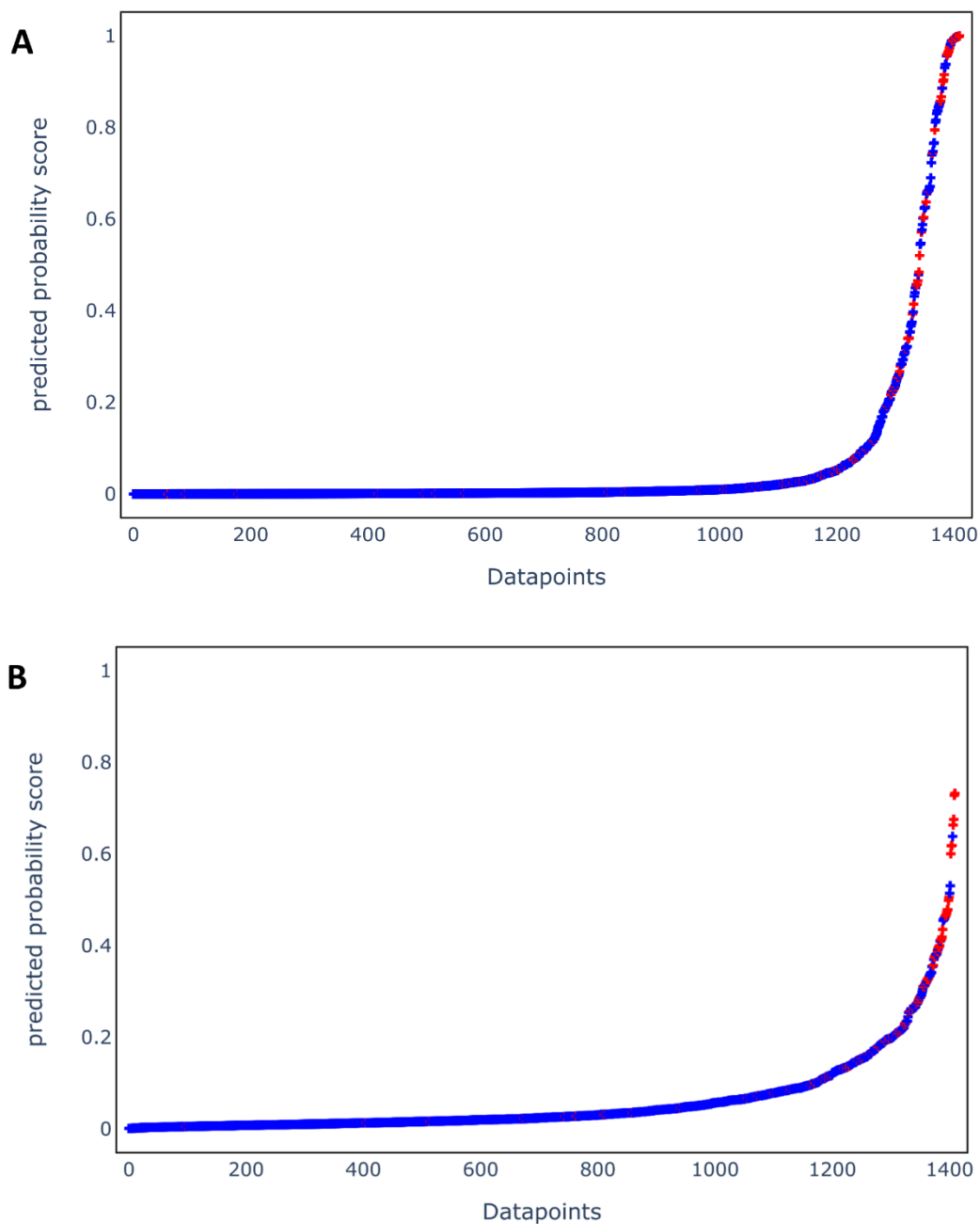


Figure 5-17 Predicted probability scores for the p53 assay. Shown are the predicted probability scores from XGB-FN (A) and Macau (B) of a single run of each model for each compound in the test set, sorted by ascending scores. Experimentally actives are coloured red, while inactives are coloured blue.

Table 5-11 Analysis of the classification threshold. The full range of possible predicted probability scores (0-1) was divided into 10 bins of equal width and compounds were placed in the bin according to their predicted probability score by a single run of either the XGB-FN or the Macau model. The first column for XGB-FN and Macau breaks down which proportion of compounds in the respective bin of predicted probability scores are experimentally determined actives. The second column lists the MCC scores obtained if the upper bound of the respective bin is chosen as classification threshold.

bin	XGB-FN		Macau	
	Experimentally actives (total data instances)	MCC (at upper threshold)	Experimentally actives (total data instances)	MCC (at upper threshold)
0-0.1	0.031 (1251)	0.349	0.015 (1171)	0.43
0.1-0.2	0.054 (37)	0.396	0.144 (132)	0.487
0.2-0.3	0.231 (26)	0.39	0.333 (51)	0.449
0.3-0.4	0.294 (17)	0.376	0.483 (29)	0.375
0.4-0.5	0.5 (10)	0.341	0.688 (16)	0.247
0.5-0.6	0.333 (6)	0.332	0.333 (3)	0.257
0.6-0.7	0.214 (14)	0.340	0.833 (6)	0.148
0.7-0.8	0.286 (7)	0.339	1 (2)	0
0.8-0.9	0.231 (13)	0.360	0 (0)	0
0.9-1	0.655 (29)	-	0 (0)	-

5.4 Discussion

5.4.1 Comparison of traditional single task and multi-task models

In this study, a thorough comparison between traditional single task and multi-task methods was conducted on two different in vitro toxicity datasets using compound-based splits. The study included RF, XGB and DNN as single task models and multi-task DNNs, FN and Macau as multi-task models. Across both datasets, no discernible difference in performance was found between single task and multi-task methods in terms of MCC. On the Tox21 dataset, both multi-task DNN and XGB-FN slightly outperformed XGB as best single task approach. However, the opposite was found on the Ames dataset. The small differences in performance between the methods suggest that neither single task nor multi-task models seem to be generally superior.

An insightful feature of this study was the analysis of model performance across different runs of a model with different random initialisations and otherwise identical parameters. It was revealed that stochastic effects may have a large influence on the performance of a single model. For some of the assays and methods the variance of performance was larger than for others. Assays with relatively few data points or strongly imbalanced assays were found to have comparatively high variances. This can be explained by the fact that seemingly small variations in predictions made by a model can have quite large impacts on the numeric values obtained for MCC which is sensitive to correct classifications. Notably, variances were smaller when models were evaluated using the ranking-sensitive ROC-AUC as metric as can be seen in Tables B1+B2 (Appendix B). Concerning different ML

algorithms, DNNs (both single task and multi-task) were found to have relatively large variance across different assays. This may be due to the number of stochastic processes involved in their training. These include the random initialisation of network parameters, the composition of batches of data points for updating the network parameters as well as the random selection of neurons for dropout. By considering the range of scores a model achieves, a more robust comparison between different models was possible. Limiting the analysis to a single score would lead to the occurrence of seemingly large difference in performance between models that are the result of pure chance, which could be misinterpreted as meaningful differences. Furthermore, this strategy helps to put the observed differences between models into context. In this case, the observed differences between single task and multi-task models are not significant when considering the variances of the models' performances on the single assays.

Multi-task DNNs performed better than single task DNNs on both datasets, but worse than XGB as best single task model on the Ames dataset. There may be potential to further improve the performance of multi-task DNN models. For instance, unrelated tasks may impede the learning of other tasks, as reported by Xu et al. (Xu et al., 2017). The prediction of completely unrelated tasks may require the network to learn fundamentally different representations in the hidden layers. As the training of the multi-task DNN is done by adjusting the weights and biases averaged across all tasks, conflicts between different tasks may hinder the overall success of learning. Hence, a careful examination of which assays should be included in the multi-task DNN might improve the performance.

FNs based on XGB models achieved the highest scores on the Tox21 dataset. However, as shown in Table 5-9, the FN models performed slightly worse than their single task counterpart in most other cases. Varnek et al. reported a superior performance of FN models over single task models (Varnek et al., 2009), however, the datasets investigated in that study were very small in size (tens of compounds) and therefore it is difficult to compare the results across the studies. An explanation for the lower performance in this study may be the inclusion of auxiliary assays that are not well predicted in the first step by the single task models. The presence of wrong predictions in the auxiliary task may propagate into wrong predictions for the target assay. This could be addressed by careful examination which assays should be included in the FN model, similar to the multi-task DNNs. A clear improvement in performance was observed, when experimentally determined toxicity labels for test compounds were used as input to the Feature Net models. Experimentally determined labels indicate the true label of a compound with higher confidence than predictions do, which may explain the surge in performance. These types of models are considered as imputation models and are discussed further below.

As other matrix factorisation techniques, Macau is primarily suited to perform imputation tasks. Based on a sparse toxicity matrix, Macau learns latent space representations of assays and compounds. However, as the learning in Macau can include side information about compounds, such as chemical fingerprints, the method can embed compounds without any measured toxicity labels in the same latent space and hence make predictions for compounds that have not been tested in any of the assays. This corresponds to conventional multi-task QSAR modelling. The obtained MCC scores for Macau were worse than those of single task and other multi-task techniques for both datasets. This is in contrast to the ROC-AUC scores measured (Table A5-1), where Macau actually achieved the highest average score on both datasets. An attempt to increase the MCC scores of Macau imputation models was made by adjusting the threshold for classification and this is discussed below. A similar approach might be useful for Macau as a conventional multi-task model when there are no observed toxicity labels for test compounds.

To some extent, our findings seem to contradict reports on the superiority of multi-task models over single task models in the literature. The Merck Kaggle Challenge was won by an approach largely based on multi-task DNNs (Ma et al., 2015) with the results being further investigated by Xu et al. (Xu et al., 2017). However, a direct comparison with our results is not possible due to differences in the datasets and the nature of the learning tasks which are classification here and were regression in the study by Xu et al. Nevertheless, even though on average multi-task DNNs outperformed single task DNNs in that study, the differences between the two approaches were also small. Mayr et al. found multi-task DNNs to outperform several single task algorithms, based on average performance, on the Tox21 dataset (Mayr et al., 2016), yet the differences were small and comparable to our findings (Table 5-2). Moreover, the selection of the metric for model evaluation may influence the conclusions. When using ROC-AUC instead of MCC as metric, both multi-task DNN and Macau outperformed the single task approaches on both the Ames and Tox21 dataset. For FN models using solely predicted activities as features in the test set, no consistent improvements over single task models (for MCC and ROC-AUC as metric) were found, which is in contrast to previous reports on FNs (Sosnin et al., 2019; Varnek et al., 2009) and on the related approach using predicted bioactivity profiles (Norinder et al., 2020).

5.4.2 Comparison of single task and multitask imputation models

For both the Ames and the Tox21 dataset, the best single task imputation model was outperformed by multi-task imputation models across all the assays. Numerically, the increases in MCC scores over single task models were larger for the Ames dataset than for the Tox21 dataset. The best three imputation methods (Macau, XGB-FN, multi-task DNN) achieved virtually the same average MCC score

on the Ames dataset. RF-FN and DNN-FN performed somewhat worse, but were still clearly better than the single task models. On the Tox21 dataset, XGB-FN was the best method followed by multi-task DNN. The other FN methods also performed better than the single task models. Macau achieved lower MCC scores than the single task models, which is discussed below.

Among the different FN methods, those composed of XGB classifiers outperformed RF and DNN on both datasets by a considerable margin, suggesting that XGB as a base classifier might be particularly suited to form successful FN models. An advantage of FN models is the simplicity of implementation since they are based on existing single task QSAR architectures applied to multi target datasets. A limitation for very large datasets (i.e. many assays) might be the computational cost of FN models, as two models (two steps) have to be trained for each assay in the dataset. This would be especially time consuming if a hyperparameter optimisation step is carried out for each assay separately. However, this can be avoided if a set of hyperparameters suitable for a wide range of assays is selected. The FN approach as used in this study is somewhat reminiscent of the pQSAR approach (Martin et al., 2019). Both approaches consist of two steps with the first step filling gaps in the dataset using single task models. The second step of the pQSAR model uses only the assay labels (known or predicted) as features rather than combining them with chemical descriptors.

The multi-task DNN, used as an imputation method, yielded competitive MCC scores on both datasets under study here. In contrast to a multi-task DNN in its conventional use, the network is trained on all the compounds in the dataset (provided there is at least one label for the compound in the training set). Therefore, the whole range of compound structures can be accounted for, and all available labels are used to learn meaningful internal representations of the compounds. An advantage of this technique compared to FNs is that only a single model needs to be trained for the whole dataset, which is computationally cheap. As discussed in the previous section, a careful examination of which assays should be included in the network might lead to even better performances, but would require additional effort. While the multi-task DNNs achieved high scores, the variance in performance between different runs of a model was relatively large. This behaviour is undesirable as a single run may lead to a poorly performing model. This could be addressed by using ensembles of multiple runs, which in turn would increase the computational cost of the method.

The Macau method achieved the highest average MCC score on the Ames dataset, but a poor average MCC score on the Tox21 dataset. This contrast may be attributed to an inappropriate classification threshold, as shown for the assay p53 (Figure 5-17, Table 5-11). Most of the assays in the Tox21 dataset are strongly imbalanced leading to a low number of compounds being predicted as toxic. Unlike for the other methods, the imbalance was not countered by assigning different weights to the

classes during learning in the Macau method. As could be shown for p53, adjusting the classification threshold can lead to higher MCC scores, which are comparable to the other imputation techniques. The classification threshold then needs to be treated as an additional hyperparameter and optimised prior to predicting the test set.

Previous studies on imputation for bioactivity prediction in the literature (Alchemite and pQSAR) were based on regression. This study represents the first one to investigate classification tasks. It was shown that for classification tasks imputation models provide a substantial benefit in performance over single task and multi-task models. This is in agreement with the findings of previous studies on regression tasks (Irwin, Mahmoud, et al., 2020; Martin et al., 2019; Whitehead et al., 2019). All imputation techniques used in this study could be adapted to perform regression tasks. However, their suitability for datasets of larger dimensions (thousands of assays) would need to be tested. Intriguingly, Alchemite and pQSAR, two conceptually different imputation techniques, achieved approximately the same average score on the Novartis benchmark set (Irwin, Mahmoud, et al., 2020). This is in concordance with findings from this study that fundamentally different imputation techniques may result in models of comparable performance.

Having demonstrated the superior performance of imputation, a series of investigations was carried out that aimed to characterise the factors that contributed to the improvements. Similar detailed investigations have not been reported in the literature. The investigations explored the effects of chemical similarity, amount of data labels and relatedness between toxicity assays.

For standard QSAR models, a key determinant for a model's performance is chemical similarity between the compounds used for training and those on which predictions are made (Sheridan et al., 2004). For chemicals very dissimilar to the compounds in the training set, a QSAR model typically struggles to make reliable predictions. The concept of an AD for a model accounts for this limitation (Mathea et al., 2016). This general trend was also observed for the datasets under study in this project for the single task methods. While the multi-task imputation models achieved higher scores than single task imputation models on all bins of chemical similarity in the datasets, the numerical increase in MCC score was higher for less similar compounds. This effect was stronger on the Ames dataset, but was also observed on the Tox21 dataset for assays for which a single task QSAR model performed particularly poorly on chemically dissimilar compounds. It can be concluded that imputation models not only perform better than single task models, but they may also increase the chemical space for which a model makes reliable predictions and hence possess a wider applicability domain.

In a wide sense, imputation means the process of filling gaps in a sparse dataset by leveraging patterns present in the dataset. Intuitively, this process should be easier the more complete a dataset is, as

more information is available to be exploited. In this study, the role of sparsity in imputation models was investigated by dividing the test sets into bins depending on the number of toxicity labels that were in the training set for each compound. Unsurprisingly, it was found that the imputation models outperformed the single task models by a wider margin for compounds with many toxicity labels in the training set. Nevertheless, both multi-task DNN and Macau outperformed XGB models on the Ames dataset for compounds with no or just one toxicity label present in the training set, indicating that very little information may be sufficient to observe improvement over single task QSAR models. The experiments on pairwise FN models confirmed that knowing the label of just a second toxicity assay can improve the predictions substantially, although this depends on which assay is used as auxiliary assay. In conclusion, imputation models perform better the more information is available, but a small amount of information can be enough for an imputation model to be useful.

A consideration of high practical relevance is which specific toxicity assays should be included in an imputation model to obtain the best results. The extent to which a single additional assay could contribute to the success of FN models was investigated by training pairwise FN models and LOAO FN models. The Ames dataset contains pairs of assays measured in the same bacteria strain (with or without metabolic activation) and the respective paired assay seemed to have the largest contribution on the FN model of a given assay. Intuitively, this makes sense as these pairs are expected to be closely related and hence knowing the outcome of one should be useful to predict the other. The Tox21 dataset contains two pairs of assays that measure the same target in slightly different test systems (NR-AR/NR-AR-LBD and NR-ER/NR-ER-LBD) and hence these assays are also closely related. With the exception of the pair NR-AR-LBD/NR-AR (the first being the target assay), those paired assays as expected provided the strongest benefit. Nonetheless, some assays that are not obviously related still contributed to the success of the imputation model.

Moreover, the relation between each pair of assays was measured by computing the MI-entropy ratio. This metric expresses how much of the total information contained in the target assay is also contained in the auxiliary assay. This metric was found to be moderately correlated to the contribution of single assays to the success of the full FN models measured as the benefit of pairwise FN models over single task models or the loss of LOAO FN models compared to full FN models. While the overall correlation was not very high, the largest contributions of single assays were found for closely related assays, consistent with the initial findings described above. There are several reasons why a higher correlation was not observed. Firstly, the number of training labels is different for each assay, which biases the correlation as a higher number of training labels for an assay is generally associated with a better performing model. Secondly, the obtained improvement over a single task model is inherently linked to the performance of the single task model. The maximal achievable performance of a model

for an assay is limited by the experimental uncertainty (Sheridan et al., 2020) and in cases where a single task model performs very well (e.g. TA1537) only small numerical increases may be achievable, even if an imputation model with a closely related assay (TA1537_S9) is used. More generally, it seems that a large increase in MCC score is easier to achieve for an assay where the single task model performs poorly. Despite the moderate correlation values, measuring the relatedness between assays as done in this project can provide useful insight on which assays are expected to be strong contributors to the overall success of an imputation model. In a practical setting, this knowledge can guide which in vitro toxicity assays are particularly useful to measure, as these measurements may be valuable to predict the outcomes for related toxicity assays using an imputation model.

In summary, the comparison between multi-task imputation models and single task imputation models demonstrated a clear benefit in performance for multi-task models. Additional studies provided understanding of which predictions in the dataset benefitted most from the imputation approaches and estimated the contribution of single assays to the success of the imputation models. This knowledge is useful to develop a suitable strategy for other imputation tasks and is expected to be applicable in other chemical datasets for bioactivity rather than toxicity.

5.5 Conclusion

This study found little differences in performance between traditional single task and multi-task QSAR models, whereas multi-task imputation models clearly outperformed single task imputation models. It must be stated that multi-task imputation uses experimental information about toxicity endpoints for test compounds, whereas single task imputation models do not. This is likely the reason why these models can achieve much better performance scores.

Thus far, the imputation models have only been tested on two multi-target in vitro toxicity datasets of relatively small size (12 assays). To further investigate the usefulness of multi-task imputation techniques, a follow-up study on a larger and more diverse dataset is presented in the next chapter.

Chapter 6 Imputation on a large-scale toxicity dataset

6.1 Introduction

In the previous chapter, multi-task and imputation models were studied on two different toxicity datasets, each consisting of 12 assays. It was found that multi-task approaches for imputation clearly outperformed single task approaches. However, datasets of in vitro bioactivity and toxicity data are often of large scale (hundreds or thousands of assays) and the suitability of imputation approaches for these datasets needs to be tested. For that purpose, the ToxCast dataset (Richard et al., 2016) was selected as an example for a large-scale toxicity dataset. After the different imputation techniques introduced in the previous chapter had been tested on the ToxCast dataset, additional experiments to better understand the behaviour of the models were conducted. The GHOST technique (Esposito et al., 2021) was used in an attempt to obtain better performing models by shifting the classification threshold. Then, the sparsity of the training set was artificially increased to investigate the impact of sparsity on the models' performance. Finally, the usefulness of information theory metrics for the selection of auxiliary assays was studied. Those experiments aimed to enhance the understanding of imputation models by examining their performance under various circumstances.

6.2 Methodology

6.2.1 Dataset

The ToxCast dataset (Richard et al., 2016) was downloaded from the MoleculeNet platform where it is provided with binary labels (Z. Wu et al., 2018). The ToxCast dataset represents a large-scale in vitro toxicity dataset, containing 8615 compounds and 617 assays (reduced to 7787 compounds and 416 assays for this study after standardisation and filtering steps were applied, see below). The dataset comprises a wide range of in vitro toxicity endpoints including receptor interaction, enzyme inhibition, developmental defects and cell viability. Notably, it includes the assays from the Tox21 dataset.

The ToxCast dataset is sparser than the datasets studied in the previous chapter with a completeness of 35.5%. Large differences exist in data completeness between the assays in the dataset with values ranging from 1.4% to 92.1% complete. Overall, 6.9% of the labels are positive (i.e. active), with the percentages for individual assays ranging between 0.9% and 82.4%.

Compounds were provided as SMILES strings in the dataset. Those were standardised and aggregated using the same steps as described in section 5.2.2. To enable a robust model evaluation, only assays with at least 50 toxic and 50 non-toxic labels were kept resulting in a dataset of 7787 compounds and 416 assays.

Since the goal of this study was to better understand the benefits of multi-task imputation models, only assay-based splits (80/20) were applied to the ToxCast dataset. Due to the low number of overall labels for some of the assays, the splitting was done in stratified manner (according to toxicity labels).

6.2.2 Model training and evaluation

As in Chapter 5, Morgan fingerprints of radius 2, hashed to 2048 bits were used as chemical descriptors. The algorithms used in this study were XGB for single task imputation and XGB-FN, multi-task DNN and Macau for multi-task imputation. Due to the large number of assays, no specific hyperparameter optimisation was performed. Instead, hyperparameter values frequently selected in the optimisation procedure in the previous chapter were manually selected. The selected hyperparameters are reported in Appendix C (Tables C1, C2 and C3).

Due to the dataset's larger scale, evaluation of each model was limited to one model instance resulting from a single random seed. Models were evaluated on the test set using MCC and ROC-AUC as metrics.

6.2.3 GHOST methodology

GHOST (generalized threshold shifting procedure) was proposed as a method to automatically find an ideal decision threshold for classification models using merely training instances (Esposito et al., 2021). In GHOST, a classifier is trained using all training examples and probability scores are determined for each training instance. N bootstrap samples are then drawn from all training instances and the optimal threshold is found for each sample by trying a range of different thresholds. The chosen decision threshold is taken as the median of the individual optimal thresholds. Two different approaches were used to find the optimal threshold for a bootstrap sample. In this project, the method originally described elsewhere was used (Song et al., 2014), which determines the point of the ROC curve closest to the upper left corner (0,1). The GHOST method was applied on the models trained on the ToxCast dataset and, as in the original paper, the thresholds considered were: 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55. In the multi-task settings, thresholds were optimized for each task separately.

6.2.4 Experiments on sparsity

The effect of overall dataset sparsity on single task and multi-task imputation model performance was investigated. In particular, sparsity in the training set was artificially increased by randomly sampling 1000 labels per assay for model training with all other labels removed. For assays with fewer labels in the training set, all labels were kept. The overall completeness of the training set was reduced from 29.4% to 10.9% in this scheme. Models were evaluated on the same test set as for the models trained using all available labels to enable comparison of model performances.

6.2.5 Impact of assay relatedness on model performance

It was investigated to what extent the relatedness between target assay and auxiliary assays can explain increased performance of multi-task performance compared to XGB as a single task model. Using the MI-entropy ratio (see section 5.2.6.3) to evaluate assay relatedness, for each assay the average relatedness to all other assays as well as the average relatedness to the 10 most similar assays was computed. Finally, the correlation to the increase in performance over XGB was computed for each multi-task technique.

The usefulness of the MI-entropy ratio to select auxiliary assays from a large dataset was tested using one exemplary target assay from the ToxCast dataset. This was the assay 'TOX21-AromataseInhibition' as it showed a large increase in performance when using multi-task imputation models compared to single task models. In addition, the assay possesses a relatively large number of experimental labels in the test set (1431 in total with 221 of these toxic), which enables a robust evaluation of model performance. Increasing numbers of auxiliary assays (1, 3, 5, 10, 20) were selected either randomly in an additive approach (i.e., the 3 sampled auxiliary assays include the assay that was sampled for 1 auxiliary assay, and so on) or according to the MI-entropy ratio in descending order. For this experiment, 20 different random seeds were used during model training for a more robust evaluation.

6.3 Results

6.3.1 Comparison of single task and multi-task imputation models

Initially, overall performance of the different techniques was compared using MCC and ROC-AUC as metrics. Due to the large number of assays, a comparison of all techniques on a per-assay basis is not straightforward to visualise. Instead, scores for individual assays across the dataset were sorted in descending order for each technique and are visualised as line plots in Figure 6-1 (A: MCC, B: ROC-AUC). A similar visualisation was previously done when evaluating the pQSAR technique (Martin et al., 2019). Table 6-1 reports maximal, median and minimal score for each technique.

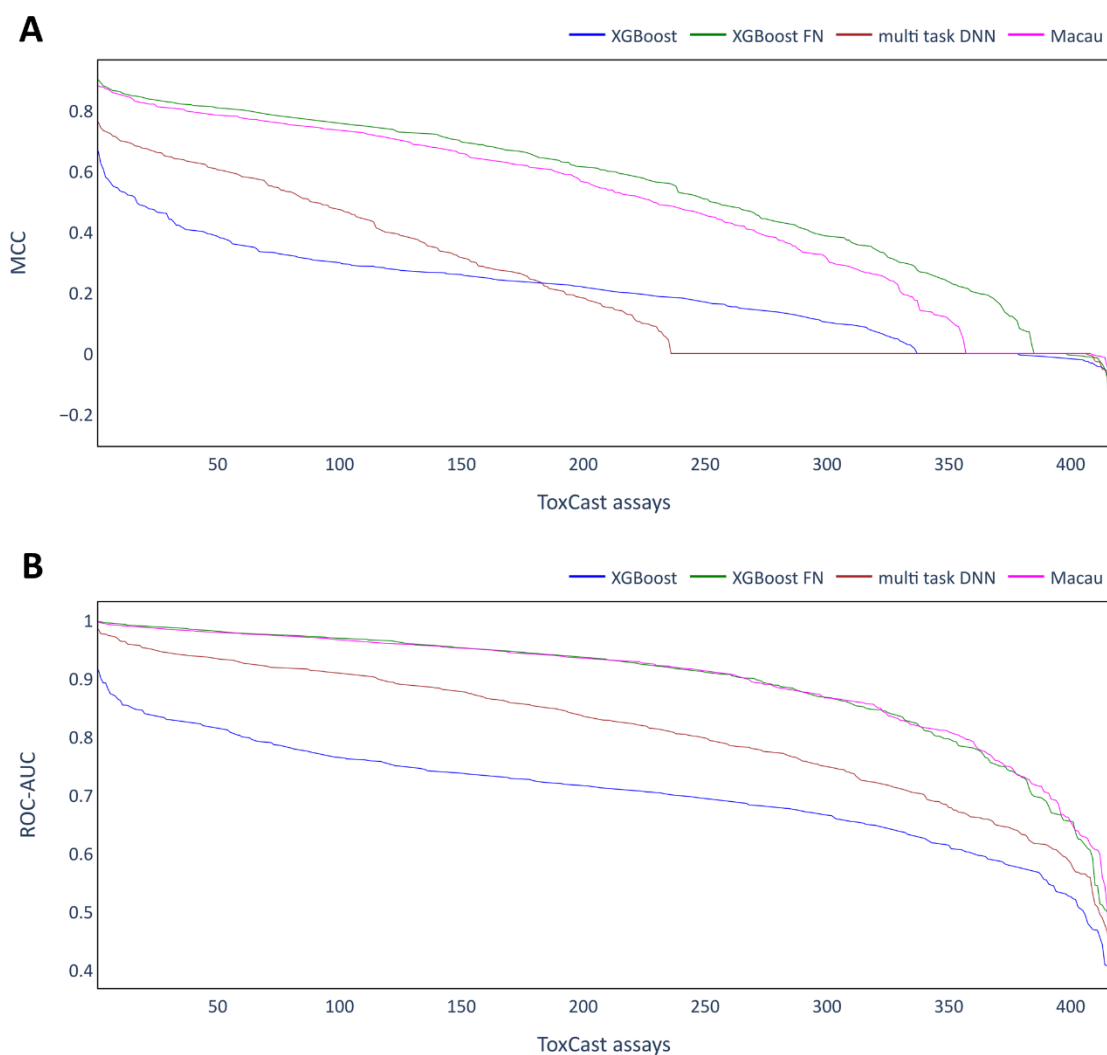


Figure 6-1 Performance of imputation models on the ToxCast dataset. A: MCC, B: ROC-AUC. Scores for individual assays were sorted in decreasing order for each technique.

Table 6-1 Overview of performances for different techniques on the ToxCast dataset. Shown are max, median and min MCC and ROC-AUC scores across the assays in the ToxCast dataset.

	XGB	XGB-FN	MT-DNN	Macau
MCC-max	0.667	0.901	0.762	0.881
MCC-median	0.207	0.604	0.155	0.544
MCC-min	-0.091	-0.182	-0.241	-0.058
ROC-AUC-max	0.915	0.998	0.985	0.996
ROC-AUC-median	0.712	0.934	0.829	0.933
ROC-AUC-min	0.402	0.481	0.434	0.412

A wide range of MCC scores was obtained for the different models and the different assays. XGB as a single task approach was clearly outperformed by XGB-FN and Macau models, confirming the superiority of multi-task imputation approaches observed in the previous chapter. For multi-task DNN, quite high MCC scores were found for some of the assays (larger max performance than XGB in Table 1), yet also very low MCC scores around zero were found for many of the assays. Also, when considering ROC-AUC scores, XGB was clearly outperformed by XGB-FN and Macau which achieved very similar scores. The ROC-AUC scores for multi-task DNN were between those for XGB and those for the other two multi-task imputation approaches.

It is notable that MCC scores of 0 or even below were observed for some of the assays for all of the modelling methods, with these occurring most frequently for multi-task DNN models where more than 150 assays had such very low scores. Very low MCC scores may be the result of an inappropriate classification threshold, as was demonstrated for the SR-p53 assay in the previous chapter (section 5.3.7). In this study, the GHOST approach was used to investigate the impact of adjusting the classification threshold on MCC scores (see following section). Nonetheless, the higher ROC-AUC scores clearly support the finding that XGB-FN and Macau were the most performant multi-task techniques on the ToxCast dataset.

6.3.2 Using GHOST to adjust classification thresholds

The GHOST methodology was applied to all techniques and assays of the ToxCast dataset in order to check whether optimising the classification threshold may lead to higher MCC scores. Shown in Figure 6-2 are line plots for all techniques, a scatter plot comparing individual assays for Macau with and without GHOST and boxplots comparing MCC score distributions for Macau with or without GHOST.

Overall, the trends are similar to when GHOST was not used with XGB-FN and Macau outperforming MT-DNN and XGB. What can be clearly seen in both the line plot and the scatter plot for Macau is that the number of assays with very low scores of around zero was considerably reduced. This means that multi-task DNN undoubtedly outperformed XGB, yet it performed still worse than Macau or XGB-FN. When focussing on Macau (Figure 6-2B), it can be observed that GHOST was very effective in increasing MCC scores for assays with very low scores before (<0.2). In contrast, for assays with high scores before (>0.6), performance for the majority of assays was slightly decreased. Overall, this resulted in a 75th percentile score lower and a 25th percentile score higher compared to not using GHOST for Macau, while the median score was very similar (Figure 6-2C). In conclusion, GHOST may be useful to improve models performing poorly due to an inappropriate classification threshold, but for already good models a drop in performance may be observed. Due to the mixed results, GHOST was not used in the following experiments.

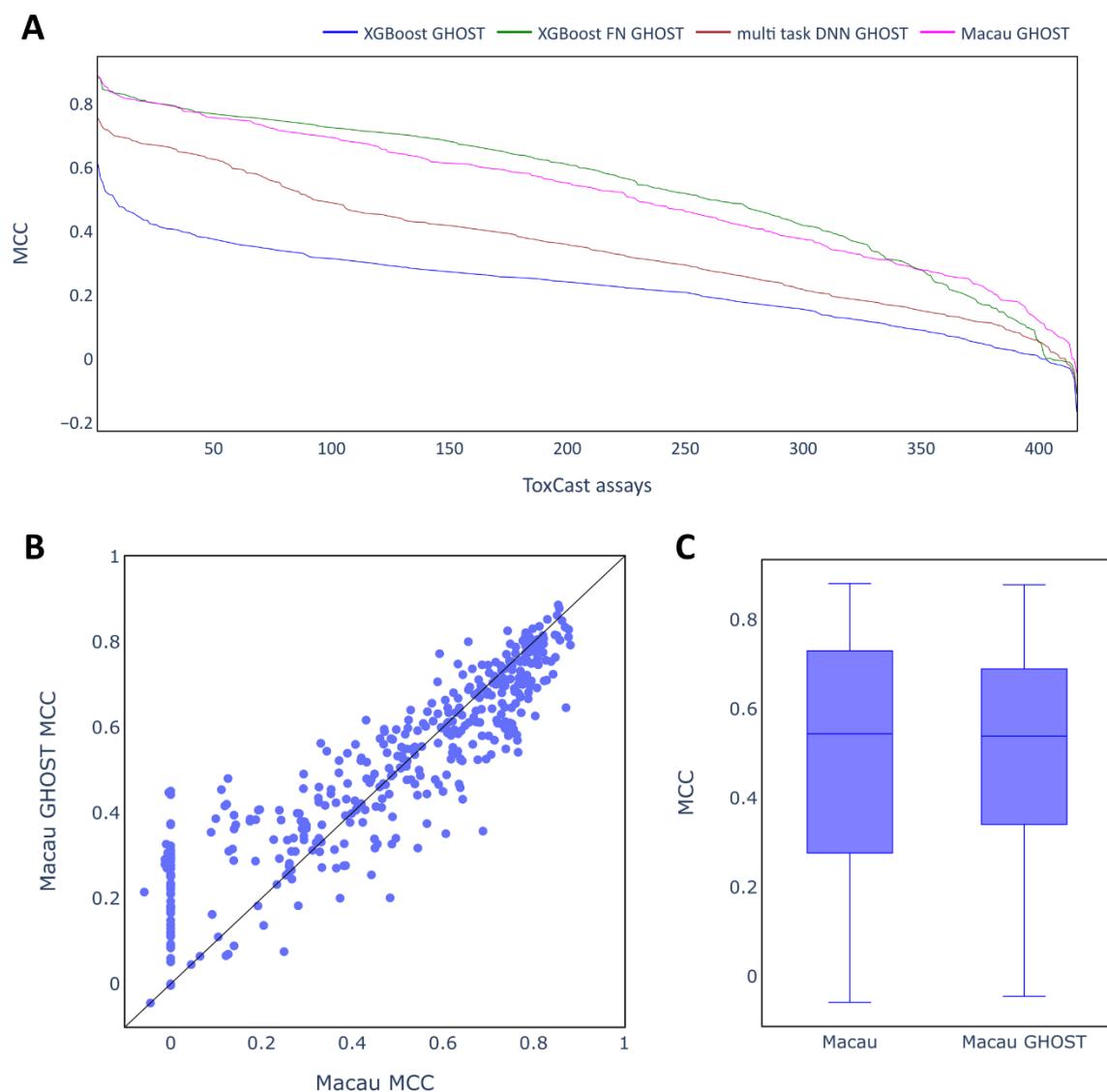


Figure 6-2 Performance of the imputation models using the GHOST approach. **A:** line plots showing performance on the test set for all algorithms as in Figure 6-1. **B:** scatter plot contrasting MCC scores of Macau model for all individual assays. **C:** box plots contrasting distribution of MCC scores for all assays.

6.3.3 Effect of sparsity on model performance

Sparsity is a key determinant for the success of imputation models. In the previous chapter, its effect has been studied by binning the test set according to the number of available auxiliary toxicity labels. It was found that multi-task imputation models are most effective for compounds where a large number of auxiliary toxicity labels is available, corresponding to low sparsity. The larger size of the ToxCast dataset enabled a more detailed study of the effect of sparsity on imputation. Labels in the ToxCast training set were removed to artificially increase sparsity in the dataset. As described above,

the number of training labels was reduced to 1000 for all assays, with no changes made for assays with fewer labels in the original dataset. Figure 6-3A shows the performance of Macau, XGB-FN and XGB models on the dataset with increased sparsity in comparison to the original models on the same test set. In Figure 6-3B and 6-3C, assay-wise performances are contrasted for XGB-FN and Macau, respectively, whereby assays with unchanged number of training labels (see above) are coloured in red.

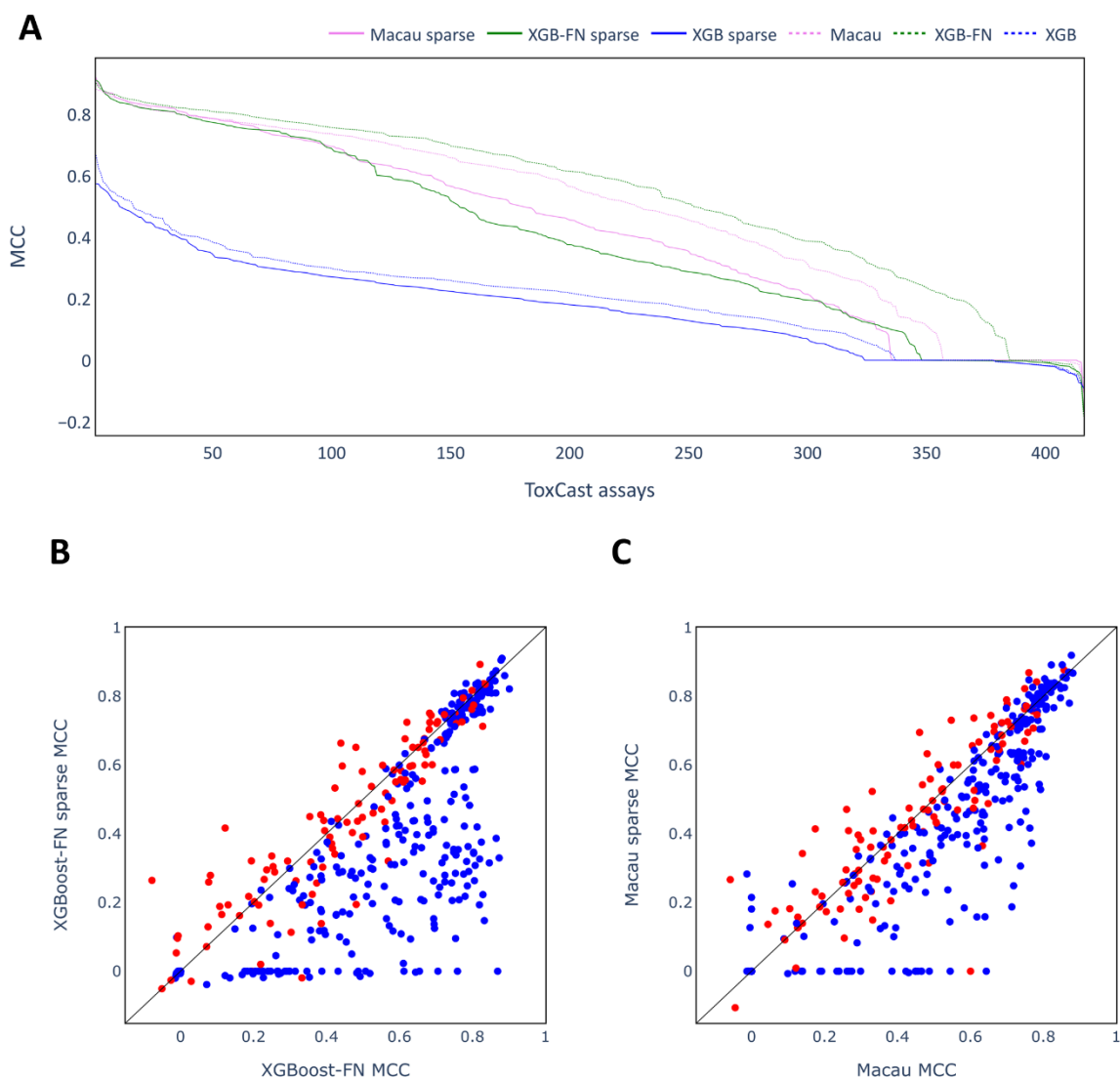


Figure 6-3 Performance of models trained on the training set with increased sparsity. **A:** line plot comparing performances of different techniques on the test set. Dotted lines indicate performances of models on original datasets (sparsity not increased). **B:** scatter plot contrasting MCC scores of XGB-FN model for all individual assays. **C:** scatter plot contrasting MCC scores of Macau model for all individual assays. Assays whose number of training points was unchanged (≤ 1000 data points originally) are coloured red, those with reduced number of training points (> 1000 data points originally) are coloured blue

The performance across the assays was decreased for both single task and multi-task imputation approaches on the data with increased sparsity, however, the multi-task approaches remain clearly superior to XGB. Macau seems more robust towards increased sparsity compared to XGB-FN, as the decrease in MCC scores is less pronounced for this technique. Decreases in performance were more pronounced for assays where training labels were removed (the blue dots in Figure 6-3B and Figure 6-3C), whereas there was less impact on the scores for assays with fewer labels in the original dataset. This may be due to the overall structure of the ToxCast dataset whereby some compounds were tested in the majority of assays (Richard et al., 2016). This means that assays with only a few labels are likely to contain mostly compounds that were tested in a large number of assays and therefore test compounds for those assays will have many experimental labels to use. Even if the overall sparsity is increased, the test compounds of these assays will still have a high number of experimental labels, so that little or no decrease in performance was observed. In particular, it may be that those auxiliary assays most closely related to the target assays (with fewer than 1000 labels) also had fewer than 1000 labels and hence the most important source of information would not have been removed. For a single assay, overall sparsity of the dataset may not be the main determinant for the success of multi-task imputation approaches, and it seems that having information from at least some related assays may be crucial instead. The effect of assay relatedness on model performance was further investigated in the following section.

6.3.4 Impact of assay relatedness on model performance

In the previous chapter, the effect of relatedness between assays was measured using pairwise and LOAO FN models. Those experiments investigated the impact of single auxiliary assays. Due to the larger number of assays, those experiments would be prohibitively expensive for the ToxCast dataset. Instead, mean MI-entropy ratios to either all auxiliary assays or to the 10 with the highest values were computed. In Figure 6-4 the correlation between the average MI-entropy ratio and increase in MCC score compared to XGB is shown for XGB-FN (all compounds: 6-4A, 10 closest assays: 6-4B) and Macau (all compounds: 6-4C, 10 closest assays: 6-4D).

Naturally, the average MI-entropy ratio for the 10 highest values will be much higher with values of up to around 0.35 compared to around 0.07 as highest average value to all auxiliary assays. Nonetheless, in all cases similar trends can be observed: a higher average MI-entropy ratio is correlated with larger increases in MCC compared to XGB. The correlation for Macau models was found to be somewhat stronger than for XGB-FN models (0.727 vs 0.649 for all assays and 0.799 vs 0.719 for the 10 highest values; Pearson correlation coefficients). Moreover, taking the average for

just the 10 highest values was more strongly correlated and hence a better indicator of how well the multi-task imputation model (which uses all auxiliary assays) will perform compared to XGB as single task model. This suggests that the auxiliary assays most closely related to the target assay play a significant role in the observed increases in model score. This hypothesis was followed up in the next experiment focussing on a single target assay.

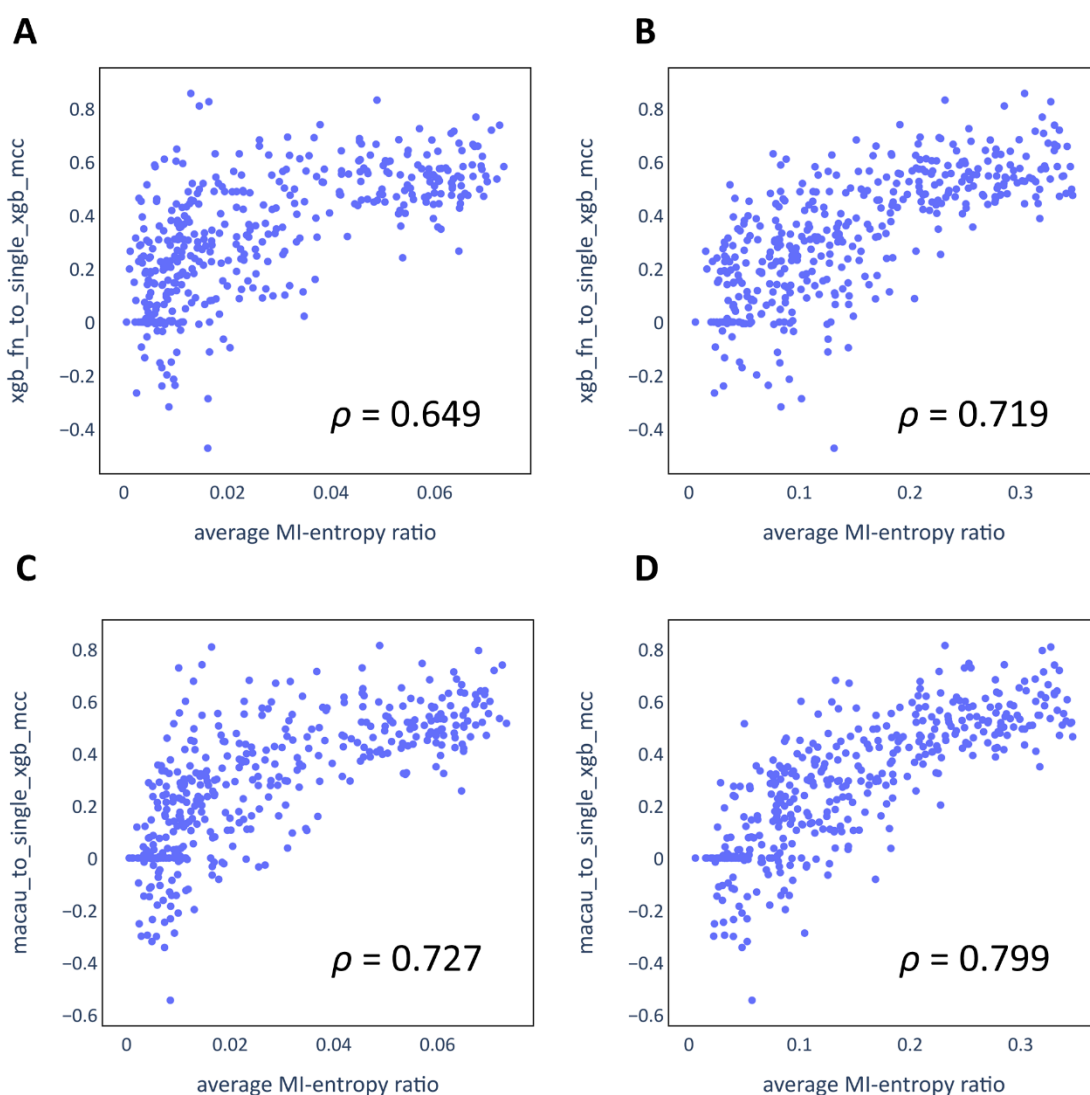


Figure 6-4 Correlation between MCC changes and mean MI-entropy ratio. The mean MI-entropy ratio was either calculated for all auxiliary assays or only the 10 highest for the respective target assay. Also shown are Pearson correlation coefficients. **A:** XGB_FN all assays, **B:** XGB_FN Top-10 assays, **C:** Macau all assays, **D:** Macau Top-10 assays.

It was investigated how well multi-task models perform when auxiliary assays to be used in the model were selected according to MI-entropy ratio. The performance was compared to the situation when an equal number of randomly selected assays was used as auxiliary assays. In Table 6-2 the selected

assays (according to MI-entropy ratio and randomly) are shown for TOX21-Aromatase-Inhibition as target assay. For the 20 assays selected in total according to this metric, the MI-entropy ranges between 0.369 and 0.256. For randomly selected assays, the values are much lower and in the range between 0.003 and 0.261 (one of the Top-20 assays was randomly selected).

The two assays most closely related to the target assays identify antagonists to the thyroid hormone receptor (TOX21_TR_LUC_GH3_Antagonist) and the androgen receptor (TOX21_AR_BLA_Antagonist_ratio), respectively. The protein of the target assay, aromatase, catalyses the conversion from androgens to estrogens. It is plausible that inhibitors of the androgen receptor might also inhibit aromatase, for which androgens are the substrate. It is somewhat surprising that the assay measuring inhibition of the thyroid receptor is more closely related to the target assay. Nonetheless, the thyroid hormone receptor is like the androgen receptor an intracellular one with typically hydrophobic ligands, which might explain their relatedness in the assay labels (Flamant et al., 2006; Matsumoto et al., 2013). Among the Top-20 assays are further ones related the intracellular receptors such as one measuring antagonism to the farnesoid X receptor (TOX21_FXR_BLA_antagonist_ratio) (Forman et al., 1995).

Table 6-2 Selected auxiliary assays for TOX21-Aromatase-Inhibition. Assays were either selected according to their MI-entropy ratio or randomly.

Assays selected according to MI-entropy ratio	MI-entropy ratio	Randomly selected assays	MI-entropy ratio
TOX21_TR_LUC_GH3_Antagonist	0.369	ATG_RARb_TRANS_dn	0.018
TOX21_AR_BLA_Antagonist_ratio	0.308	TOX21_MMP_ratio_up	0.019
BSK_SAg_CD40_down	0.295	ATG_Oct_MLP_CIS_up	0.124
BSK_SAg_CD69_down	0.289	ATG_DR4_LXR_CIS_dn	0.136
TOX21_AR_LUC_MDAKB2_Antagonist2	0.287	ATG_HSE_CIS_dn	0.003
TOX21_AR_LUC_MDAKB2_Antagonist	0.284	OT_ER_ERaEra_1440	0.007
BSK_LPS_SRB_down	0.279	ATG_Ets_CIS_dn	0.048
BSK_4H_Pselectin_down	0.276	BSK_3C_HLADR_down	0.253
BSK_3C_Proliferation_down	0.270	BSK_KF3CT_IP10_down	0.209
BSK_3C_Vis_down	0.270	BSK_hDFCGF_TIMP1_down	0.206
NCCT_HEK293T_CellTiterGLO	0.269	TOX21_HSE_BLA_agonist_ch2	0.043
BSK_SAg_SRB_down	0.267	TOX21_VDR_BLA_agonist_ch2	0.004
BSK_3C_SRB_down	0.267	NVS_ADME_hCYP19A1	0.073
TOX21_FXR_BLA_antagonist_ratio	0.267	APR_HepG2_MicrotubuleCSK_24h_dn	0.047
BSK_CASM3C_Proliferation_down	0.265	BSK_SAg_MIG_down	0.246
BSK_SAg_CD38_down	0.261	BSK_LPS_MCP1_down	0.261
BSK_LPS_MCP1_down	0.261	ATG_NRF2_ARE_CIS_up	0.121
BSK_3C_IL8_down	0.261	BSK_3C_uPAR_down	0.214
BSK_SAg_IL8_down	0.257	NHEERL_ZF_144hpf_TERATOSCORE_up	0.131
BSK_SAg_MCP1_down	0.256	TOX21_PPARD_BLA_Antagonist_ch1	0.012

For model training, 1, 3, 5, 10 or 20 auxiliary assays were selected. MCC score ranges of models are shown for XGB-FN models and Macau models in Figure 6-5A and 6-5B, respectively. As a comparison, scores of XGB and the respective full multi-task imputation model (using all auxiliary assays) were added.

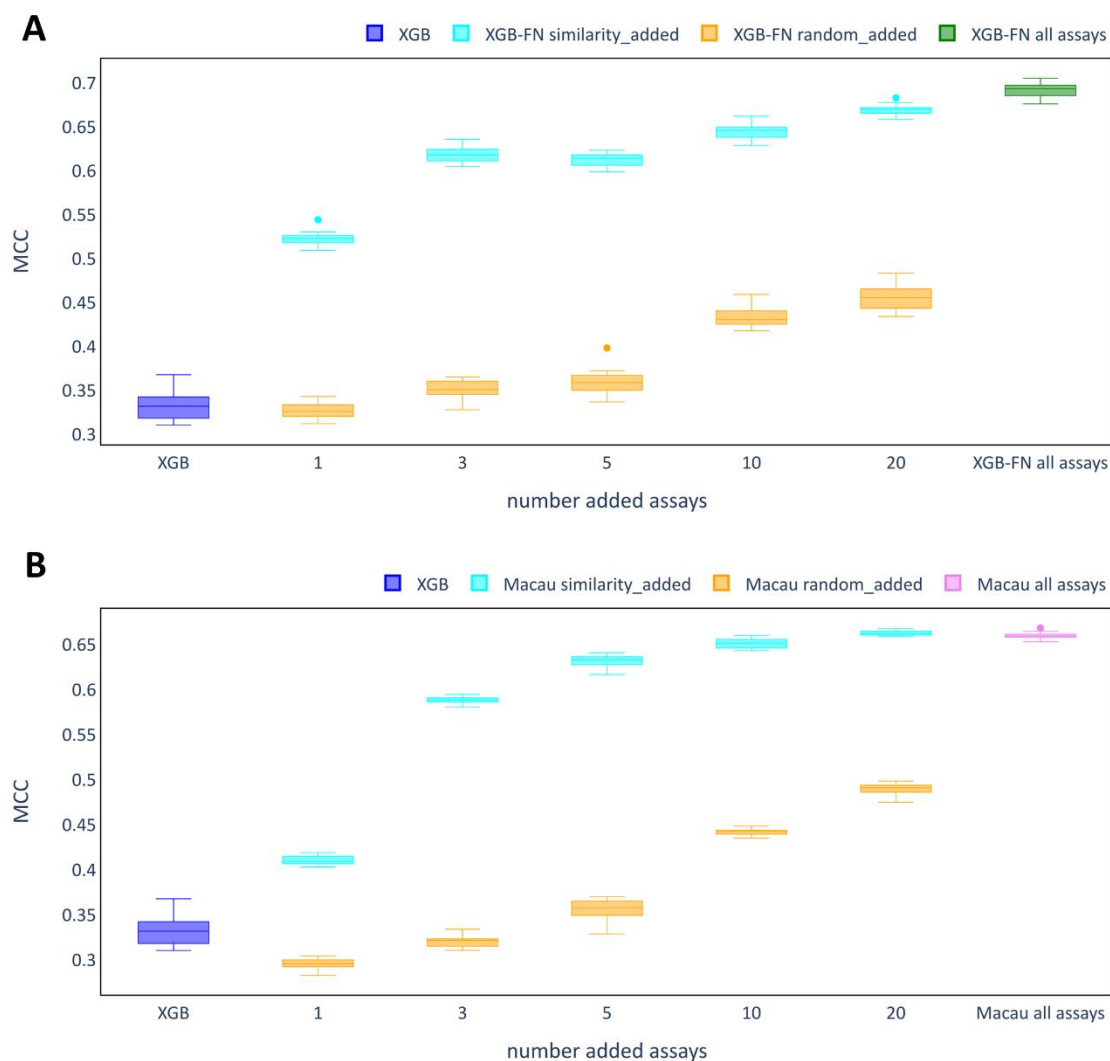


Figure 6-5 Auxiliary assay selection for the assay 'TOX21-Aromatase-Inhibition'. Auxiliary assays were selected according to the MI-entropy ratio ('similarity_added') or randomly ('random_added'). As a comparison, performance of XGB as a single task model and the respective multi-task model using all remaining auxiliary assays were included. Plotted are the MCC scores for the test set across 20 runs with different random seeds. **A:** XGB-FN models. **B:** Macau models.

For both XGB-FN and Macau, those models trained with auxiliary assays selected with the MI-entropy ratio criterion clearly outperformed those trained with randomly selected auxiliary assays. Macau models trained with the 20 most similar assays even performed slightly better than the models with all assays. For both XGB-FN and Macau, adding just one auxiliary assay provides a clear improvement over XGB models with a further strong increase after two more assays (three in total) were added. For the random assay selection, clear improvements were only observed after at least 10 assays were added. As can be seen in Table 6-2, some of the randomly selected auxiliary assays have relatively high MI-entropy ratios (>0.2). This may explain why clear improvements in model scores were also observed when assays were selected randomly.

6.4 Discussion

The findings in this chapter confirm the conclusion from the previous chapter that multi-task imputation models are superior to single task models. XGB was clearly outperformed by XGB-FN and Macau on the ToxCast dataset. Multi-task DNN outperformed XGB when using ROC-AUC as metric, although for many assays a MCC score of 0 was obtained which hinted at the decision threshold being inappropriate as the result of imbalanced training data. The GHOST methodology was therefore applied to the models in order to optimise the thresholds individually for each assay and method. The optimised multi-task DNN model outperformed the single task XGB, however, it did not perform as well as XGB-FN and Macau. Across the different techniques, the GHOST approach was successful in improving MCC scores for assays with poor scores (MCC in many cases 0) when the default threshold was inappropriate. However, for assays with high scores using the default threshold, a slight decrease in MCC score was found for some assays. It can be concluded that the GHOST approach may be helpful when applied to imputation models, yet it should not necessarily be applied to all the assays. Situations where the GHOST approach seems useful are when assays have a poor MCC score, and the ROC-AUC score indicates reasonably good performance in ranking toxic compounds higher than non-toxic ones. In particular, it may be that the GHOST approach would be successful in improving the performance of Macau on assays from the Tox21 dataset, for which Macau achieved relatively good ROC-AUC scores but poor MCC scores. For the p53 assay it was already shown in the previous chapter that a classification threshold different to 0.5 would have led to a higher MCC score. However, in that case a range of different thresholds was evaluated on the test set. In practice, only training data may be used to optimize, as it is done in GHOST (Esposito et al., 2021).

Even when GHOST was used, multi-task DNN performed worse than XGB-FN and Macau on the ToxCast dataset. This is different to the smaller datasets used in the previous chapter where multi-task DNN achieved very competitive scores. It could be that a more careful selection of hyperparameters including architecture may be required for better scores. In general, it may be challenging for a multi-task DNN to achieve competitive scores on a large number of assays with widely different numbers of training instances. A number of different weighting schemes (including weighting on task size, i.e. numbers of labels per assay) for different tasks in a traditional multi-task DNN were tested in a recent study (Humbeck et al., 2021). However, the study found at best very little improvements in comparison to unweighted training which would indicate that it may be very challenging to further improve the performance of the multi-task DNN models in the present study.

In the previous chapter, it was shown that multi-task imputation models perform better for compounds with a large number of experimental labels for auxiliary assays available, but also that

very little information may be sufficient to outperform single task models. In this chapter, the impact of artificially increasing sparsity was investigated. The completeness of the training set was considerably reduced from 29.4% to 10.9% (by only keeping 1000 labels per assay) in a conducted experiment. It was found that increased sparsity reduces performance for both single task and multi-task imputation models. Nonetheless, multi-task imputation models remained clearly superior to XGB which further supports the notion that relatively little information may be sufficient for multi-task imputation models to be successful.

Other experiments investigated the relevance of additional information by analysing the relatedness between target assay and auxiliary assays. In the previous chapter, it was shown using pairwise and LOAO FN models that some auxiliary assays may be more useful than others for a particular target assay. To some extent the relatedness evaluated using the MI-entropy ratio could explain those effects. In this chapter, it was confirmed that the relatedness between assays is a key determinant for the success of multi-task imputation models. It was shown that the benefit of a multi-task over single task models is correlated with the average MI-entropy ratio to auxiliary assays (both all and the Top-10 most related). Moreover, it was shown for an exemplary target assay how the MI-entropy ratio can be used to identify the most useful auxiliary assays.

Clearly, both the type and amount of information in the form of experimental labels for auxiliary assays determine the success of imputation models. Having many auxiliary labels and, in particular, labels of closely related assays lead to the largest improvements in model performance. The MI-entropy ratio introduced in this work provides a useful means to formalise and quantify the concept of relatedness between different toxicity assays.

6.5 Conclusion

Naturally, multi-task imputation approaches are restricted to predict toxicity for compounds where the toxicity has already been measured for other endpoints. A very common application of QSAR models is to predict toxicity of virtual compounds, i.e. compounds that were not yet synthesised and therefore no information about any toxicity endpoints is available. This situation corresponds to traditional multi-task modelling and no substantial benefit over the best single task approaches should be expected in most situations.

For toxicity predictions across different endpoints for compounds with no available toxicity data, multi-task DNNs may be the best choice. Depending on the dataset, these models might provide a slight benefit in performance over single task models. Furthermore, the use of a multi-task DNN can

save computation time compared to the training of single task QSAR models for each individual endpoint. Generally, a prerequisite of deep learning methods is the availability of sufficient data (usually several thousand compounds for QSAR models). For the datasets under study, XGB was found to be the best performing single task QSAR model and it is also a time efficient algorithm, which makes it a good default choice for single task QSAR models. XGB may have many hyperparameters that need to be selected, but a previous study provides a good starting point for QSAR models (Sheridan et al., 2016).

Multi-task imputation models should be considered if toxicity predictions are to be made for compounds which have already been characterised in some toxicity tests. This may be the case when, for instance, a drug candidate was already tested in some in vitro toxicity assays or if additional toxicity endpoints need to be evaluated for industrial chemicals, as governed under the REACH regulation. As clearly shown in the present study, the inclusion of data for other toxicity endpoints may yield superior QSAR models. Endpoints closely related to the target endpoint are most valuable. Related endpoints can be identified using a toxicologist's intuition or objective criteria such as correlation based on numerical data or metrics like the MI-entropy ratio for categorical data. However, even endpoints that are not obviously related to each other may still prove useful in imputation models.

A well-defined applicability domain is essential to use a QSAR model to support a regulatory decision. The present study suggests that imputation models can make reliable predictions for a wider chemical space. Hence, imputation models may be particularly useful in situations where the applicability domain limits the use of QSAR models in practice. In conclusion, imputation models have the potential to improve the performance of QSAR model for toxicity prediction used in practice and to extend the range of situation where their use would be justified to replace other testing methods.

Chapter 7 Interpretation of neural networks for QSAR modelling

Interpretability of QSAR models is an important issue, especially when they are used to make high-stakes decisions in the context of evaluating the safety of chemicals. Being able to understand why a model made a particular decision increases the confidence in the predictions and thus the acceptance of the model tremendously. This point is also reflected in the fifth OECD principle for the use of QSAR models in the context of regulatory decision making which states that models should be interpretable (OECD, 2004). Neural networks (especially DNNs) have been found to be a very well-performing algorithm for QSAR modelling. However, their complex structure (i.e. composition of numerous non-linear functions) precludes a straightforward and intuitive way of interpreting their models, which is why they are often referred to as black boxes (Loyola-Gonzalez, 2019).

Several approaches to achieve interpretability for DNNs and other ML algorithms in the context of QSAR modelling have been described. One possibility is to assign importance to input features. This can be done either globally (i.e. the importance of features for the model's overall performance) or locally (i.e. the importance of features for the model's individual predictions) (Jiménez-Luna et al., 2020). Methods that determine local feature importance are also called attribution methods. Some attribution methods have been specifically developed for neural networks (gradient-based), while others can be applied to any ML technique (e.g. LIME, SHAP, perturbation methods). Other approaches attempt to interpret neural networks by exploring the chemical meaning of neurons in hidden layer of the networks.

7.1 Feature attribution methods

Local importance can be determined by approximating the complex model f with a simple, interpretable one g , as done in the LIME (Local Interpretable Model-agnostic Explanations) approach (Ribeiro et al., 2016). The explanation model in LIME has the form:

$$g(x) = \phi_0 + \sum_{i=0}^M \phi_i x_i$$

Where x is a binary vector representation of the data instance to be explained of dimensionality M and ϕ_i is the coefficient indicating the importance of the i^{th} element of x . To train the explanation

model, artificial instances are generated which represent sampled subsets of the bits set on the vector of the instance x . These artificial instances are labelled using the original model f . The explanation model is of linear form and during training higher weight is given to artificial instances more similar to x according to some similarity measure. The LIME approach has not been widely used with QSAR modelling thus far, however, it was used recently with the aim to identify relevant features in a k-NN model trained to predict activity of compounds against *Pseudomonas aeruginosa* (Bugeac et al., 2021). However, the authors did not demonstrate the validity of the LIME approach in combination with QSAR models (i.e. showing that the method identifies features known to be relevant for a certain task).

The SHAP (Shapley Additive Explanations) method (Lundberg & Lee, 2017) was first applied to QSAR models by Rodriguez-Perez and Bajorath (Rodríguez-Pérez & Bajorath, 2020a) and can be considered an extension to LIME. This method combines a linear feature attribution model with the concept of Shapley values, which, originating in game theory (Shapley, 1953), allocate contributions to participants of a collaborative game. Applied to model interpretability, this concept can be used to measure the contribution of individual features to a model's prediction of an instance. Shapley values measure the importance ϕ of a feature i to a prediction as the change in the model's prediction when the feature is added, averaged over all permutations of feature subsets according to the formula

$$\phi_i = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|! (|N| - |S| - 1)! [f(S \cup \{i\}) - f(S)]$$

Where N is the number of features and S are the feature subsets not containing i . This approach is prohibitively expensive, as it requires retraining the model with all possible feature subsets. However, by sampling artificial instances, a local explanation model similar to LIME can be trained where the feature attribution values represent approximated Shapley values. SHAP is a model-agnostic approach and hence can be used to explain any supervised machine learning model based on a binary input vector.

Rodriguez-Perez and Bajorath used SHAP in combination with a variety of QSAR models. They tested the approach with different ML algorithms (RF, SVM, DNN), and two different chemical fingerprints (ECFP4 and MACCS) on 10 bioactivity classification datasets retrieved from ChEMBL. The method outputs a SHAP value for each feature of an instance expressed as a positive or negative contribution towards the prediction. The validity of the concept was confirmed in an experiment where descriptors with high SHAP values were removed prior to model training. This led to a strong decrease of global model performance, while the removal of randomly selected features resulted merely in a slight decrease of model performance. By mapping the top-ranked features back to the instance's molecular

structure, atoms contributing strongly to a prediction can be highlighted leading to a visualisation of the model interpretation.

A different method specific for interpreting predictions made by neural networks is called integrated gradients (IG) (Sundararajan et al., 2017). IG belongs to the gradient-based methods (Ancona et al., 2018) which assign importance to an input feature by determining its gradient with respect to the model output (i.e., the partial derivative for the feature value of a given instance). Gradient-based methods can only be applied to differentiable models (DNNs are differentiable). In the IG method, the gradient of each feature is integrated along a straight line between an input vector x and a baseline vector x' (in the case of chemical fingerprints the baseline vector is when all bits are set to zero). The straight path between x' and x can be described with the term

$$x' + \beta \times (x - x')$$

Where β takes values in the range [0,1]. The attribution α for a feature i of an instance x is computed by

$$a_i(x) = (x_i - x'_i) \int_{\beta=0}^1 \frac{\partial F(x' + \beta \times (x - x'))}{\partial x_i} d\beta$$

Where $F()$ is the neural network model. In practice, the integral can be approximated by replacing it with a sum of partial derivatives evaluated at m equally spaced steps on the path from x' to x as follows:

$$a_i(x) \approx (x_i - x'_i) \sum_{k=1}^m \frac{\partial F(x' + \frac{k}{m}x(x - x'))}{\partial x_i} \times \frac{1}{m}$$

A useful property of IG is that the sum of all attributions for a given instance x is equal to the difference in the model's output for x and the baseline x' . IG has been applied to QSAR models together with Morgan fingerprints (Preuer et al., 2019). Similar as for the SHAP approach, positive and negative contributions of fingerprint bits can be mapped back to atoms of a test compound. To visualise the interpretation of a prediction, atoms with high positive or negative attribution can be highlighted with colours. IG has also been applied to GCNs (Jiménez-Luna et al., 2021).

Other commonly used approaches are based on manipulating input instances (also known as perturbation-based) and these are model-agnostic (i.e. applicable to any model). For instance, the method 'similarity maps' (Riniker & Landrum, 2013) determines the importance of atoms to a model prediction by removing fingerprint bits associated with the respective atom and comparing the prediction for the modified fingerprint to the reference fingerprint belonging to the unmodified

compound. A similar approach extends the idea to determine contributions of arbitrary fragments (P. G. Polishchuk et al., 2013). To determine the contribution of the selected fragment to a prediction, the prediction for the original compound is compared to the prediction for the modified compound after removal of the fragment. The difference in prediction then is attributed to the removed fragment. This can also be applied to combinations of distant fragments to test for synergistic effects. The validity of both methods was evaluated by focussing on compounds where the importance of substructures to the observed activity is known. In another study functional groups were added to existing compounds and the difference in predicted activity was used to estimate the effect of modifications to the compound (Wenzel et al., 2019). This approach is intended to aid medicinal chemist in modifying lead compounds to obtain desired molecular properties.

In most of the above mentioned studies, the proposed method was validated by reporting examples where the provided explanation matches the true cause of activity. However, no thorough evaluation across a complete dataset was conducted. Sheridan used a perturbation-based approach to investigate the robustness of attribution methods with regard to different ML algorithms (including DNNs, RFs and XGBoost) and chemical descriptors (including ECFP4, APDP, DRUGBITS as a fingerprint describing common groups found in drugs) on a range of datasets (Sheridan, 2019). It was observed that the determined atom attributions are very sensitive to the selected method-descriptor pair, and more sensitive than the corresponding predictions of molecules' activities. Moreover, by using data sets where the theoretical contribution is fully known (e.g. the 'activity' of a compound was defined as the to the number of negative charges in the compound), it was shown that not all models yield the expected atom contributions. These findings demonstrate that the atom colouring is not as robust as was expected and Sheridan concluded that more studies are required to validate the suitability of a particular attribution method on a range of datasets and chemical descriptors. Recently, the use of benchmark datasets for interpretability of QSAR models was proposed (Matveieva & Polishchuk, 2021). The publication contains a range of synthetic datasets for regression and classification tasks where the 'activity' of compounds is fully determined by the presence of certain atoms, functional groups or pharmacophores. One main conclusion from that study was that high prediction accuracy is a prerequisite to achieve good model explanations, yet a good prediction does not guarantee good explanations. Making use of benchmark datasets seems like a sensible approach to advance the field by identifying the most useful techniques.

7.2 Interpretation of hidden layers

While feature attribution methods can be useful to understand the predictions made by a DNN, they do not shed light on the precise mechanisms by which learning in the network happens. Hidden layers of a DNN perform non-linear transformations on the representation input into the network. For a network to make accurate predictions of toxicity, the hidden layers need to encode chemical features linked to the task the DNN was trained on. Several studies have investigated the chemical information encoded in hidden layers.

In analogy to classical chemical fingerprints, the activations of neurons in a hidden layer can be considered as a neural fingerprint that has learned a chemical representation tailored to solve the prediction task (Menke & Koch, 2021). In the cited study the neural fingerprint was used in a similarity-based virtual screening experiment. In order to find actives for a given target, compounds with a similar neural fingerprints (taken from a DNN trained to predict activity on the target) to a query compound were retrieved. In a different study, the similarity of activations across all hidden layers was proposed as a measure of task-specific chemical similarity (Allen et al., 2020). By retrieving training compounds most similar to a test compound, this metric was used to support and rationalise the model's predictions. Sosnin et al. used DNNs to predict acute toxicity and analysed the hidden representations of chemicals with the t-SNE method, which embeds them in a 2D space (Sosnin et al., 2019). Distinct clusters of compounds having high acute toxicity emerged, which presumably correspond to different mechanisms of toxicity. All those studies demonstrated that hidden representations of chemicals in DNNs are meaningful in the context of the investigated bioactivity or toxicity tasks. However, no attempts were made to explicitly extract the chemical meaning learned in the hidden layers.

Information learned by DNNs has been most extensively studied for visual tasks performed by CNNs. It has been shown that a DNN constructs features of increasing complexity throughout the different layers of the network (Lee et al., 2011). When detecting faces, for instance, lower layers detect simple structures like blobs and edges from the raw pixels, while deeper layers combine those simple structures to more complex objects such eyes and noses. Analogously, when learning representations for chemicals, a DNN may detect the presence of simple substructures in the lower layers and combine those to more complex substructures that are meaningful for the task at hand.

Some attempts have been made to understand the chemical features learned in hidden neurons of a DNN. It was shown on the Tox21 dataset that the activation of hidden neurons can be correlated to the presence of toxicophores (known toxicophores for various toxicity endpoints were considered) in

the compound (Mayr et al., 2016). Furthermore, it was shown that the size of detected toxicophores (in atoms) increases in deeper layers (Preuer et al., 2019). However, those studies did not investigate whether those detected toxicophores are related to the modelled toxicity endpoints of the Tox21 dataset. A hidden neuron may in principle be responsive to a chemical pattern without the network using this information for the eventual prediction. The two cited studies shed some light on the mechanisms by which DNNs may learn features, but no attempts were made to leverage this information to interpret predictions made by a specific model.

7.3 Objectives

The aim of this project is to interpret QSAR models for toxicity prediction based on neural networks by understanding what chemical patterns are learned by single neurons of a network. Similar efforts have been made for image recognition tasks and are referred to as feature visualisation or activation maximisation (A. Nguyen et al., 2019; Olah et al., 2017). Approaches to achieve this include (i) inspecting exemplary images that strongly activate a neuron (Szegedy et al., 2014), (ii) optimising images in the input space to strongly activate a neuron (Erhan et al., 2009) and (iii) using generative models with the objective to create images that strongly activate a neuron (A. Nguyen et al., 2016). For the task of predicting toxicity of chemicals, it is of interest to find chemical (sub-)structures that strongly activate a neuron. Approaches for feature visualisation are distinct from approaches attributing importance to input features. The former approaches return features learned in hidden layers of the network, while attribution methods determine what features in the input space drive a prediction. In the domain of image recognition, feature visualisation and attribution have been used complementarily to enhance the understanding of a DNN (Olah et al., 2018).

In this project, neural networks based on substructure fingerprints, specifically Morgan Fingerprints from the RDKit package, are used. Initially neural networks consisting of a single hidden layer are considered before attempting to extend the developed approaches to DNNs. Based on trained neural networks, the following objectives were pursued and are described in different chapters of this thesis:

- To explore the space of neuron activations of training compounds including correlations between different neurons and the link between activation of single neurons and the prediction made by the network. (Chapter 8)
- To explore chemical patterns detected by single neurons using both training compounds and learned weight parameters of the network. (Chapter 8)

Chapter 7: Interpretation of neural networks for QSAR modelling

- To develop a method to automatically annotate hidden neurons of a neural network with substructures that activate the neuron and compare the detected substructures to known toxicophores. (Chapter 9 and 10)
- To develop a method that combines substructures known to activate hidden method with an attribution method measuring the importance of a neuron to a prediction and compare this method to a feature attribution method. (Chapter 10)
- To apply the developed methods to DNNs (2 hidden layers). (Chapter 11)

Chapter 8 Exploration of chemical features learned in hidden neurons of neural networks

8.1 Introduction

Neural networks and deep neural networks (DNNs) represent popular techniques for toxicity prediction, which are considered as difficult to interpret black-box models. To gain a better understanding of their inner workings, this chapter explores characteristics of a neural network model containing a single hidden layer trained for predicting toxicity. The fundamentals about neural networks have been described in section 3.4.8.

The overarching aim of the work in this chapter is to understand how the network has learned to predict toxicity by identifying the role of individual neurons of the network. Specifically, it is investigated whether chemical (sub-)structures can be identified as causes for the activation of individual neurons. This can be considered as an application of feature visualisation, which is a technique used in image recognition tasks to understand what visual stimuli a hidden neuron responds to (Olah et al., 2017). In a binary classification task, features learned in hidden neurons may be linked to positive or negative predictions. Here, features learned by neurons linked to toxic predictions are inspected with the aim of identifying chemical features linked to toxicity.

The chapter begins by introducing the data to be explored and the basic model building process. The methodology is then presented including a brief introduction to the structure of a neural network. In particular, a description is provided of how activation of hidden neurons results from the compound's chemical features and learned model weights. Confidence is introduced as a means to evaluate links between activation of individual neurons and predictions made by a model. The feasibility of the intended concept is explored on a relatively simple neural network with just one hidden layer. In a first step, general analyses on neuron activations were conducted, namely inspecting the range of activations for various hidden neurons. In a further analysis, correlations between the activations for different hidden neurons were evaluated to determine whether different neurons may learn similar features. In the following steps, attempts were made to determine which chemical features are detected by a specific hidden neuron. This was done by inspecting both compounds that strongly activate the neuron and fingerprint bits that have high learned weights in the hidden neuron. Finally, it was investigated how certain known toxicophores cause activations of various neurons across the

network to gain insights in the workings of the network as a whole. All of the analyses served the purpose of determining whether chemical features causing activation of hidden neurons can be found.

8.2 Methodology

8.2.1 Dataset

Ames mutagenicity was selected as the toxicity endpoint for this study, as it represents a well understood mechanism of toxicity and many different toxicophores have been identified (Kazius et al., 2005; Sushko et al., 2012). This means there are clear expectations on the chemical features a neural network needs to discover to accurately predict Ames mutagenicity and therefore features identified by the neural network can be compared to known causes for mutagenicity as a means of validation. To obtain the Ames dataset for this study, data from the following sources were combined:

- A curated version of the Hansen dataset (Hansen et al., 2009) created by Sherhod et al. (Sherhod et al., 2014)
- The ISSSTY dataset (Benigni et al., 2013)
- The EURL-ECVAM (European Union Reference Laboratory for Alternatives to Animal Testing) Ames positives DB (Corvi & Madia, 2017)
- CGX (Carcinogenicity Genotoxicity experience) database (Kirkland et al., 2005)
- Genotoxicity and Carcinogenicity database for marketed pharmaceuticals (Snyder, 2009)

The ISSSTY dataset contains data for a number of different bacteria strains as well as a label for the ‘overall call’ (positive if at least one strain is positive). In this study, the overall call was used. Only compounds labelled as negative or positive were kept, compounds labelled as equivocal or inconclusive were removed. The other sources contain data for an overall call only (that is, a single label). The curated Hansen dataset contains binary labels and no further changes were made. For the EURL-ECVAM dataset, compounds labelled as equivocal were removed. This dataset originally contained no SMILES strings. Where possible, SMILES strings were retrieved from CAS numbers using the CIRpy package in Python (Swain, 2015). For the remaining data sources, compounds with missing or equivocal labels were removed. After these processing steps, each compound was labelled with a binary outcome for the Ames Test.

Subsequently, the SMILES of all compounds were standardised using the same procedure as described in Chapter 5. In the following, data instances with identical SMILES were aggregated using a majority

vote of the labels. If equal numbers of positive and negative labels were found for a given SMILES, the compound was dropped from the final dataset. The final dataset consists of 7662 compounds.

The dataset was split into a training set, a validation set and a test set using a random split with proportions 80:10:10. The neural network model instance used in this chapter was obtained by training on the training dataset and evaluating on the validation set as described in the following section. The explorative analyses in this chapter were done using the training set. The validation set was used for hyperparameter optimisation of the model and is used in the following chapter to develop and validate a method to extract chemical features causing hidden neuron activation. The test set was held back for final evaluation.

8.2.2 Model training

The investigations in this chapter are applied to a neural network model containing a single hidden layer to test the intended approach initially on a fairly simple model. In Chapter 11, the investigations will be extended to DNNs. The model instance was obtained using a grid search for hyperparameter optimisation. The evaluated and selected parameter values are given in Table 8-1. The models were implemented in Pytorch (Paszke et al., 2019) with binary Morgan fingerprints of radius 1 mapped to 2048 bits used as chemical descriptors. Using radius 1 gave the best model performances in preliminary studies (results not shown). The ReLU activation function was used for neurons in hidden layers. In all cases, binary cross entropy was used as the loss function and optimised using the Adam optimiser.

Table 8-1 Parameters used for hyperparameter optimisation. Selected parameters in bold.

Hyperparameter	Tested values
Neurons per layer	512 , 1024, 2048
Batch size for optimisation	16 , 32, 64
L2 regularisation of neuron weights	0, 0.00001, 0.001
Dropout	0, 0.2, 0.5
Learning rate	0.0001, 0.00033, 0.001

Each model was trained for a maximum of 10 epochs using early stopping to prevent overfitting. Specifically, the performance on the validation set was recorded after each epoch and the best performing model instance (after a particular epoch) was retained. The models were evaluated using

ROC-AUC score. The best performing model instance was used throughout this chapter. As can be seen in Table 8-1, the obtained model instance's hidden layer contains 512 neurons. The performance of the obtained model is reported in the results section.

8.2.3 Neural network structure

Figure 8-1 shows the architecture of a DNN with two hidden layers. Each data instance (for training or prediction) is represented as a feature vector provided in the input layer to the network. For this project, substructure fingerprints were used as chemical descriptors. Each bit set on in a substructure fingerprint can be attributed to a defined substructure. In the case of RDKit's Morgan Fingerprint, substructures are circular chemical environments centred at a particular atom of the compound.

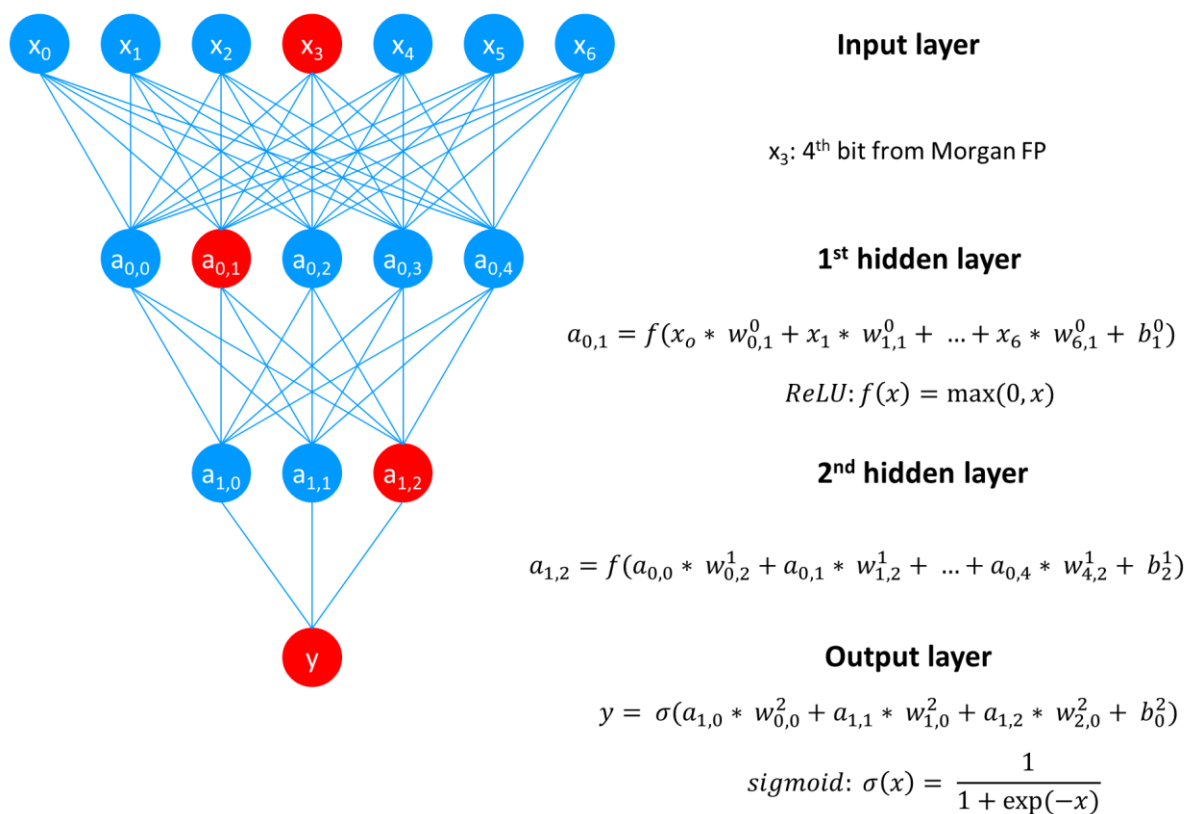


Figure 8-1 Architecture of a DNN. Shown is the architecture of a simple (feedforward) DNN with 2 hidden layers. On the right side, equations for all neurons highlighted in red are provided.

When a test compound is entered into a trained network, a numerical activation value is computed for each neuron, which depends on activations of neurons in the preceding layer as well as learned weights and biases. Specifically, each neuron is connected to all of the neurons of the preceding layer

by particular learned weights. This means that a neuron in the first hidden layer receives input directly from all bits of the chemical fingerprint used as input. In general, the activation of each hidden neuron represents a linear combination of neuron activations in the preceding layer followed by the application of a non-linear activation function. Thus, neurons in the second hidden layer receive inputs from the first hidden layer and so on. Activation of a hidden neuron for a particular data instance (train or test compound) refers to the numeric value the neuron holds after application of a non-linear activation function as shown in the equations in Figure 8-1. In general, activations are dimensionless and have no explicit meaning in the network beyond that.

While activation values have no apparent meaning, a model capable of accurately predicting toxicity must contain some information related to the modelled toxicity endpoint in its hidden representation. In each hidden layer, the representation of an instance is transformed into a different form encoding information linked to the compound's toxicity. This is achieved during training by tuning weights and biases to minimise the loss on training instances. Properties such as toxicity of a chemical are linked to its structure and hence it is expected that activation of hidden neurons corresponds to some meaningful representation of a chemical's structure. Specifically, individual neurons may when activated detect the presence of chemical groups linked to toxicity (i.e. toxicophores).

Neurons in the last (here second) hidden layer are directly linked to the output neuron of a network. For binary classification tasks, a sigmoid function is normally applied to the output neuron in order to obtain an estimated probability for the toxic class. Zero as input to the sigmoid function yields an estimated (default) probability of 0.5. Hence, activation of neurons in the last hidden layer possessing a positive output weight contribute towards an instance being predicted as toxic, whereas the opposite is the case for neurons with a negative weight. Inspecting the output weights of neurons in the last hidden neuron can be used to determine whether features learned in them are relevant for toxic or non-toxic predictions.

A different method to determine the association of a hidden neuron to an output class is computing its confidence (of a positive prediction) for training compounds. This requires the activations of hidden neurons to be binarised using an activation threshold.

The confidence of a neuron is calculated according to the following formula:

$$confidence_{neuron} = \frac{n_{neuron\ activated\ AND\ positive\ prediction}}{n_{neuron\ activated}}$$

where n is the count of training compounds fulfilling the stated conditions. For instance, if 100 compounds possess an activation higher than a selected threshold (e.g., 0.25) and 90 of these compounds are predicted as toxic, then a confidence of 0.9 is obtained which indicates a strong link

between activation of the neuron and toxic predictions. This method is not restricted to neurons in the last hidden layer, it can be applied to all hidden neurons.

8.2.4 Exploration of neuron activations

In a first attempt to understand the characteristics of the network's hidden layer, neuron activations for training compounds were explored. Specifically, for each hidden neuron, the maximum and the average activation values were determined. Then, the distribution of those values obtained for all hidden neurons was analysed.

Next, in order to gain some understanding of the relations between different neurons, the pairwise correlations of neuron activations for all training compounds were computed. Each hidden neuron has an activation value for each training compound and hence can be represented as an n-dimensional vector (here n=5889, i.e., the number of training compounds). The pairwise correlation between two neurons was computed as the Pearson correlation coefficient between the neurons' vectors.

To further explore the role of hidden neurons within the network, their link to positive and negative predictions was investigated. For the activation of each hidden neuron, the confidence of a positive prediction was computed, as defined above. This analysis requires a threshold to be set for the activation value to determine when a neuron is activated. Since this selection is arbitrary, different threshold values were examined. It is worth noting that these thresholds are merely artificial constructs to support this analysis. In practice, activation of a neuron is not a binary process, and instead defined by the equations shown in Figure 8-1.

8.2.5 Exploration of chemical features learned in hidden neurons

Two different sources of information were considered to attempt to identify chemical features responsible for neuron activation: first, the compounds most strongly activating a neuron; and second, the bits of the chemical fingerprint having high weights for the neuron. Neurons linked to toxic predictions according to confidence values were analysed, as these are expected to detect chemical features responsible for a compound being mutagenic. In addition, some neurons linked to non-toxic predictions and neurons with neutral confidence values were also analysed. The analysed neuron was characterised as being associated with toxic or non-toxic predictions, and confidence values (for different thresholds) and its output weight (to the single neuron in the output layer, see above) are reported.

For the compound analysis, the 12 training compounds most strongly activating the neuron (referred to as Top-12 compounds) are displayed and common chemical substructures among them were searched for manually as potential chemical features learned by the neuron.

For the analysis of fingerprint bits, the 18 bits having the highest positive weights (referred to as Top-18 bits) are displayed. Also displayed are bits that encode atom environments linked to the potential chemical features identified from the Top-12 compounds analysis. This was done to examine whether those bits have significant positive weights, even though they are not part of the Top-18 bits. Due to the nature of Morgan fingerprints, a chemical group such as a multi-atom toxicophore is encoded by multiple fingerprint bits and all those bits need to be considered to determine the neuron activation caused by the group.

8.2.6 Network-wide activation analysis for prototypical toxicophore compounds

To understand what activations single compounds cause across the network, activations for simple compounds containing known toxicophores for mutagenicity were obtained. In particular, the number of activated neurons was determined for various thresholds. Since all the compounds possess well-characterised chemical groups linked to mutagenicity, this analysis allows an estimation of how many hidden neurons may potentially be responsive to compounds having the specific toxicophore. Finally, neurons strongly activated by the aromatic azide compounds were analysed in more detail. In particular, activations for azide compounds in those neurons were compared to mean activations for all training compounds. In addition, learned weights for fingerprint bits corresponding to various atom environments linked to azide groups were compared between those neurons.

8.3 Results and Discussion

8.3.1 Model evaluation

Figure 8-2 reports various classification metrics of the selected model on both the training and the validation set. Unsurprisingly, the model performs better on the training set. Nonetheless, it performs reasonably well on the validation set with a ROC-AUC of around 0.9, accuracy of 0.82 and an MCC of 0.65. The balanced accuracy is similar to the (regular) accuracy indicating that the model possesses comparable predictivity for both mutagenic and non-mutagenic compounds.

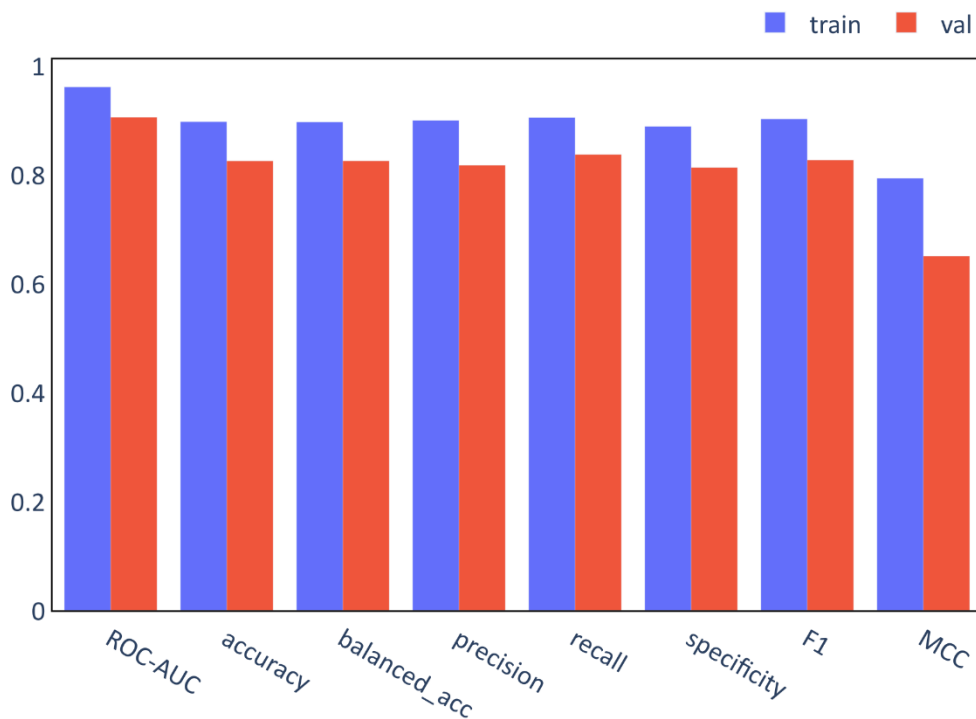


Figure 8-2 Performance of the neural network model. Compared are performances on the training and the validation set using a range of classification metrics.

8.3.2 Exploration of neuron activations

The first analysis of the trained network was conducted on hidden neuron activations obtained when training compounds are given as inputs to the model. Figure 8-3 shows the maximal (A) and mean (B) activation for all hidden neurons computed across all the training compounds and Table 8-2 lists the percentiles of these distributions.

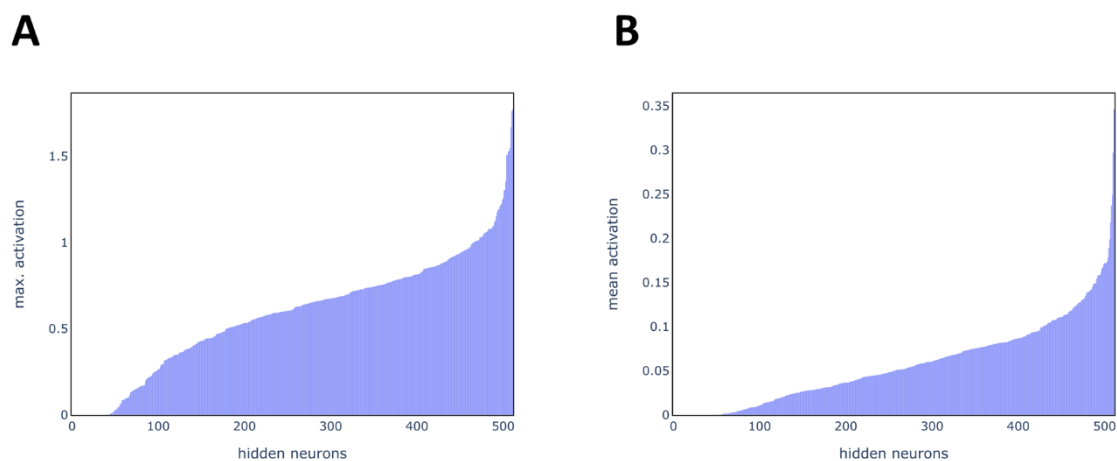


Figure 8-3 Maximum and mean activation of hidden neurons. The maximum and mean activation values for the hidden neurons were computed across all training compounds.

Table 8-2 Percentiles of distributions for maximum and mean activation of neurons. The percentiles describe the distributions depicted in Figure 8-3.

	Max activation	Mean activation
20 th percentile	0.274	0.011
40 th percentile	0.537	0.037
50 th percentile	0.610	0.050
60 th percentile	0.681	0.063
80 th percentile	0.847	0.089
100 th percentile (max)	1.77	0.35

For most neurons, the highest activation values found are below 1 and the mean activation values across training compounds are below 0.1. For some of the neurons, none of the training compounds cause strong activation. For instance, for 46 of the neurons there were no activation values above 0.01. These neurons seem to be irrelevant to the model, as their activation does not vary between different compounds.

To further understand the activation space of hidden neurons, the distributions of activation values for single neurons over training compounds were inspected. For this analysis, four exemplary hidden neurons were selected, based on the different characteristics of the distributions. Figure 8-4 shows the distribution of training compound activations for the neurons 1-173 (A), 1-392 (B), 1-478 (C) and 1-511 (D).

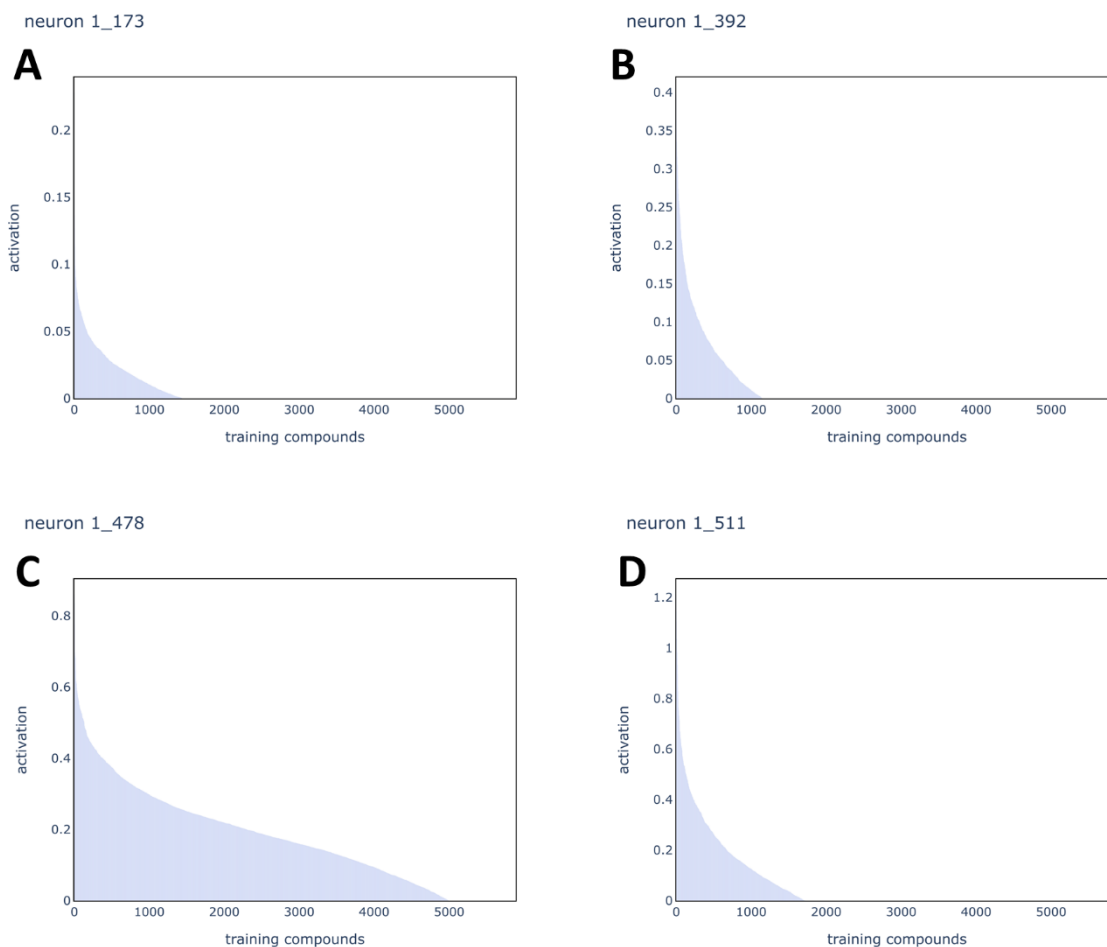


Figure 8-4 Activations for individual neurons. A: 1-173, B: 1-392, C: 1-478, D: 1-511. Note that the activation axes for the neurons are on different scales.

Some differences can be observed between the different distributions. For neuron 1-478, around 5000 of the training compounds cause an activation noticeably larger than zero, whereas for the other neurons fewer than 2000 compounds cause an activation. In addition, the maximal activations observed vary from around 0.2 for neuron 1-173 to 1.2 for neuron 1-511. These findings demonstrate that some neurons seem to be activated by a wide range of compounds and hence different input descriptors, whereas for others a narrower set of compounds leads to significant neuron activations.

In the following, pairwise correlations between hidden neurons were analysed in order to investigate potential relations between different hidden neurons. In particular, activations for all training compounds were used as a vector to represent activation space for each neuron. A high positive correlation between two neurons suggests that those neurons detect similar chemical patterns, whereas negative correlations indicate compounds that possess high activations for neuron A, and

low activations for neuron B, or vice versa. Neuron correlations for the training set are visualised as a heat map in Figure 8-5A and as a histogram in Figure 8-5B.

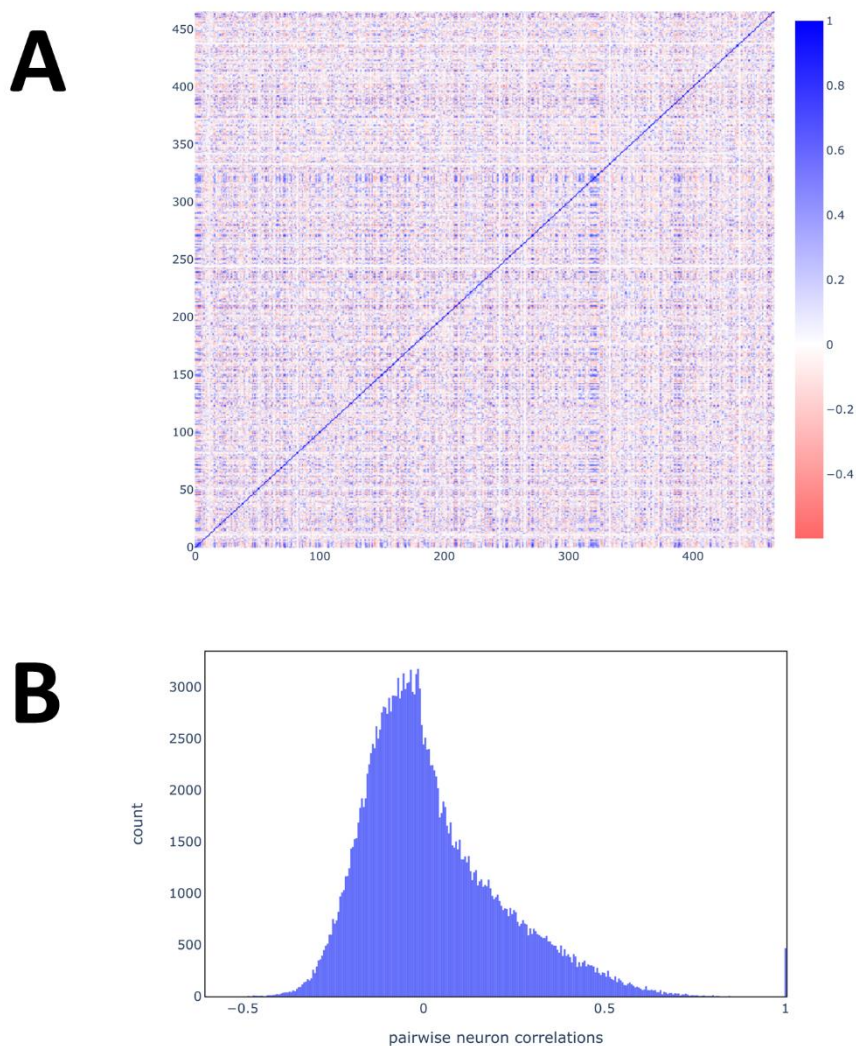


Figure 8-5 Pairwise correlations of neuron activations. For the training set, all pairwise neuron correlations were computed and visualised as a heatmap (A) and a histogram (B). Only neurons with a maximum activation larger than 0.01 in the training set were included. This filtering led to the exclusion of 46 of the 512 hidden neurons.

In the heatmap, blue cells indicate a positive correlation between neurons, red cells a negative correlation, and white cells a correlation coefficient of around 0. The highest pairwise correlations (not considering correlations of neurons with themselves) were found to be around 0.8. It can be hypothesised that these pairs of neurons learned to detect similar or even the same chemical patterns. Negative correlations in the dataset were found to be weaker than the positive ones (strongest around -0.5). Negatively correlated pairs of neurons cannot be expected to detect the same or similar chemical features. A hidden layer neuron may possess learned negative weights for certain bits of the

chemical fingerprint, which may lead to low activations for compounds with such bits set on. However, since negative activations are prevented by the ReLU function, the occurrence of very strong negative correlations between neurons may be hindered. This may explain why negative correlations overall were found to be less strong than positive ones.

Confidence values were computed to investigate the link between neuron activation and predictions made by the model. Confidence values of the different neurons are displayed in Figure 8-6 as scatter plots together with the support values indicating the proportion of compounds that activate the neuron (given the respective threshold).

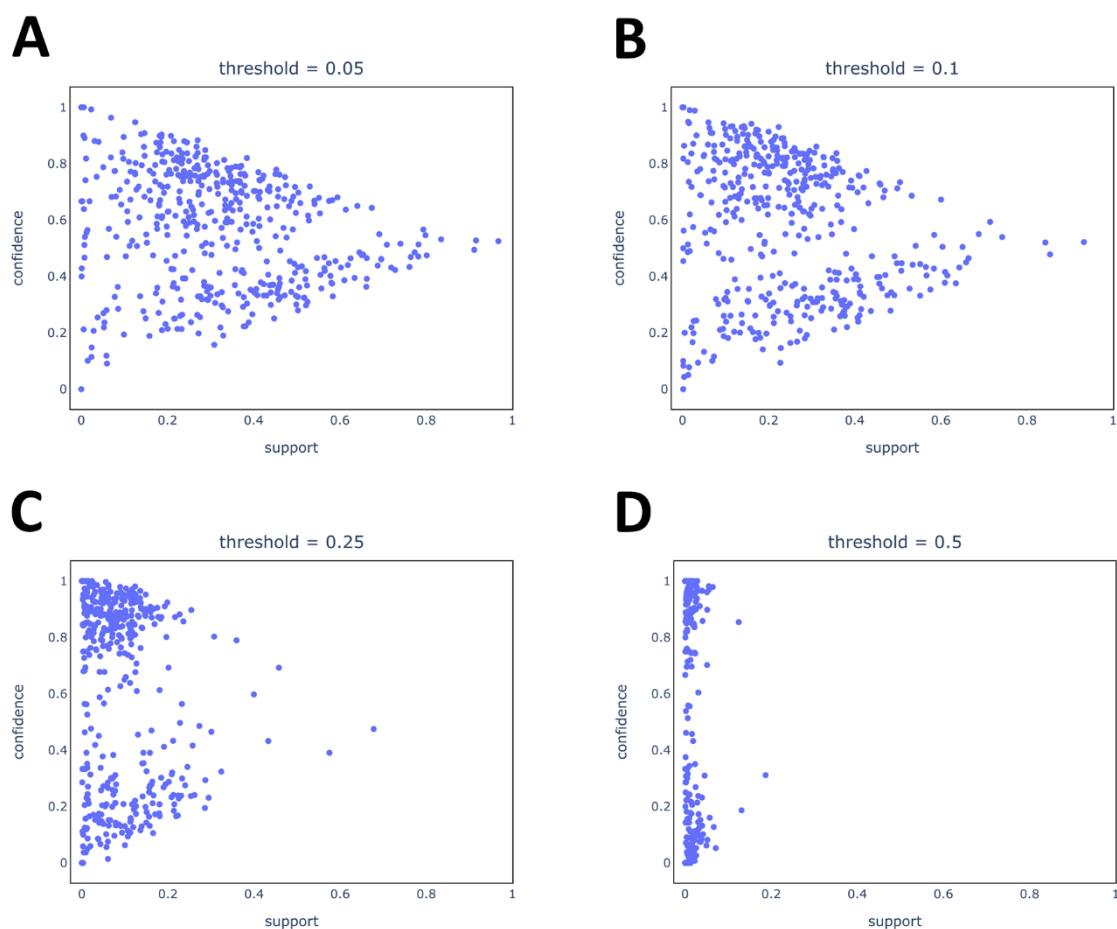


Figure 8-6 Confidence analysis of hidden neurons. The confidence for a positive prediction (given the activation threshold) and the corresponding support are shown for all hidden neurons. Thresholds are 0.05 (A), 0.1 (B), 0.25 (C), 0.5 (D).

Neurons possessing a high confidence are strongly linked to positive (i.e. mutagenic) predictions of compounds and hence are hypothesised to detect chemical patterns linked to mutagenicity (in the Ames test). Conversely, neurons with very low confidence values are linked to negative (i.e. non-mutagenic) predictions. These neurons may detect chemical structures typically not found in

mutagenic compounds. Neurons with a confidence value close to 0.5 do not seem to be generally linked to positive or negative predictions. Substructures detected by these neurons are not likely to be directly linked to positive or negative predictions.

The plots in Figure 8-6 clearly show that the activation threshold strongly affects the support and the confidence observed for the neurons. For the lower two thresholds (0.05 and 0.1), the full range of values from zero to one occurs for both support and confidence. However, neurons with a high support value (>0.6) possess confidence values close to 0.5. The activation of these neurons (for the low thresholds) does not discriminate between positive and negative predictions. Neurons with lower support values are associated with confidence values on the full range from zero to one. The activation of neurons with low confidence values is linked to negative predictions, whereas activation of neurons with high confidence values is linked to positive predictions. The low support values for both types of neurons indicates that only a low proportion of compounds activates these neurons.

Fewer compounds activate the neurons at higher thresholds for neuron activation as in C (0.25) and D (0.5). For the highest threshold, no neuron has a support above 0.2. It can also be observed that at these higher thresholds fewer neurons possess confidence values close to 0.5. Hence, when setting a more restrictive threshold for activation, the activation becomes more strongly linked to negative or positive predictions. Neurons strongly linked to positive predictions upon strong activation can be hypothesised to detect chemical patterns linked to toxicity. Unravelling such chemical patterns is expected to give insights into the workings of a network.

8.3.3 Exploration of weights for fingerprint bits

As described above, neurons in the first hidden layer receive input from the bits of the fingerprint via learned weights. Weights of different magnitude for different bits of the fingerprint are the reason why neuron activation varies between different compounds. Analysis of which bits possess high weights can help to identify chemical features responsible for neuron activation. Typical distributions of weights are given in Figure 8-7 for the same neurons as in Figure 8-4.

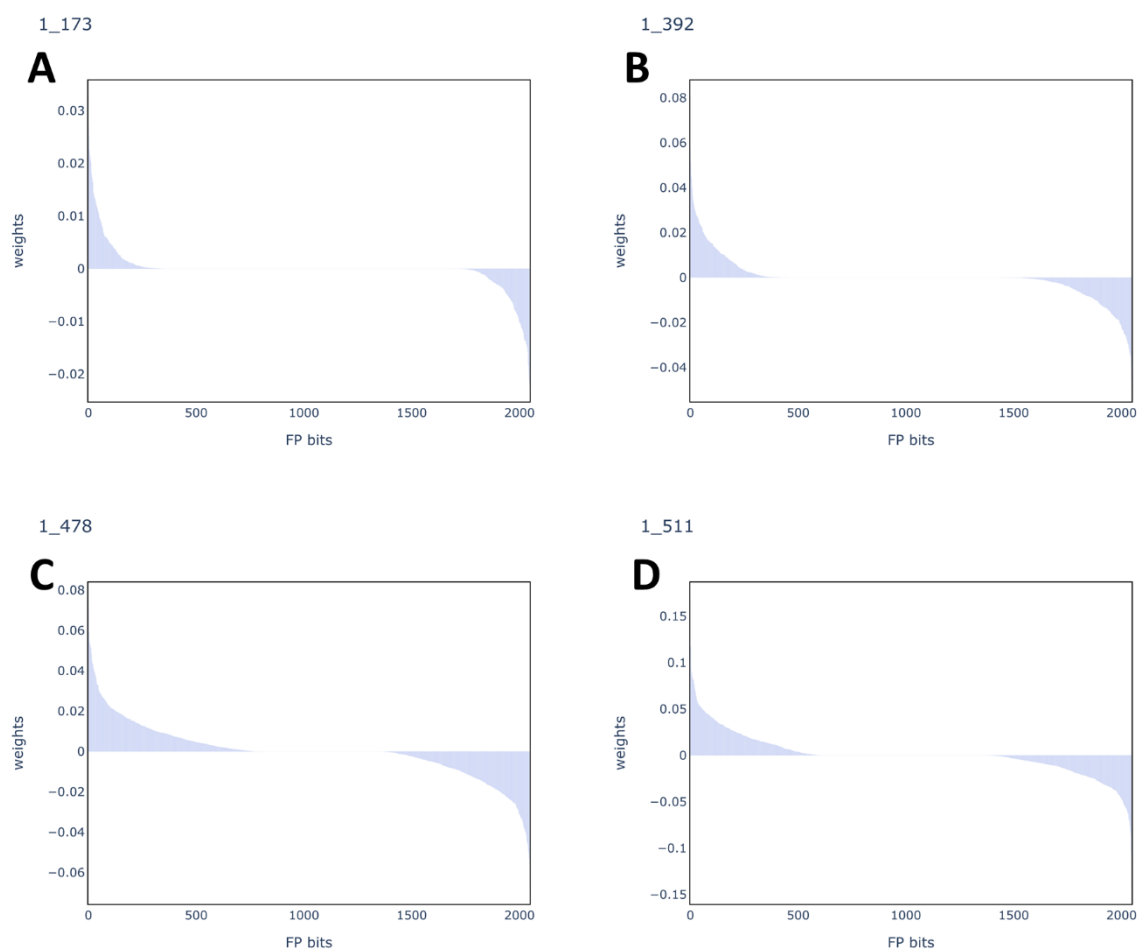


Figure 8-7 Weight distributions for individual neurons. A: 1-173, B: 1-392, C: 1-478, D: 1-511. The weights axes for the neurons are on different scale.

Bits with positive weights, negative weights and weights of (close to) zero exist for all neurons. Among the four shown neurons, the most non-zero weights were found for neuron 1-478 and the least for neuron 1-173. Notably, the number of positive and negative weights is comparable for a given neuron. Differences between the neurons exist in the magnitudes of positive and negative weights. The maximum positive and negative weights for neuron 1-511 are around 0.15, whereas they are in the range 0.02-0.03 for neuron 1-173. A common feature of the distributions is a gradual decrease in weights for different bits. This makes it difficult to distinguish between relevant and clearly non-relevant bits. For instance, for neuron 1-511 the bit with the 100th-highest weight has a weight one fourth of the maximum weight for this neuron. Therefore, a large number of bits may have significant impact on neuron activation. Even for neurons with smaller numbers of non-zero weights, an analysis of all those individual weights would be very cumbersome.

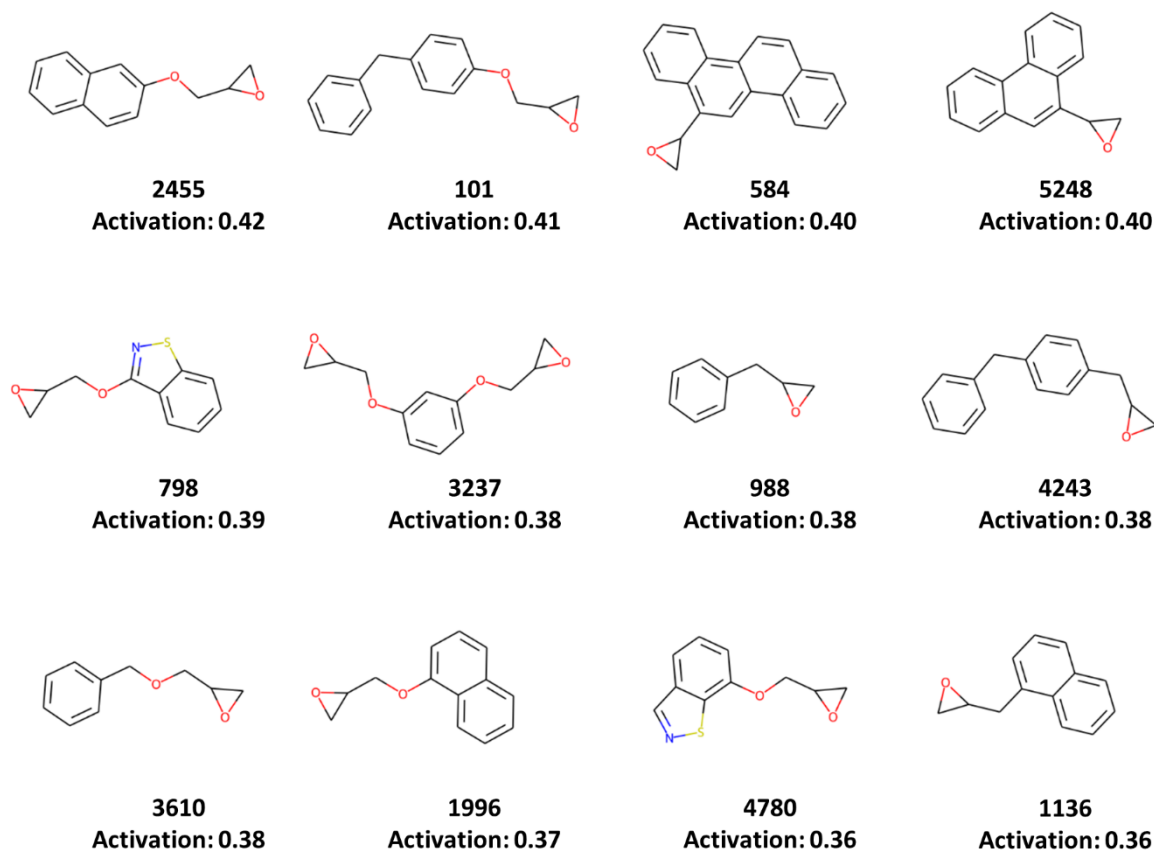
8.3.4 Exploration of compounds strongly activating neurons

The following analysis attempts to identify chemical features detected in individual hidden neurons. In a first step, the compounds most strongly activating some exemplary neurons were inspected. Analysed were four neurons linked to positive predictions (1-43, 1-69, 1-153, 1-180) and two neurons linked to negative predictions (1-18, 1-128).

The Top-12 compounds, that is, the 12 compounds with the highest activation values for neuron 1-43 are shown in Figure 8-8. The positive weight of the neuron of 0.103 to the output neuron indicates that activation of the neuron contributes to positive predictions. This is supported by the high confidence values found for this neuron at thresholds 0.05, 0.1 and 0.25. A confidence value for the threshold 0.5 could not be computed as the maximum activation value for this neuron was 0.42. By inspecting the compounds, an epoxide moiety can be identified as a common substructure of all compounds. The epoxide functional group is known to cause mutagenicity (see section 2.3.2). Moreover, all compounds possess an aromatic ring, and six of them possess a phenolic ether group.

Neuron 1-43:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.77, 0.82, 0.9, -
 Weight to output neuron: 0.103



Identified substructures:



Figure 8-8 Analysis of compounds strongly activating neuron 1-43. Confidence values at different thresholds and the weight contributed to the output neuron are included to characterise the neuron's link to positive/negative predictions. Shown are the 12 compounds from the training set with strongest activation values (the compound identifiers and activation values are shown). Substructures shared by the compounds were manually identified and are listed under 'Identified substructures'.

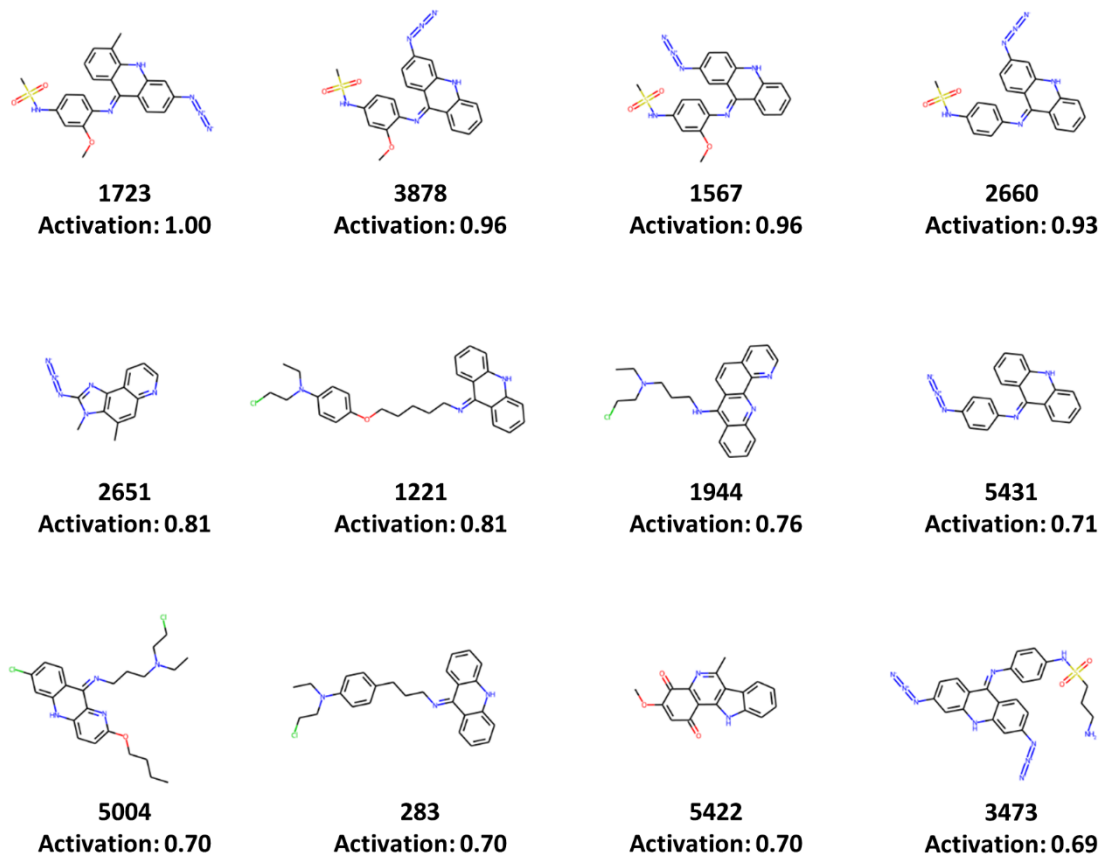
The Top 12 compounds for the neuron 1-69 are shown in Figure 8-9. This neuron is also strongly linked to positive predictions with a high positive weight value and high confidence values. By carefully inspecting the Top-12 compounds several common substructures were identified. Most prominently, seven of the compounds contain the azide group, which is a known toxicophore (Kazius et al., 2005),

attached to an aromatic ring. 10 of the compounds contain an acridine ring system (or a derivative), which is also a toxicophore for mutagenicity (Brown et al., 1980). Furthermore, an aromatic sulphonamide group and aliphatic 2-chlorine amine substructure can be found. The former is not known as a toxicophore, but possesses some similarity to alkyl esters of sulfonic groups (alert TA411 in ToxAlerts). Aliphatic halides (and hence chloride) in general represent an alert for mutagenicity (Fishbein, 1976) and together with the amine, the group is similar to the nitrogen mustard group (Benedict et al., 1977). These findings suggest that a single neuron may detect more than one relevant toxicophore.

Neuron 1-69:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.71, 0.75, 0.87, 1.0

Weight to output neuron: 0.178



Identified substructures:

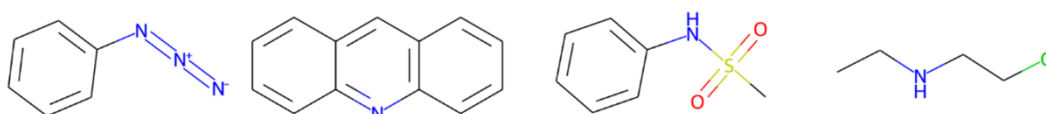


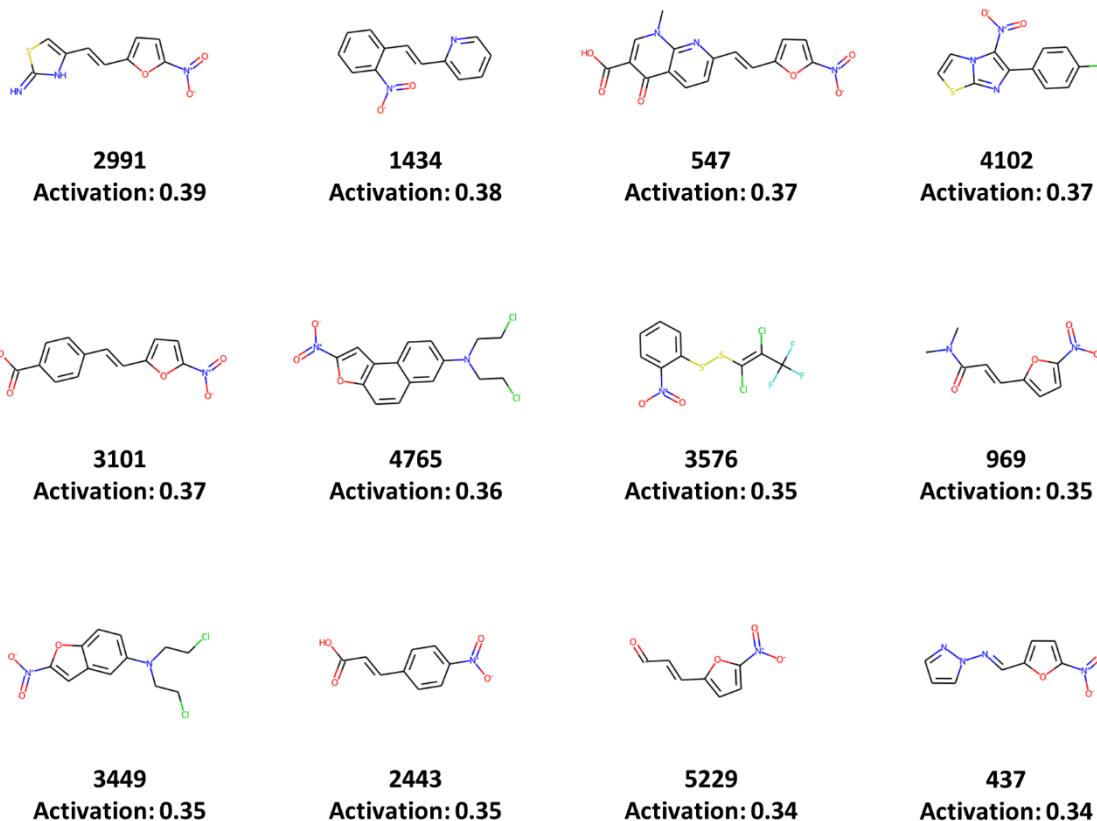
Figure 8-9 Analysis of compounds strongly activating neuron 1-69. Details are given in the caption of Figure 8-8.

The Top-12 compounds (Figure 8-10) for neuron 1-153 all possess an aromatic nitro group, a known toxicophore for mutagenicity (see section 2.3.2). In this set of compounds, the nitro group is attached to different types of aromatic ring including phenyl, furane and bi- or tricyclic systems. Compounds 4765 and 3449 contain a nitrogen mustard group (N-halide structure), also known to be mutagenic. It may be that the neuron also recognises this group as a relevant feature, yet by inspecting the compounds, the aromatic nitro group appears to be the primary feature detected by neuron 1-153.

Confidence values and the weight to the output neuron indicate that activation of this neuron is linked to positive predictions. Compared to other neurons, the weight to the output neuron is relatively low. However, this does not necessarily indicate a relatively weak link between the chemical features learned in this neuron and the toxicity end point (here mutagenicity). It may be that a particular toxicophore is detected by a large number of different neurons, which all may contribute to positive predictions. The range of neurons activated by compounds containing a single toxicophore is investigated below (see section 8.3.6).

Neuron 1-153:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.85, 0.91, 0.99, -
Weight to output neuron: 0.050



Identified substructures:

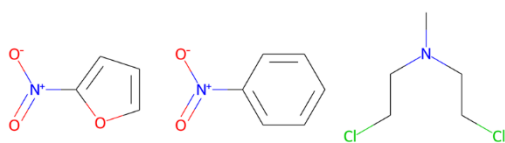


Figure 8-10 Analysis of compounds strongly activating neuron 1-153. Details are given in the caption of Figure 8-8.

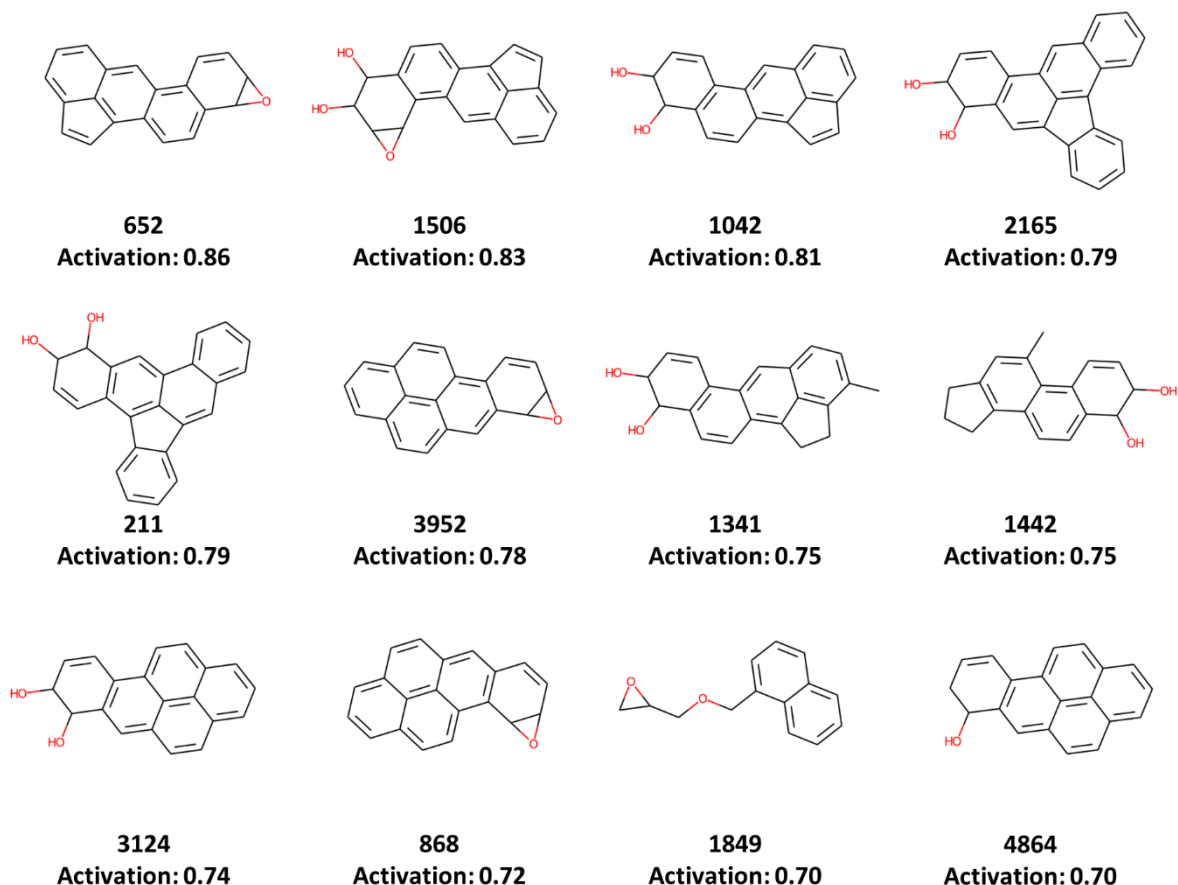
Polycyclic aromatic hydrocarbon (PAH) systems can be identified as a common structural feature among the most strongly activating compounds for neuron 1-180 (Figure 8-11). PAHs are a well characterised toxicophore for mutagenicity (Benigni & Bossa, 2011). The PAH ring systems found in the compound set are of varied constitution (e.g. pyrene or anthracene systems). In addition, all of the compounds possess either an epoxide group or are (mono- or di-) hydroxylated, suggesting that

this also represents a chemical pattern contributing to activation of the neuron. It is known that the mutagenicity of PAHs is mediated through diol and epoxide metabolites formed by CYP450 enzymes. It seems that the neuron detects both the basic PAH structure and attached groups associated with mutagenicity.

Neuron 1-180:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.70, 0.76, 0.90, 0.98

Weight to output neuron: 0.234



Identified substructures:

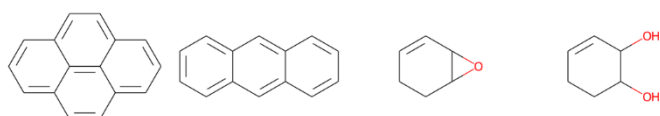


Figure 8-11 Analysis of compounds strongly activating neuron 1-180. Details are given in the caption of Figure 8-8.

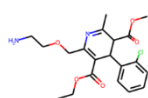
So far, only neurons linked to positive predictions were presented. In the following, two neurons associated with negative predictions are presented.

Neuron 1-18 possesses a strong negative weight (-0.180) to the output neuron. Additionally, the confidence values suggest that activation of this neuron is linked to negative predictions. An interesting substructure among the Top-12 compounds (Figure 8-12) is the pyranose ring found in saccharides. However, this substructure occurs only in three out of the 12 compounds. Seven of the compounds contain a pyridine ring. Neither of these structural motifs is listed as an alert for mutagenicity in the ToxAlert database.

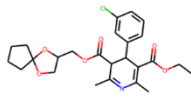
Neuron 1-18:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.23, 0.17, 0.06, 0

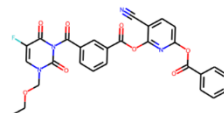
Weight to output neuron: -0.180



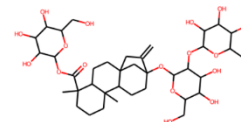
910
Activation: 0.70



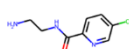
3301
Activation: 0.61



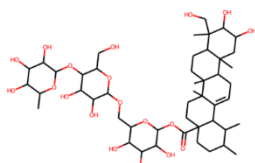
4868
Activation: 0.56



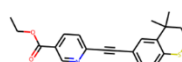
1174
Activation: 0.55



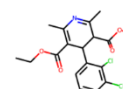
5077
Activation: 0.52



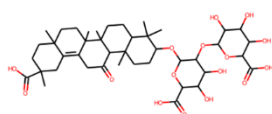
2253
Activation: 0.51



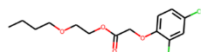
1241
Activation: 0.51



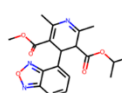
4845
Activation: 0.51



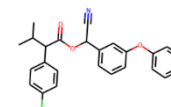
11
Activation: 0.48



690
Activation: 0.37



2447
Activation: 0.47



3988
Activation: 0.45

Identified substructures:

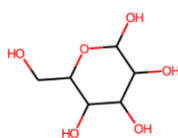


Figure 8-12 Analysis of compounds strongly activating neuron 1-18. Details are given in the caption of Figure 8-8.

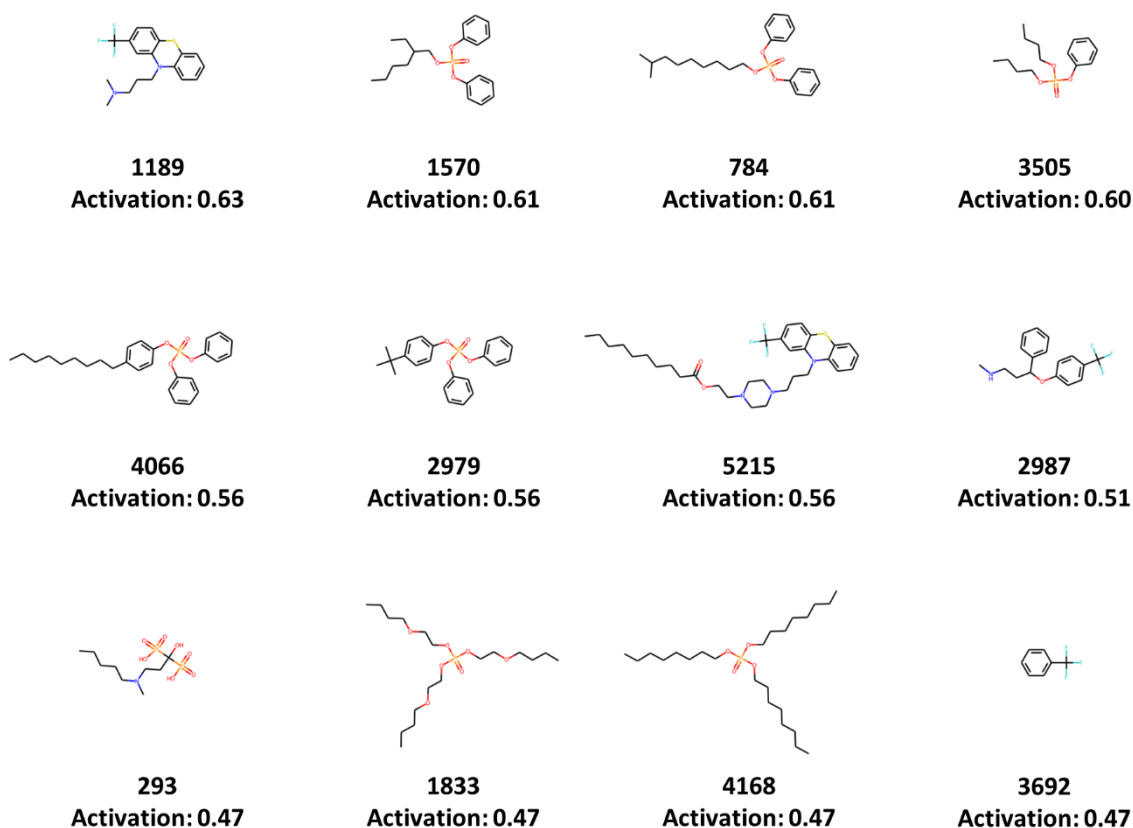
Neuron 1-128 represents a second example of a neuron strongly linked to negative predictions. Eight of the Top-12 compounds (Figure 8-13) are esters of phosphoric acid (both with aliphatic and aromatic alcohol components). Interestingly, alkyl esters of phosphonic esters have been described as a structural alert for mutagenicity (Ashby & Tennant, 1988), but not esters of the phosphoric acid. All the phosphoric ester compounds in this set have been labelled as non-mutagenic. A further structural

motif identified among the compounds are long aliphatic chains (displayed as octyl). Alkyl chains have not been linked to mutagenicity.

Neuron 1-128:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.26, 0.20, 0.08, 0

Weight to output neuron: -0.262



Identified substructures:

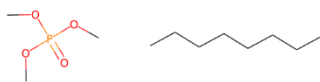


Figure 8-13 Analysis of compounds strongly activating neuron 1-128. Details are given in the caption of Figure 8-8.

Neurons linked to positive and negative predictions have been presented. Neurons linked to positive predictions can be expected to detect chemical substructures linked to Ames mutagenicity. Indeed,

the presented neurons were found to be activated by compounds containing known toxicophores for mutagenicity. These findings seem to support reports from the literature that hidden neurons in DNN models for toxicity prediction function as toxicophores detectors (Mayr et al., 2016; Preuer et al., 2019). For neurons linked to negative predictions, the interpretation of the findings is more difficult. These neurons may detect substructures rarely occurring in mutagenic compounds. This situation is distinct from positive cases where mutagenic effect can be clearly attributed to, for instance, the chemical reactivity of a certain chemical group.

By inspecting compounds that strongly activate a neuron a chemist may be tempted to identify causal explanations for the activation. However, it cannot be concluded whether a substructure causes the activation of a neuron or whether it merely co-occurs with the actual cause (correlation does not imply causation). In the following analysis, the learned weights from the input layer to the first hidden layer were explored as an additional source of information on what chemical patterns are detected by hidden neurons.

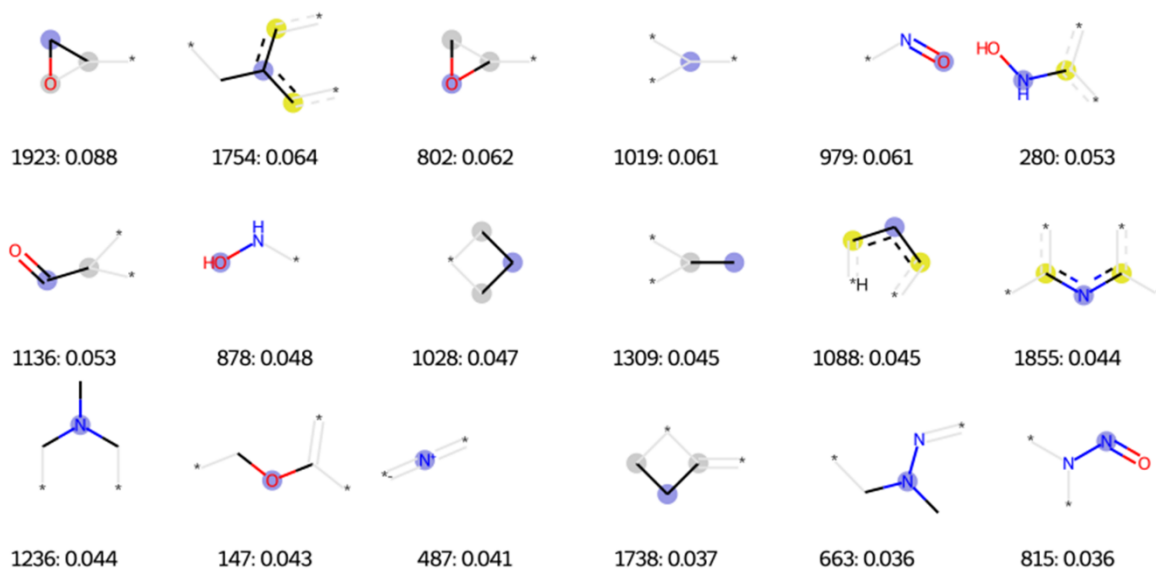
8.3.5 Exploration of fingerprint bits with high weight

To explore what information can be gained by analysing fingerprint bits having high weights for a given neuron, the same neurons as in the previous section were analysed.

Figure 8-14 shows the Top-18 bits (highest positive weights) for neuron 1-43 which, according to compounds most strongly activating it, seems to detect epoxides. Indeed, three of the four bits with the highest weights (1923, 802, 1019) encode an atom environment occurring in epoxide structures. This confirms the assumption that epoxide structures cause activation of the neuron. A further bit associated with epoxides (206) does not appear in the Top-18 bits and has a small positive weight.

Neuron 1-43:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.77, 0.82, 0.9, -
 Weight to output neuron: 0.103



Other fingerprint bits:

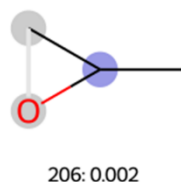


Figure 8-14 Analysis of bits with high weight for neuron 1-43. As in Figures 8-8 to 8-13, confidence values at different activation threshold and the weight contributed to the output neuron are shown to characterise the neuron. Listed are the atom environments of the 18 fingerprint bits with the highest weights for the neuron. For visualisation purposes, atom environments for the smallest compound (lowest molecular weight) in the training set were automatically generated. In case of bit collisions, a particular bit may encode more than one distinct atom environment. Under 'Other fingerprint bits' selected bits related to chemical features assumed to be detected by the neuron are shown (epoxides cause strong activation and bits linked to epoxides outside the Top-18 might have high weights). Explanation of the depicted atom environments: the blue circle indicates the central atom of the circular environment (radius 0 or 1 bonds); grey circles indicate an atom being part of an aliphatic ring, whereas yellow circles stand for aromatic rings. For peripheral atoms, the type of attached bonds (shown as grey bonds) is part of the defined atom environment

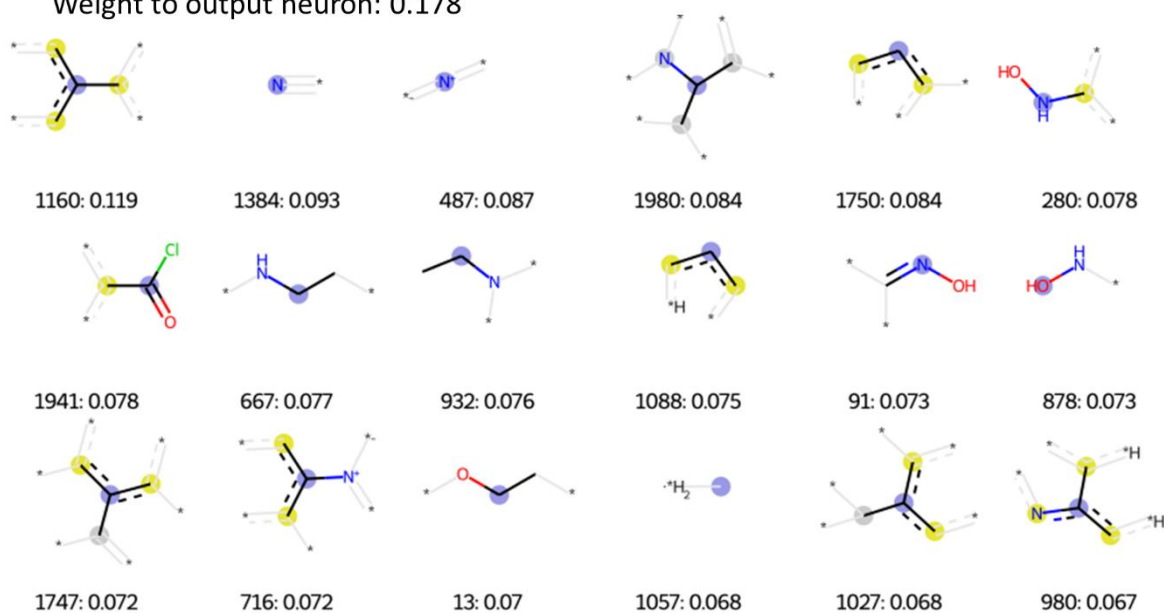
The Top-18 bits for neuron 1-69 are shown in Figure 8-15. Bit 487, which has the third highest weight, encodes for a part of the azide group. While at first sight it appears that no other bits encoding for parts of the azide group are represented in the Top-18, the bits 13, 1838, 1854, 740 all possess significant weights. An azide compound will turn on all those bits and the azide group will hence contribute the sum of all these weights to the activation of the neuron. The size of the bit weights indicates that bits beyond the Top-18 may possess weights only moderately lower than those in the Top-18 which means that for a thorough analysis a larger set of bits should be considered. The figure

reveals that bit 13 (15th-highest weight) is subject to bit collision in the dataset. In the Top-18 the bit is shown to encode for a carbon attached to an oxygen and another carbon, whereas below it is shown to encode for two double bonded nitrogen atoms attached to an aromatic ring (indicative of aromatic azide compounds). As described in the caption of Figure 8-14, the environments displayed in the Top-18 were automatically generated from the compounds with the lowest molecular weight in the full training set.

Neuron 1-69:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.71, 0.75, 0.87, 1.0

Weight to output neuron: 0.178



Other fingerprint bits:

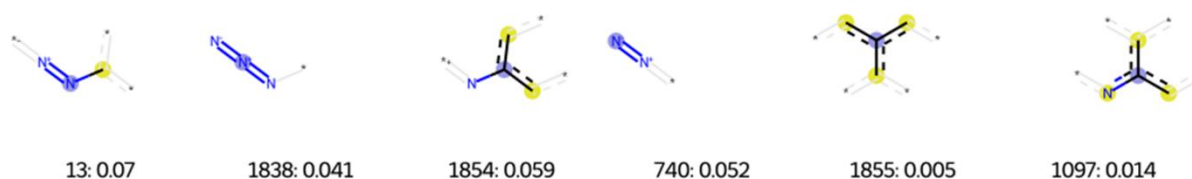


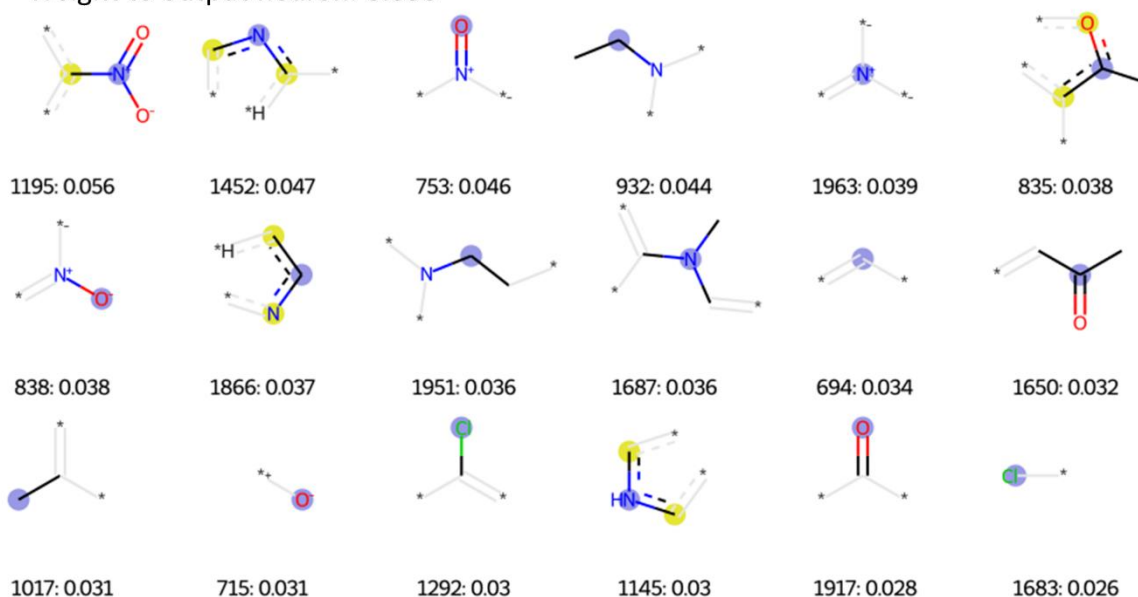
Figure 8-15 Analysis of bits with high weight for neuron 1-69. Details are given in the caption of Figure 8-14.

Neuron 1-153 (Top-18 bits shown in Figure 8-16) was linked to aromatic nitro compounds in the preceding analysis. The bit with the highest weight (1195) indeed captures all atoms that form an aromatic nitro group (including one aromatic carbon). This can therefore be considered the most specific bit for aromatic nitro compounds, which in principle should be sufficient to identify all aromatic nitro compounds. Nonetheless, the network assigned high weights to less specific bits such

as bit 753 (oxygen with double bond to positively charged nitrogen) and bit 1963 (positively charged nitrogen). The bits 1274 and 1035 (not in the Top-18, yet with significant weights) encode atom environments centred around the aromatic carbon that the nitrogen is bound to. Bits linked to the nitrogen mustard structure were not found in the Top-18.

Neuron 1-153:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.85, 0.91, 0.99, -
Weight to output neuron: 0.050



Other fingerprint bits:

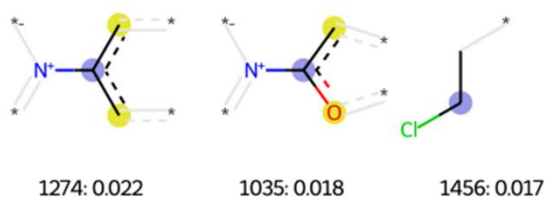


Figure 8-16 Analysis of bits with high weight for neuron 1-153. Details are given in the caption of Figure 8-14.

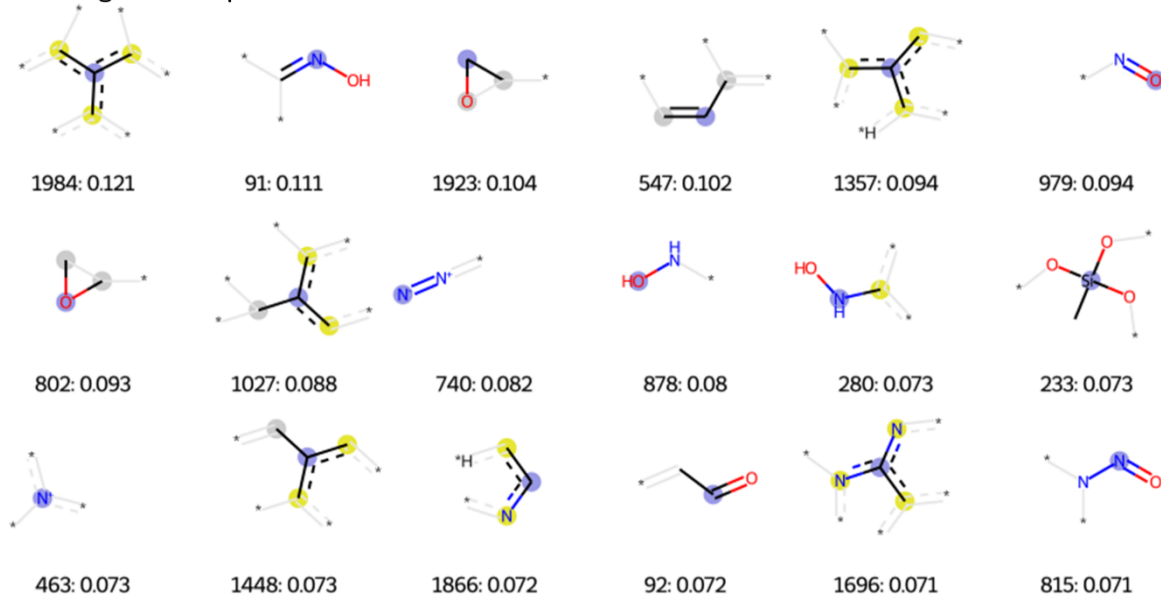
Activation of neuron 1-180 has been primarily linked to PAHs in the previous section. Atom environments indicative of PAHs consist of aromatic atoms belonging to two or more aromatic rings. The Top-18 bits are shown in Figure 8-17. The bit with the highest weight (1984) for this neuron encodes such an environment. Further examples in the Top-18 are the bits 1357 and 1696 (nitrogen heterocycles). Additionally, the bits 1750 and 875 possess appreciable weights while being outside of the Top-18. Inspection of the compounds causing the strongest activations also suggested that

epoxide and diol structures may contribute to a strong activation. The bits 1923 and 802, indicative of epoxide structures, were indeed found in the Top-18 bits supporting this assumption. The evidence for the diol structure seems somewhat controversial. Bit 1257, which is set on by any hydroxyl group in an aliphatic ring, has a negative weight. The related bit 849, which implies only single bonds in the vicinity of the hydroxyl group, has an even stronger negative weight. However, this bit is not set on by compounds found in the Top-12. In contrast, bit 1557, which implies a double bond at the carbon atom at two bonds distance from the hydroxyl group, has a positive weight. Careful examination of atom environments is required to identify which structure motifs do activate the neuron and which ones do not. It seems that hydroxyl groups in general do not activate (and in fact deactivate) the neuron, but ones next to a double bond positively contribute to neuron activation. For a more thorough analysis, all atom environments linked to hydroxyl groups would need to be investigated.

Neuron 1-180:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.70, 0.76, 0.90, 0.98

Weight to output neuron: 0.234



Other fingerprint bits:



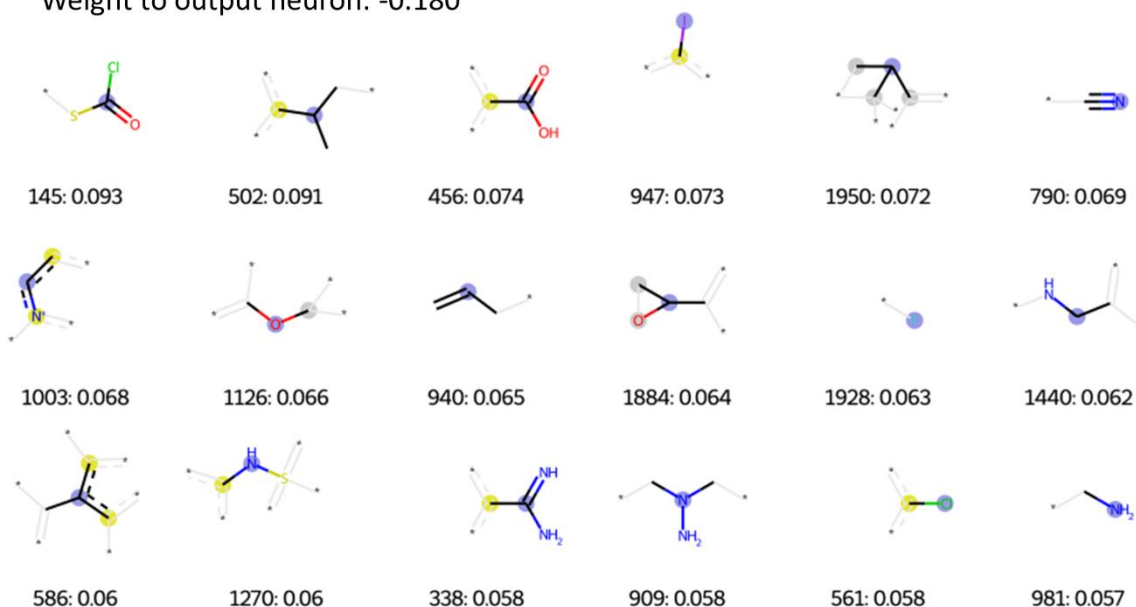
Figure 8-17 Analysis of bits with high weight for neuron 1-180. Details are given in the caption of Figure 8-14.

As described above, neuron 1-18 is linked to negative predictions. In Figure 8-18, the Top-18 fingerprint bits are presented. Three of the compounds in the Top-12 are sugar (i.e. pyranose) structures (1174, 2253, 11). The bit 1126 (8th highest weight for this neuron) is contained in the structures of 1174 and 2253 where a pyranose is attached to the rest of the structure via an ester bond. The bits 576, 1487 and 1381, which encode atom environments found in sugar structures, are not in the Top-18, yet all possess a positive weight, suggesting that sugar moieties do activate neuron 1-18. Another chemical feature identified as related to this neuron are pyridine rings. However, the bits 1603 and 1731 possess a very small positive weight and a negative weight, respectively. Hence, pyridine structures probably do not cause the activation.

Neuron 1-18:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.23, 0.17, 0.06, 0

Weight to output neuron: -0.180



Other fingerprint bits:

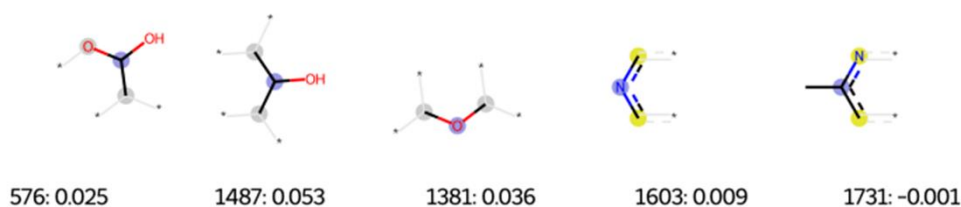


Figure 8-18 Analysis of bits with high weight for neuron 1-18. Details are given in the caption of Figure 8-14.

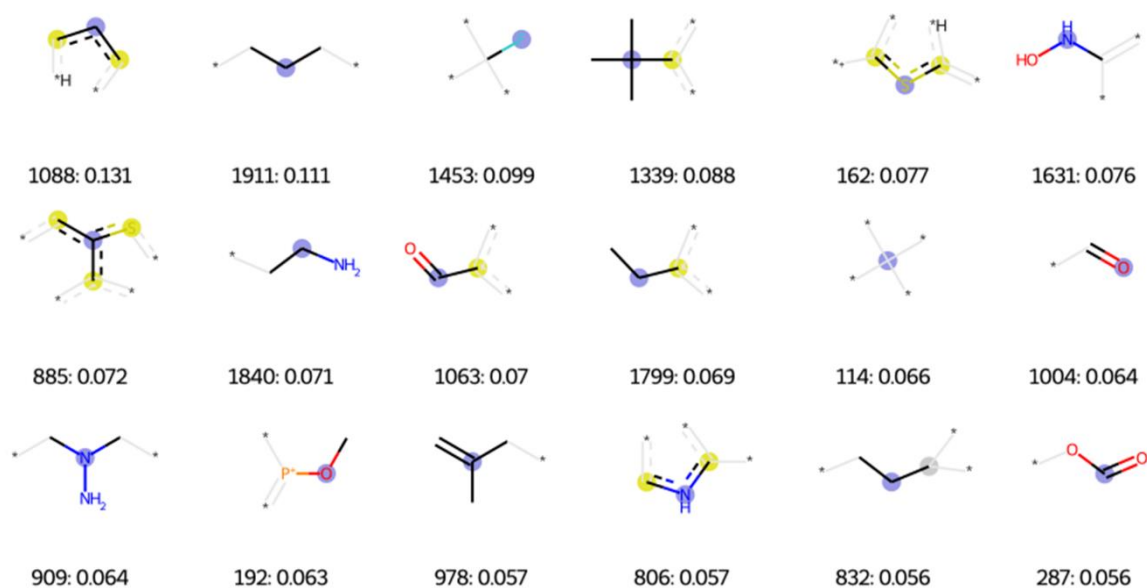
Esters of the phosphoric acid and alkyl chains were proposed as chemical features detected by neuron-128, which is also linked to negative predictions. The Top-18 bits are shown in Figure 8-19. Among

these, the bit with the second highest weight (1911) is contained in unbranched alkyl chains. Also, the bits 832, 294 and 794 possess positive weights and encode for alkyl chain atom environments. This seems to confirm alkyl chains as chemical features activating neuron 1-128. Similarly, the bits 192, 1214, 486 and 1716 encoding atom environments found in phosphoric acid esters have positive weights. Both esters of phosphoric acid and alkyl chains are hence confirmed as causes for activation of neuron 1-128.

Neuron 1-128:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.26, 0.20, 0.08, 0

Weight to output neuron: -0.262



Other fingerprint bits:

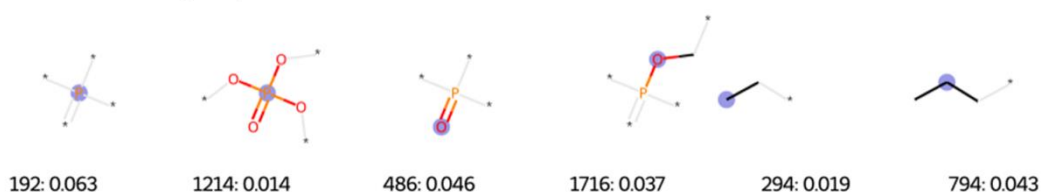


Figure 8-19 Analysis of bits with high weight for neuron 1-18. Details are given in the caption of Figure 8-14.

The learned weights of fingerprint bits provide information of the mathematical link between chemical input features and the activation of hidden neurons. However, as described above, many input bits (beyond the Top-18 bits) possess weights of comparable magnitude making it difficult to identify which chemical features are the most relevant causes for neuron activation. A particular chemical substructure may be encoded by several different fingerprint bits. The true effect a substructure has

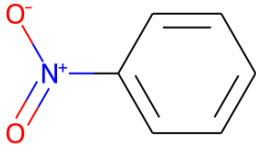
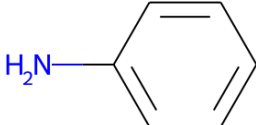
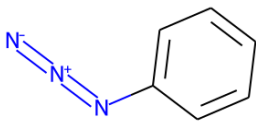
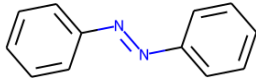
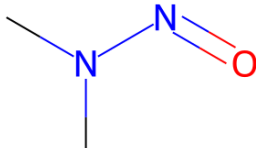
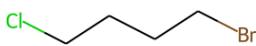
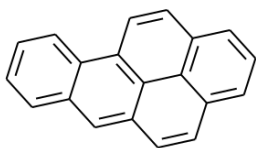
on the activation of a neuron depends on all the bits (with weights significantly different to zero) that encode this substructure. On the other hand, many of the bits in the Top-18 seem not to be linked to any structures strongly activating the neuron.

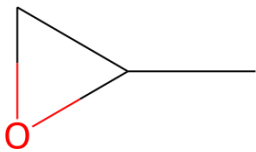
In some cases, the analysis of bit weights confirmed the substructures proposed as activating causes in the compound analysis, i.e., the inspection of compounds strongly activating the neuron, whereas in other cases, chemical substructures that are present in the Top-12 compounds, were not supported by the corresponding bits having high weights. It can be concluded that information obtained for fingerprint bit weights can help to identify causes of neuron activation proposed in the analysis of compounds with strong activation. However, a high weight for a single fingerprint bit may not be sufficient to establish a chemical substructure as a cause for neuron activation, since the chemical substructure may be associated with different bits with potentially conflicting weights. By inspecting the neuron activations for different compounds containing a substructure, conclusions about the role of the substructure may be drawn. Consequently, compound activations and weights for fingerprint bits can be considered as complementary sources of information.

8.3.6 Network-wide activation analysis for prototypical toxicophore compounds

In the previous section it was established that a particular hidden neuron may detect one (or more) chemical substructures potentially linked to mutagenicity. It was observed that certain chemical substructures may be learned in different hidden neurons. For instance, epoxide structures seemed to cause activation of neuron 1-43 as well as neuron 1-180. To better understand the behaviour of the network as a whole when a single structure is put in the network, neuron activations for several compounds containing a defined toxicophore were investigated. In Table 8-3, the numbers of activated hidden neurons for various thresholds are reported. All compounds are predicted as mutagenic by the network.

Table 8-3 Neuron activation by compounds with defined toxicophores. Shown are how many hidden neurons are activated given different thresholds.

	>0.05	>0.1	>0.25	>0.5
 <p>1: aromatic nitro</p>	169	135	40	0
 <p>2: aromatic amine</p>	160	105	15	0
 <p>3: azide</p>	170	133	43	4
 <p>4: diazo</p>	148	89	9	0
 <p>5: nitrosamine</p>	148	104	17	0
 <p>6: alkyl halide</p>	149	94	32	1
 <p>7: polycyclic aromatic hydrocarbon (PAC)</p>	157	113	28	1

 <p>8: epoxide</p>	167	106	20	1
--	-----	-----	----	---

It can be observed that for a relatively low activation threshold of 0.1, most of the compounds activate more than 100 hidden neurons. For the moderately high threshold of 0.25 the compounds activate between 9 (diazole) and 43 neurons (azide). Only one of the prototypical toxicophore compounds (azide) achieves an activation larger than 0.5 for more than one hidden neuron. All of the compounds studied here possess a defined chemical functionality linked to mutagenicity. These defined chemical features seem to be responsible for activation of a range of different hidden neurons. Since the model's prediction is a linear combination of the hidden neuron activations (followed by the application of the sigmoid function), the learned features (mathematically described by hidden neuron activation) are directly responsible for the model's prediction.

To further understand the relation between co-activated neurons, some exemplary neurons were inspected more closely. To that end, the four neurons which had an activation larger than 0.5 for the phenyl azide compound are compared. One of the four neurons is neuron 1-69 which was already identified above to be sensitive to azide structures. Table 8-4 shows the pairwise correlation coefficients for activation of the four neurons across all training compounds.

Table 8-4 Pairwise correlations of neurons activated by phenyl azide. Presented is the Pearson correlation coefficient, computed as described in section 8.2.4.

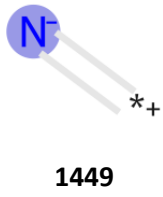
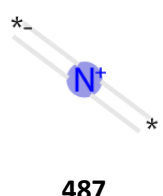

	1-69	1-382	1-441	1-473
1-69	1	0.395	0.553	0.486
1-382	0.395	1	0.246	0.526
1-441	0.553	0.246	1	0.343
1-473	0.486	0.526	0.343	1

All of the pairwise correlations are of low or medium magnitude. As can be seen in Figure 8-5B, very few pairs of neurons across the whole network possess correlation coefficients above 0.6. Correlation coefficients of low or medium size may indicate that the neurons detect the same chemical feature (in this case presumably azide), yet they may detect other chemical features along with azide. To gain a better understanding of what chemical substructures they may detect, the compounds with the

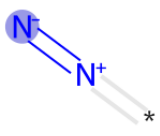
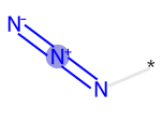
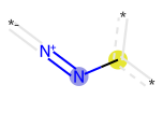
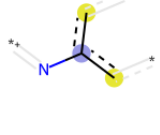

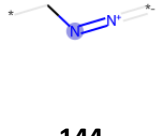
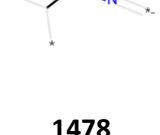
strongest activation and the fingerprint bits with the highest weights for the four neurons are compared below.

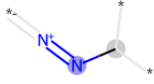
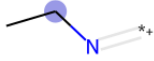

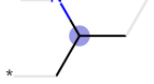
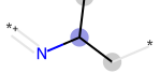
Weights for all of the bits of the Morgan fingerprint linked to the azide group were analysed for the neurons from Table 8-4. All of the bits presented in Table 8-5 are associated with azide groups, however, some of the bits may not be activated by all azide compounds. For instance, bit 13 includes an aromatic carbon atom and hence it is only set on by compounds where the azide group is directly attached to an aromatic ring. In contrast, the bits 144, 1478, 86, 1205, 1311 and 1563 are only activated by aliphatic azide compounds possessing a specific atom environment next to the azide group. The bits 1449, 487, 725, 740, 1838 are activated by all azides (aromatic or aliphatic). On the other hand, some of the bits are not specific for azide compounds which means that compounds lacking an azide group may set those bits. For example, bit 1854 is also activated by aromatic azo compounds, as it only implies an aromatic ring with a nitrogen atom bound to any other atom via a double bond.

Table 8-5 Detailed fingerprint bit weight analysis for neurons activated by phenyl azide. Bits encoding different parts of azide groups were identified and for the neurons 1-69, 1-382, 1-441 and 1-473 their weight and rank among all 2048 bits were determined.

	1-69		1-382		1-441		1-473	
	rank	weight	rank	weight	rank	weight	rank	weight
	34	0.0540	17	0.0706	22	0.0686	8	0.0740
	3	0.0867	3	0.1098	1	0.1048	75	0.0387
	68	0.0432	15	0.0614	38	0.0589	4	0.0793

Chapter 8: Exploration of chemical features learned in hidden neurons of neural networks

 <p>740</p>	39	0.0517	28	0.0614	37	0.0591	6	0.0765
 <p>1838</p>	76	0.0408	51	0.0489	48	0.0555	26	0.0534
 <p>13</p>	15	0.0696	13	0.0782	58	0.0499	15	0.0583
 <p>1854</p>	28	0.0590	14	0.0767	77	0.0422	503	0.0020
 <p>1709</p>	1223	-0	856	-0	809	0	1041	-0
 <p>144</p>	1653	-0.0078	1785	-0.0148	190	0.0251	196	0.0215
 <p>1478</p>	543	0.0176	445	0.0006	604	0.0001	388	0.0079

 86	242	0.0176	359	0.0050	615	0	182	0.0228
 1205	206	0.0204	1838	-0.0197	1436	-0.0036	1406	-0.0001
 1311	1555	-0.0036	1868	-0.0222	1761	-0.0007	1773	-0.0147
 1213	446	0.0054	308	0.0082	1341	-0.0124	1447	-0.0006
 1563	1670	-0.0084	1725	-0.0099	16660	-0.012	229	0.0185

From the table it can be seen that the bits common to all azide compounds are among the 100 bits with highest weights for the four neurons. Bit 487 is even in the Top-3 bits for three out of the four neurons. Since all azide compounds have all of those bits set on, all azide compounds will receive a strong positive contribution to the activations of all neurons from all those bits. Interestingly, the bits specific to aromatic azides (13, 1854) also possess relatively high weights for the different neurons and hence for aromatic azides an additional neuron activation occurs when compared to aliphatic azides. In contrast, the different bits associated only with aliphatic azides do not have high positive weights and hence do not significantly contribute to activation of those neurons. This may be due to the fact that the atom environments linked to aromatic azides are more frequent in the training set compared to the distinct aliphatic azides and hence present a clearer signal for the model to exploit during training. In summary, all four neurons possess high weights for fingerprint bits linked to the

azide group and hence it can be concluded that neuron activation is caused by the presence of the azide group in a compound.

In addition to weights of fingerprint bits, the activation of the four neurons by training compounds was analysed. Figure 8-20 displays the 100 highest activations as bar charts with those belonging to azide compounds highlighted in red. Also reported are the mean activations of all training compounds (n=5889) and azides (n=44).

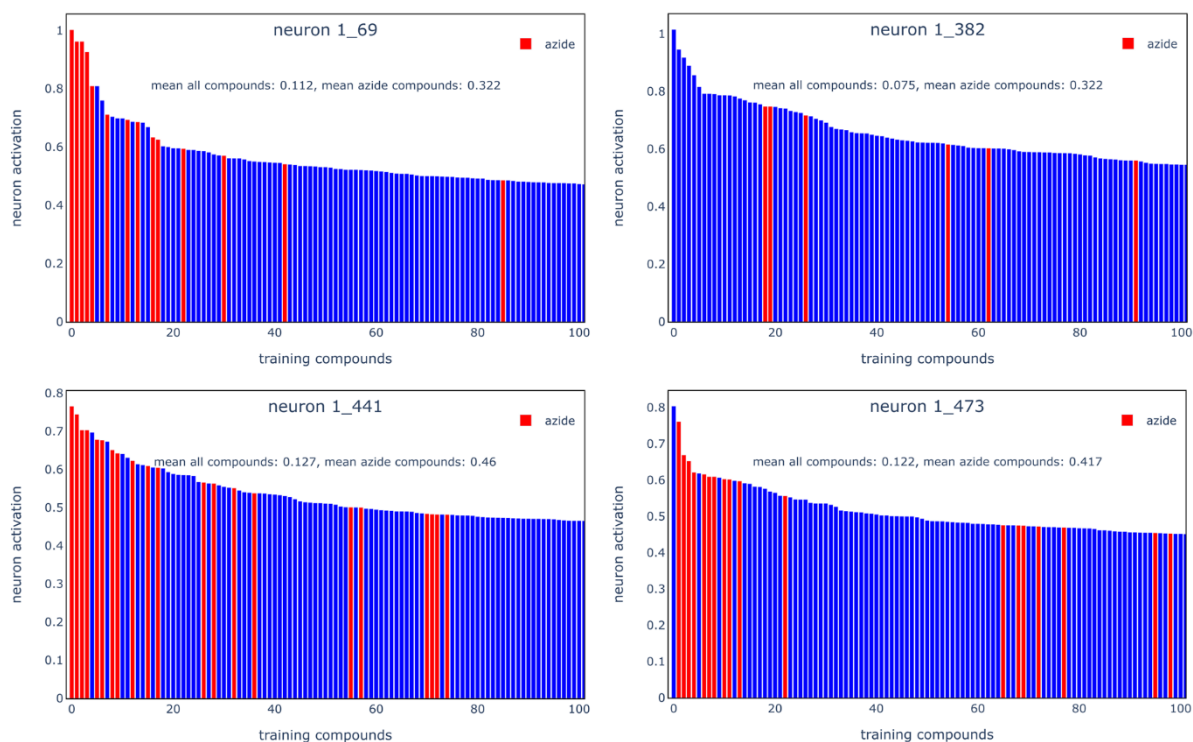


Figure 8-20 Neuron activation by azide compounds. For the neurons 1-69, 1-382, 1-441 and 1-473, the Top-100 activations in the training set are plotted with azide compounds highlighted.

For all neurons, the azide compounds have much higher mean activations compared to the full training set. Azide compounds outnumber non-azide compounds in the Top-10 for three of the neurons (1-69, 1-441 and 1-473). Both learned fingerprint bit weights and compounds with the strongest activation confirm azide as a learned chemical feature for those neurons. In contrast, for neuron 1-382 the highest ranked azide compounds are found at positions 19 and 20. While azide compounds clearly activate this neuron more strongly than randomly selected compounds, it appears that a different chemical feature might lead to the strongest activations for this neuron. To further explore neuron 1-382, the compounds most strongly activating the neuron are shown below in Figure 8-21.

Neuron 1-382:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.71, 0.74, 0.85, 0.93

Weight to output neuron: 0.254

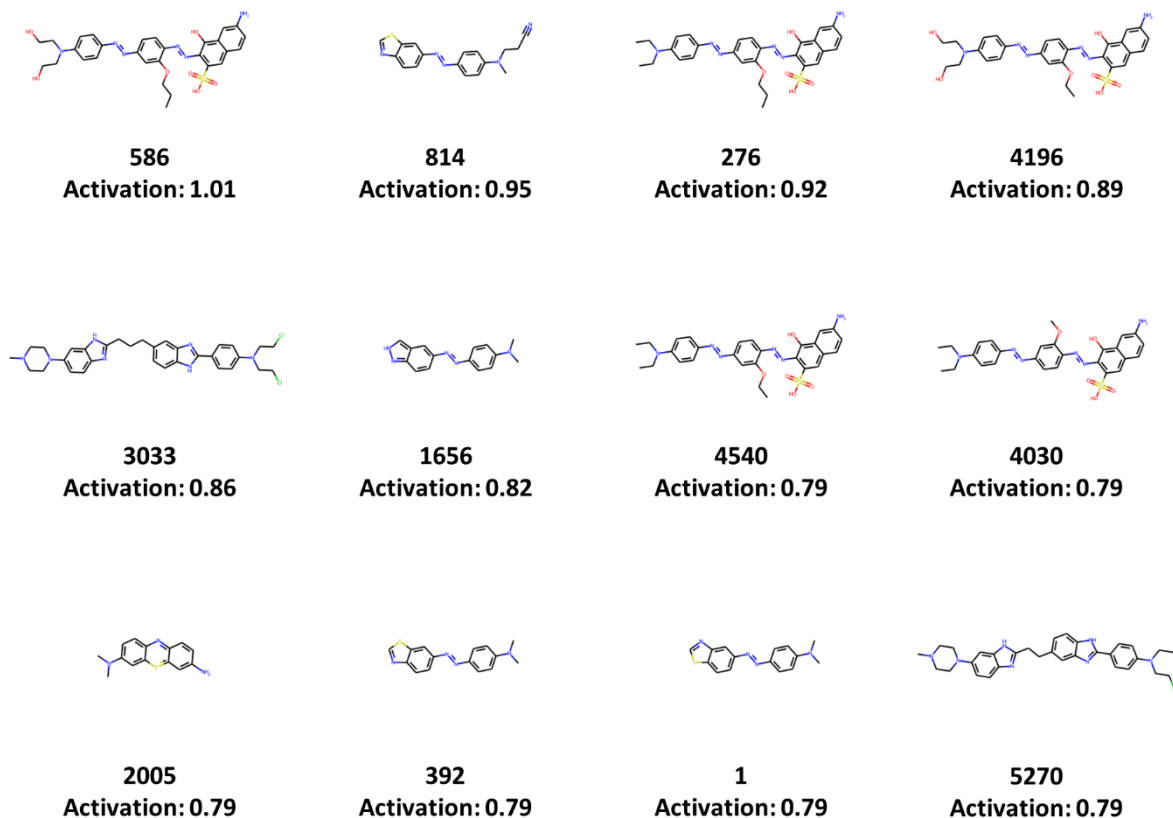


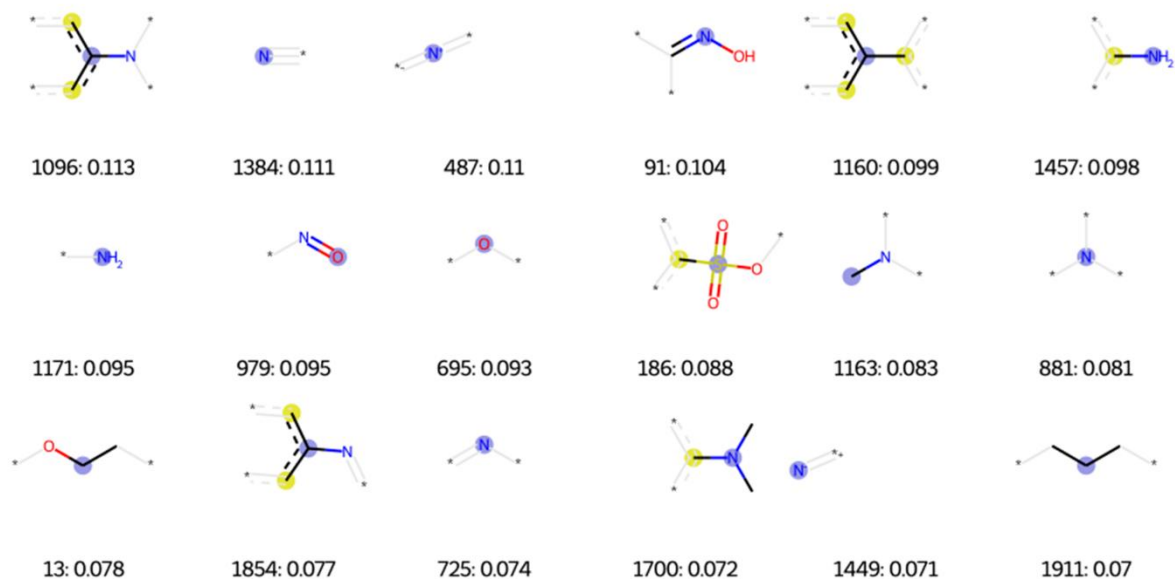
Figure 8-21 Analysis of compounds strongly activating neuron 1-382. Details are given in the caption of Figure 8-8.

Inspection of the Top-12 compounds for neuron 1-382 reveals that all compounds contain an aromatic amine, which is a known toxicophore for mutagenicity. Furthermore, nine of the 12 compounds are aromatic azo structures, another known toxicophore for mutagenicity. It seems that these are the most relevant chemical feature learned by neuron 1-382. To test this hypothesis, the fingerprint bits with the highest weights were analysed as shown in Figure 12. Many different bits linked to aromatic amines, azo compounds or azides can be found in the Top-18 bits. In addition, the bits 888 and 931 possessing significant positive weights encode atom environments linked to aromatic amines and azo compounds, respectively.

Neuron 1-382:

Confidence (thresholds: 0.05, 0.1, 0.25, 0.5): 0.71, 0.74, 0.85, 0.93

Weight to output neuron: 0.254



Other fingerprint bits:

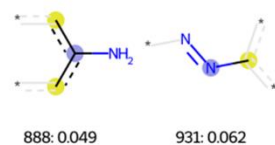


Figure 8-22 Analysis of bits with high weight for neuron 1-382. Details are given in the caption of Figure 8-14

In conclusion, neuron 1-382 has learned to detect aromatic amines, azo compounds and azide compounds, as fingerprint bits linked to all these chemical groups have been assigned high weights in the network. Many of the compounds most strongly activating neuron 1-382 combine an aromatic amine and an azo group in their structure. This provides further evidence that single neurons may be sensitive to different chemical groups. All of the four studied neurons are strongly activated by azide compounds, yet for neuron 1-382 other chemical groups seem to be more relevant learned features.

8.4 Conclusion

The explorative analyses conducted in this chapter confirm the hypothesis that hidden neurons may encode chemical features linked to the task the network was trained on (in this case Ames

mutagenicity). However, it was shown that a single neuron may detect more than a single chemical feature. This may be because learning several features in a single neuron does not reduce predictive performance of the network. If two distinct chemical features independently increase the mutagenic potential of compounds, then having a single hidden feature to detect either of them may be sufficient for the network to make correct predictions. Generally, the learning of weights in a network is not explicitly directed towards single neurons learning a single distinct chemical feature.

Using prototypical toxicophore compounds, it was also discovered that different hidden neurons may detect the same chemical feature. By analysing this phenomenon more deeply for neurons detecting azides, it was shown that those neurons may still represent somewhat different chemical features (for example, neuron 1-382 is also strongly activated by aromatic amine and azo compounds).

Thus far, the analysis of what chemical features cause strong activation of hidden neurons was performed manually by separately considering compounds that strongly activate hidden neurons and the fingerprint bits with highest weights. In the next chapter, a procedure to automatically extract the chemical features learned by hidden neurons is described. The approach consists of combining information gained from both compounds strongly activating neurons and the weights of fingerprint bits to hidden layers. It is acknowledged that weights linking interpretable fingerprint bits to hidden neurons are readily available only for the first hidden layer of a DNN model. However, it is expected that if the first hidden layer has been annotated with chemical features using this approach, then these annotations can provide information used to annotate the second hidden layer and so on. Moreover, other approaches to extract substructure may be possible (see Chapter 12).

Chapter 9 Automatic extraction of chemical features activating hidden neurons of neural networks

9.1 Introduction

In the previous chapter, it was demonstrated that strong activation of hidden neurons may be caused by chemical features linked to mutagenicity. These neurons typically are strongly linked to toxic predictions (high confidence). Hence, it can be concluded that the model's capability to make toxic predictions is dependent on chemical features detected in these neurons. Therefore, being able to determine the chemical features detected by hidden neurons may be useful to understand the model's predictions. The aim of this chapter is to develop a strategy to automatically find chemical features responsible for activation of hidden neurons using training compounds as well as the trained model. In particular, a set of chemical substructures is extracted to represent chemical features detected in each hidden neuron.

In Methodology, the general strategy to achieve the aim is described and theoretical background relevant to the employed techniques is provided. This is followed by an illustration of the application of the workflow to an exemplary hidden neuron. The same neural network model instance as in the previous chapter was analysed. Finally, the chapter is concluded by summarising key insights.

9.2 Methodology

In the following sections, details of the workflow for fragment extraction are described. The proposed approach includes several adjustable hyperparameters (expressions written in italics) for which initial values were selected based on insights from previous explorations. In the subsequent chapter, the success of the extraction is evaluated, and adjustments are made in order to improve the approach.

9.2.1 Inclusion of compounds and fingerprint bits

It was shown in the previous chapter that both compounds strongly activating a hidden neuron and fingerprint bits with high weights give insights into which chemical features are detected. For the manual analysis, a relatively low number of compounds (Top-12) and fingerprint bits (Top-18) were

analysed. However, it became clear that these are not sufficient to find all features responsible for neuron activation. For the automatic approach developed here, larger numbers of compounds and fingerprint bits were considered as described below.

The threshold for inclusion of compounds was determined by their activations. Since distributions for activation of training compounds vary between different neurons, neuron-specific thresholds were used based on the distribution of activation values. In particular, compounds having an activation larger than the mean activation for the neuron plus three (i.e. $ThreshCompound=3$) standard deviations were included.

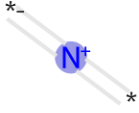
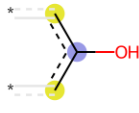
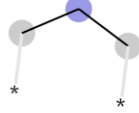
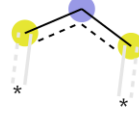
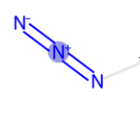

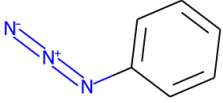
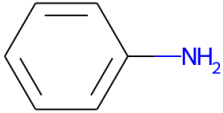
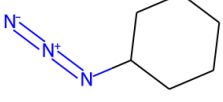
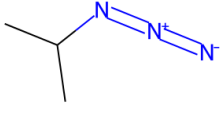
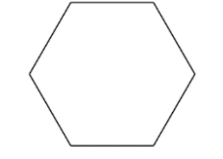
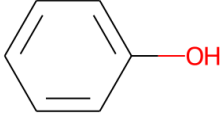
In the previous chapter, the fingerprint bits of various neurons were displayed sorted by their weights. A steady decrease in importance was found, which did not reveal a clear distinction between relevant and irrelevant bits for a given neuron. Therefore, the 95th percentile (i.e. 5% of bits with highest positive weight; $ThreshBits=0.05$) was selected for the inclusion of fingerprint bits. This corresponds to 102 bits being selected per neuron for $n=2048$ input bits.

9.2.2 Formal Concept Analysis for substructure extraction

Chemical substructures were obtained for each neuron from the selected compounds and fingerprint bits using Formal Concept Analysis (FCA). FCA was introduced by Rudolf Wille in 1982 as a tool to hierarchically organise concepts (Wille, 1982). A formal concept (FC) in this context is a triple (U, A, R) consisting of sets of objects U, attributes A and binary relations R (indicating whether an object u possesses attribute a). In a FC, all included objects share all the included attributes. Furthermore, the FC is closed in the sense that there are no further attributes shared by all the objects and, in turn, no further objects exist that possess all included attributes. A hierarchical lattice consisting of all existing FCs for a given dataset can be derived from this basic definition. To visualise this framework, a simple example related to the studied problem is provided. Objects are chemical compounds, and their attributes are bits of the Morgan fingerprint indicating the presence of a certain atom environment in the compounds. A small set of compounds and bits was selected to ensure the explanations and visualisations are suitable as an accessible introduction to FCA, inspired by the explanations given in a review on chemoinformatics applications of FCA and related approaches (Gardiner & Gillet, 2015).

Table 9-1 contains six different compounds as objects U and six different bits from a Morgan fingerprint as attributes A characterising the objects. Also displayed is which compounds set which bit of the Morgan fingerprint on (binary relations R).

Table 9-1 Binary relations between compounds and fingerprint bits as basis for FCA. A small number of compounds and fingerprints was selected to illustrate the foundations of FCA. The table indicates which compounds set on which of the selected fingerprint bits.

	 Bit 487	 Bit 745	 Bit 1028	 Bit 1088	 Bit 1838	 Bit 1854
 1	X			X	X	X
 2				X		
 3	X		X		X	
 4	X				X	
 5			X			
 6		X		X		

All FCs (eight in this case) defined by the triple (U, A, R) are displayed hierarchically in a Hasse diagram in Figure 9-1. In each box, the first line contains all compounds in the FC (the extent) and the second

line shows depictions of the atom environments encoded by the included fingerprint bits (the intent). The FC at the top of the diagram comprises the attributes shared by all compounds (which is an empty set in this case). Conversely, the FC on the bottom describes the set of compounds possessing all the attributes, which is also empty. For further explanations, the FC with the extent {1,3,4} and the intent {bit 487, bit 1838} is considered. This FC is a superconcept of the one at the top as its intent is a superset of the intent of the top concept (and equivalently, the extent is a subset of the extent of the top concept). This FC corresponds to the set of azide compounds in the dataset. Its intent, bit 487 and bit 1838 are set on by all azide compounds. Therefore, the FC can be considered as a concept capturing certain chemical characteristics. This FC is, in turn, a subconcept of the one with the extent {1}. Compound 1 is an aromatic azide compound and hence this represents a more specific concept than the more generic azides. This more specific chemical concept corresponds to more fingerprint bits shared by its members. The FC comprising all azides is also a subconcept of the one with the extent {3}. This FC corresponds to all azides also possessing bit 1028, indicating a pattern corresponding to a cyclohexane ring. As for the FC capturing aromatic azides, this is a more specific chemical concept compared to the one comprising all azide compounds. As can be seen, the FCA derives FCs with chemical meaning from information given as chemical compounds and fingerprint bits characterising the compounds.

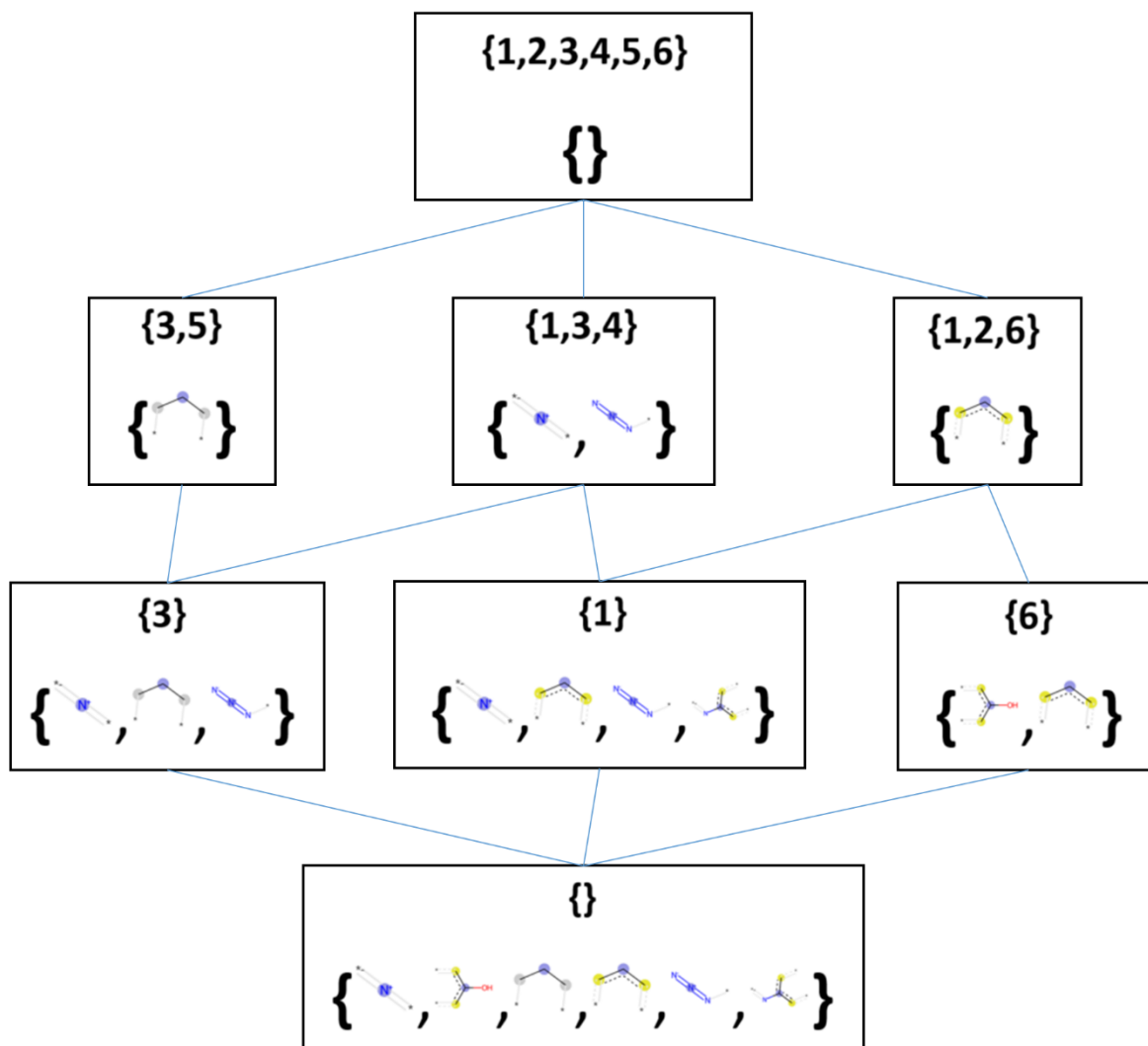


Figure 9-1 Hasse diagram depicting the lattice derived using FCA. Each box contains a FC consisting of an extent (set of compound identifiers in the first line) and an intent (set of fingerprint bits).

The capability to provide FCs with chemical meaning makes FCA a promising approach to identify chemical patterns causing activation of hidden neurons. Specifically, by applying FCA to the set of compounds strongly activating a neuron and the fingerprint bits possessing high learned weights for the same neuron, it is envisioned that the FCs that are obtained will correspond to certain chemical patterns associated with strong neuron activation. However, not all FCs may correspond to chemical features of interest. FCs may for instance contain just a single fingerprint bit and one bit alone may not be sufficient for neuron activation. Hence, only FCs whose intent reaches a certain relevance were considered. This was assessed by calculating the sum of weights of fingerprint bits in the intent. Only FCs with a sum of weights equal to half of the threshold for inclusion of compounds were included (i.e. *ThreshWeightFC*=0.5).

In addition, FCs were only considered if their extent exceeds a support threshold among all compounds selected for a given neuron. The rationale behind this is to avoid the identification of very specific chemical features (which do not generalise). The threshold initially selected was 0.2 (i.e. $ThreshSupport=0.2$; proportion of all compounds in the set selected for a given neuron).

In the next step, chemical substructures were extracted for each selected FC. For each compound in the extent of a FC, atoms matching any of the fingerprint bits contained in the intent were retrieved using the atom environments provided in RDKit. Then, fragments were obtained for each compound by connecting neighbouring atoms retrieved in the previous step. Since not all retrieved atoms for a single compound were necessarily connected, more than one fragment may have been obtained for each compound. However, only the most relevant fragment for each compound was retained. This was the fragment with the highest sum of weights for bits included in the fragment (which was not necessarily all bits of the intent). Moreover, to ensure only fragments causing a sufficiently strong activation on the neuron were included, the fragment was only retained if the sum of weights was higher than a given threshold. The same threshold as applied to the summed weights of all bits of the intent was applied. In addition, a fragment was only kept if the matches among compounds in the extent exceeded $ThreshSupport$ (see above). Finally, a structure was not kept if there was a more generic one with identical summed weight of fingerprint bits included. These steps are illustrated in the section below for an exemplary neuron.

For extracted substructures, additionally weights are assigned to individual atoms of that fragment. These weights are intended to indicate the contribution the atom makes to the whole fragment in order to make model explanations more accurate. For a given fragment there is a set of fingerprint bits each with a corresponding weight (network weight from input neuron to the hidden neuron). The weights assigned to the atoms of the fragment correspond to the summed weight of fingerprint bits that the atom is associated with. The obtained values for each atom are scaled so that the weights of all atoms of a fragment sum to 1. In the following chapter, it will be evaluated whether using this refined weighting scheme for individual atoms results in more accurate model explanations.

After a set of substructures has been extracted for a given neuron, the fragments are organised hierarchically in a network according to substructure-superstructure relationships. This facilitates the analysis of the fragments found for each neuron. The approach of hierarchically organising chemical fragments is comparable to self-organising hypothesis networks (SOHN) (Hanser et al., 2014). SOHN provides a framework to hierarchically organise structure-activity information as network. The basic element of the obtained hypothesis network is an individual hypothesis. A single hypothesis may consist of a structural or physicochemical feature related to a certain trend about a bioactivity

endpoint. Individual hypotheses that form the network may be derived from a variety of sources (e.g., a ML model or expert knowledge). The hierarchical networks may be used to analyse SAR knowledge or to build a predictive model. In the work presented here, the networks do not consist of different hypotheses for SAR trends. Instead, each node represents a chemical fragment that activates a hidden neuron (extracted according to descriptions above) and the network organises all of the chemical fragments extracted as strongly activating a given hidden neuron. The algorithm used to obtain the network is briefly described in the following paragraphs.

The fragments extracted for all compounds were organised according to superstructure-substructure relationships and these relationships were established by performing substructure matches between extracted fragments. This is distinct from SOHN where these relationships are established using memberships of data instances (matching a certain substructure or having a certain property). As done in SOHN, all fragments extracted for a neuron are initially organised in a single network where the root is a generic 'hypothesis' (i.e. a substructure of all possible fragments). In this work, the network afterwards was separated into 'subnetworks'. A subnetwork contains a single root (which is an actual substructure) and all its descendants (i.e. all superstructures of the root). The full network and the subnetworks have similarities to a rooted tree, yet the difference is that nodes in the (sub-)network may have more than one parent node.

The root of the network is initialised to be a generic 'hypothesis' or fragment (which matches all possible fragments) as mentioned above. New fragments are added by comparing them to the ones already in the network. To add a new fragment, the most specific parent fragments were found. These are substructures of the new fragments which have no child fragments that are also substructures of the new fragments. The new fragment was added to the network as a child of its most specific parents. Then, the most generic children of the fragment were found. These are superstructures of the new fragment that do not have more generic substructures being also superstructures of the new fragment. The new fragment was inserted as a direct parent of the most generic children fragments found in this way. Once a new fragment has been added to the network, redundant connections between its parent and child fragments may need to be removed. Following these steps, the network was grown so that all extracted fragments for a given neuron were included in an iterative fashion (one fragment added at a time). The exact implementation for the network construction closely follows the description of the SOHN algorithm in the original paper (Hanser et al., 2014) although, as stated above, the purpose of the network is different from its original use.

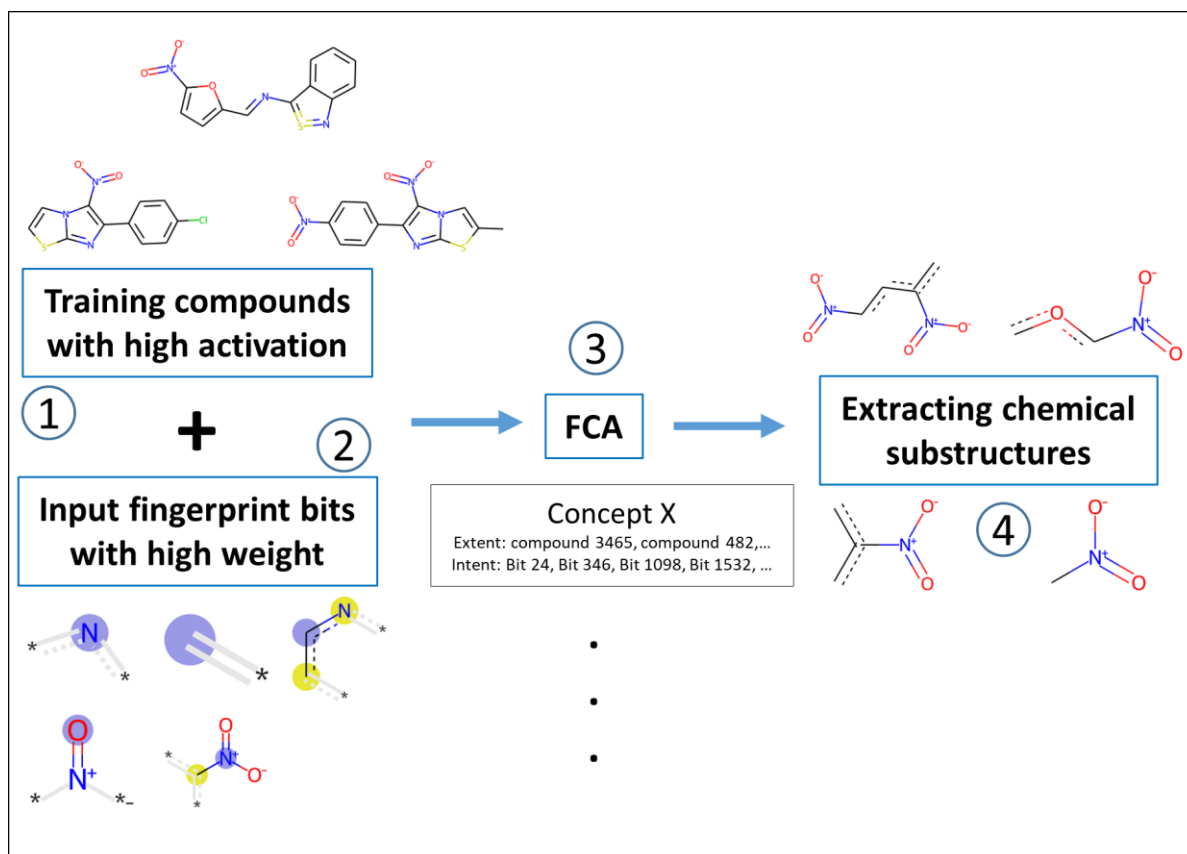
After the complete network has been obtained, it is separated into subnetworks where the root structures are genuine fragments instead of the generic 'fragment'. This was done to facilitate the

subsequent analyses, as the root structures form a good starting point to inspect what substructures were extracted for a neuron. Exemplary subnetworks obtained according to this procedure are presented in section 9.3 below.

It is worth noting that the computational cost of building the network increases strongly with increasing number of extracted substructures. Therefore, the maximum allowed number of substructures extracted per neuron was chosen to be 200. In order to ensure that the most relevant substructures are extracted also for neurons when this limit stops the substructure extraction, the FCs are considered in decreasing order of support for the extent.

9.2.3 Summary of the workflow

Figure 9-2 summarises the approach to obtain chemical substructures that strongly activate a given neuron.



- ① Pick compounds with activation above $\text{mean} + \textit{ThreshCompound} \times \textit{SD}$
- ② Pick *ThreshBits* (fraction of 1) bits of the fingerprint
- ③ Do FCA with compounds and bits, retain concepts with extent support above *ThreshSupport* (fraction of 1) and *ThreshWeightFC* (fraction of *ThreshCompound*)
- ④ For each compound in the concept retain only one substructure and retain it only if the weight of included bits is above *ThreshWeightFC* and the substructure is supported by at least *ThreshSupport* (fraction of 1) of the compounds in the extent of the FC.

Figure 9-2 Overview of the developed method for extracting fragments responsible for neuron activation. The expressions in italics represent adjustable parameters.

9.3 Illustration of the workflow

In the first step of the workflow for substructure extraction, compounds and fingerprint bits were selected for each neuron. While the number of fingerprint bits was fixed (95th percentile), the compounds selected depended on the distribution of activations observed for a given neuron. Figure 9-3 shows the number of compounds selected per neuron. The maximum number of selected

compounds per neuron was 219, while the median was 110. No compounds were selected for 31 of the 512 neurons and hence no chemical structures were extracted for those neurons.

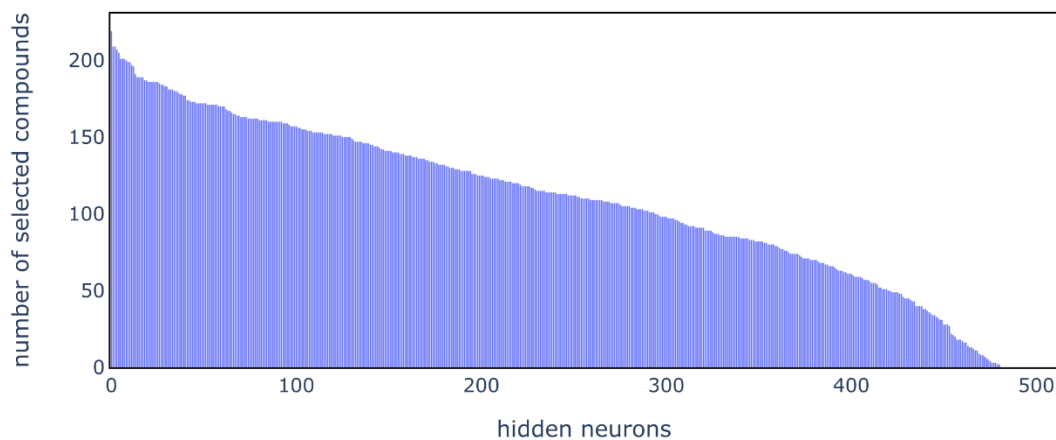


Figure 9-3 Selected compounds per neuron. The plot shows the number of selected compounds (activation above threshold) for each hidden neuron sorted in descending order.

For neuron 1-69, the sets of selected compounds and fingerprint bits are visualised in Figure 9-4. 51 compounds possess a larger activation than the threshold (0.53 for this neuron). Naturally, only fingerprint bits set in any of the selected compounds are of relevance for the FCA. Of the 95th percentile of fingerprint bits for neuron 1-69 (102 bits), only 52 bits appeared in the selected compounds. This is reflected in Figure 9-4B, where the bits appearing in the selected compounds are displayed in red.

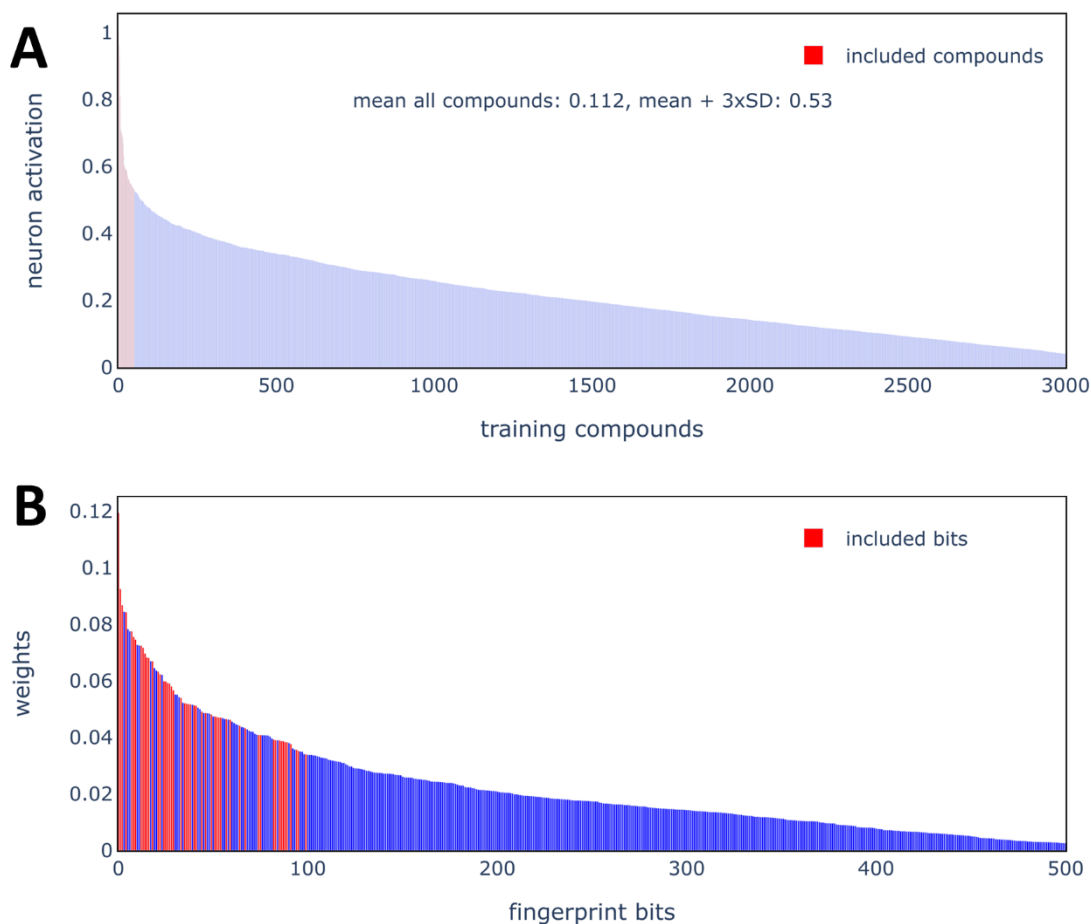


Figure 9-4 Selected compounds (A) and fingerprint bits (B) for neuron 1-69. Training compounds and fingerprint bits are sorted according to neuron activation and weight, respectively. In **B**, bits set in the selected compounds are highlighted in red. For visualisation purposes, only the Top-3000 compounds (out of the 5889 training compounds) and the Top-500 bits (out of 2048 bits) are included in the plots.

Performing FCA on the obtained sets of compounds and fingerprint bits for neuron 1-69 yielded a lattice of 441 formal concepts. In Figure 9-5, a scatter plot shows the support of compounds in the extent (among the set of all selected compounds for the neuron) and the potential to activate the neuron (as the summed weight of all bits in the intent) for all FCs. Red lines have been added to the scatter plot to indicate the thresholds applied to filter the FCs based on support and summed weight of the fingerprint bits. The selected thresholds were 0.2 for support and half of the threshold applied earlier for inclusion of compounds based on neuron activation (0.265). Hence, only concepts in the upper right section of the plot (45 of 441 FCs) were retained to extract chemical substructures.

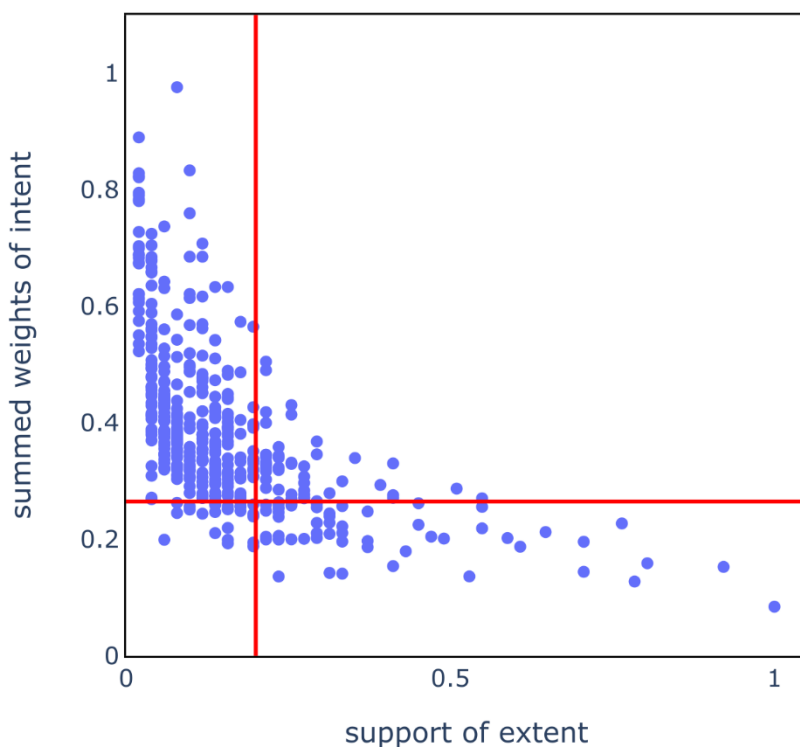


Figure 9-5 Support and potential for neuron activation of FCs. The support of the extent indicates what proportion of the 51 included compounds are in the extent of the particular FC. The summed weight of the intent corresponds to the sum of weights for the fingerprint bits forming the intent of the FC. The red lines in the plot indicate the thresholds for inclusion applied to this neuron. Only FCs above the horizontal AND to the right of the vertical line are included. For clearer separation of the dots in the plot, the y axis is cut off at a summed weight of 1.1. The only formal concept with a summed weight above this is the one containing all bits (summed weight: 2.88), which has a support of 0.

The extraction of chemical substructures is demonstrated for a single FC of neuron 1-69. The extent of this particular FC contains 11 compounds and hence it passed the selected threshold for the minimal support (0.2×51 compounds). The bits in its intent sum to 0.505 and hence it is the FC with the highest summed weight that passed the threshold for minimal support. In Figure 9-6, the atom environments corresponding to the bits in the intent are displayed. Six of the eight bits are associated with azide groups. The bits 1088 and 1750 encode atom environments linked to an aromatic ring. Based on the intent, this FC corresponds to aromatic azide compounds.

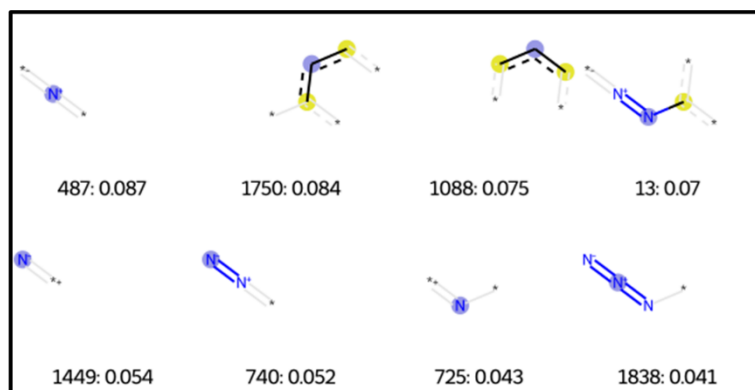


Figure 9-6 Fingerprint bits forming the intent of the selected FC.

Figure 9-7 shows the compounds in the extent with the atoms that match any of the bits highlighted. In all structures, the azide group as well as at least one atom of the connected aromatic ring are highlighted. All of the compounds possess at least two unconnected fragments. The fragments not connected to the azide group comprise further aromatic rings in the compounds. As described above, only the fragment with the highest summed weight of involved bits was retained for each compound. In all cases this was the fragment containing the azide group.

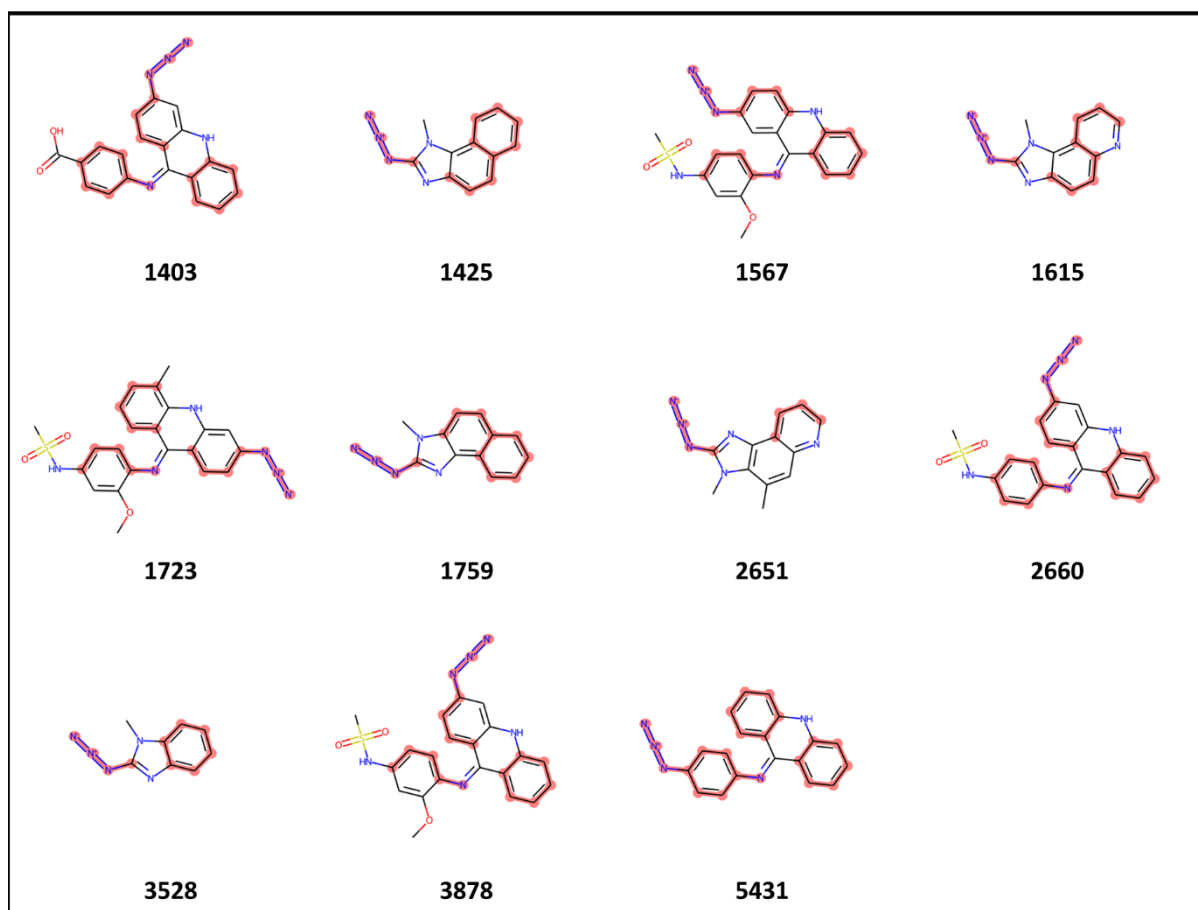
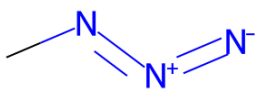
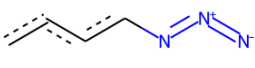
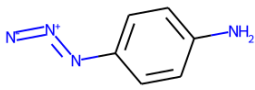


Figure 9-7 Compounds forming the extent of the selected FC. Highlighted in red are all atoms being part of any of the bits in Figure 9-6.

The extracted substructures may be identical for some compounds. For instance, compounds 1425, 1615, 1759, 1651 and 3528 all yielded an azide group attached to a single aromatic carbon atom. In total, three distinct substructures were obtained for the investigated FC. These substructures are shown in Table 9-2. It can be seen that the substructures are closely related. Sub0 is a substructure of Sub1 and Sub2, while Sub1 is a substructure of Sub2. Hence, the structures represent related chemical fragments of different specificity. Sub0 matches azide groups attached to any aromatic carbon, whereas Sub1 requires 4 aromatic carbon atoms and matches phenyl rings but not imidazole-like heterocycles (as in compound 1425). While Sub0 matches all compounds in the extent of the FC, Sub1 corresponds to a higher summed bit weight. Sub1 therefore causes a stronger activation of neuron 1-69. Sub2 matches only compound 5431 and interestingly, while being more specific than Sub1, it comprises the same fingerprint bits. This is because the single nitrogen atom in the fragment of compound 5431 (like the uncharged nitrogen in the azide group) sets bit 725 on. Sub2 represents a

very specific substructure supported by a single compound of the FC. As described in Methodology, Sub2 was not retained as its support among compounds in the FC was too low.

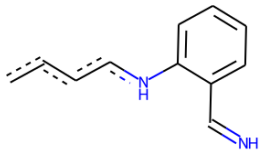
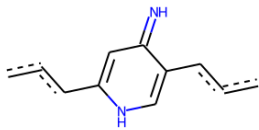
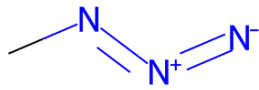
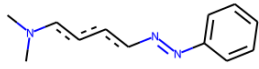
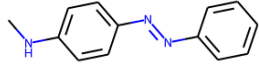
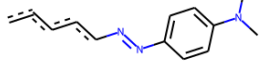
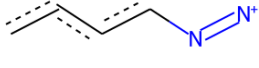
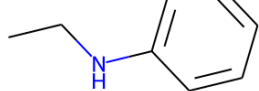
Table 9-2 Extracted substructures for the selected FC.

Substructure	Supporting compounds	Supporting bits	Summed bit weight
 <p>Sub0</p>	1403, 1425, 1567, 1615, 1723, 1759, 2651, 2660, 3528, 3878, 5431	13, 487, 725, 740, 1449, 1838	0.346
 <p>Sub1</p>	1403, 1567, 1723, 2660, 3878, 5431	13, 487, 725, 740, 1449, 1750, 1838	0.430
 <p>Sub2</p>	5431	13, 487, 725, 740, 1449, 1750, 1838	0.430

None of the extracted substructures included bit 1088. Matches with bit 1088 only occurred in the fragments with lower summed weight for a given compound (phenyl rings not connected to the azide fragment). The presence of such rings in compounds leads to even stronger activation of the neuron, yet this information was not retained since the extraction method does not allow for disconnected fragments. Despite this loss of information, the extracted substructures capture the major part of contributions to neuron activation for this FC. If bit 1088 is part of any connected fragments strongly activating the neuron, it is possible that this may be discovered in a different FC.

Applying the above steps to all retained FCs of neuron 1-69 yielded a total of 25 unique fragments. To organise this information, the fragments were arranged into subnetworks according to substructure-superstructure relationships as described in Methodology. For neuron 1-69 this procedure yielded eight subnetworks, of which four (Subnetwork 4, Subnetwork 5, Subnetwork 6 and Subnetwork 8) contain just a single fragment. Table 9-3 gives an overview of the subnetworks.

Table 9-3 Subnetworks of extracted substructures for neuron 1-69. Shown are the root fragment, the size of the subnetwork (number of fragments) and the range of summed weight the contained fragments possess.

Subnet work ID	Root	Size	Range of summed weights
1		14	0.271-0.346
2		14	0.271-0.346
3		3	0.346-0.490
4		1	0.319
5		1	0.319
6		1	0.319
7		5	0.284-0.490
8		1	0.277

Subnetwork 3 consists of substructures with azide groups (including Sub0 and Sub1 from Table 9-2). This subnetwork is shown in Figure 9-8. The terminal leaf structure of Subnetwork 3 is a very specific azide fragment with several aromatic rings and an imine group. The root structure of Subnetwork 7 (shown in Figure 9-8), contains a neutral and positively charged nitrogen connected by a double bond attached to an aromatic ring. The motif of the nitrogen atoms is part of the azide group. In fact, two child fragments contain the azide group, and these fragments overlap with Subnetwork 3.

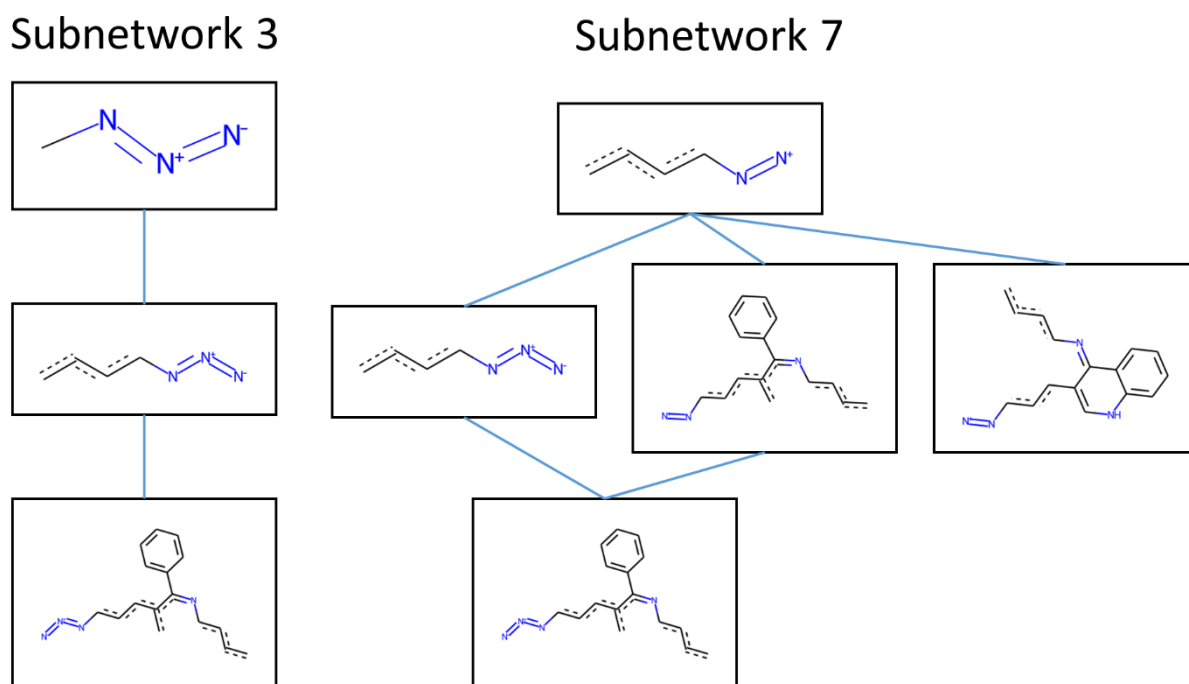


Figure 9-8 Substructures forming Subnetwork 3 and Subnetwork 7.

Subnetwork 1 and Subnetwork 2 were formed from distinct root structures, yet they are very closely related. While each subnetwork consists of 14 fragments, there is significant overlap between the subnetworks (16 unique fragments in the union of both subnetworks). Both subnetworks contain substructures related to the acridine scaffold with an imine group attached to the central ring. More specific substructures in both subnetworks include additional aromatic rings and various nitrogen groups. In contrast, the more generic substructures that do not encode the full acridine ring system match compounds containing heteroatoms in the acridine-like scaffold.

The roots (and only fragments) of Subnetworks 4, 5 and 6 represent aromatic diazo structures with an additional amine group (secondary or tertiary amine) attached to one of the aromatic rings. Diazo structures themselves are a known toxicophore for mutagenicity, yet for neuron 1-69 an additional amine group seems to be necessary to strongly activate the neuron (i.e., to be above the thresholds defined for inclusion of substructures). The root of Subnetwork 8 is N-ethylaniline, indicating that a generic secondary aromatic amine causes activation of neuron 1-69.

Subnetwork 1

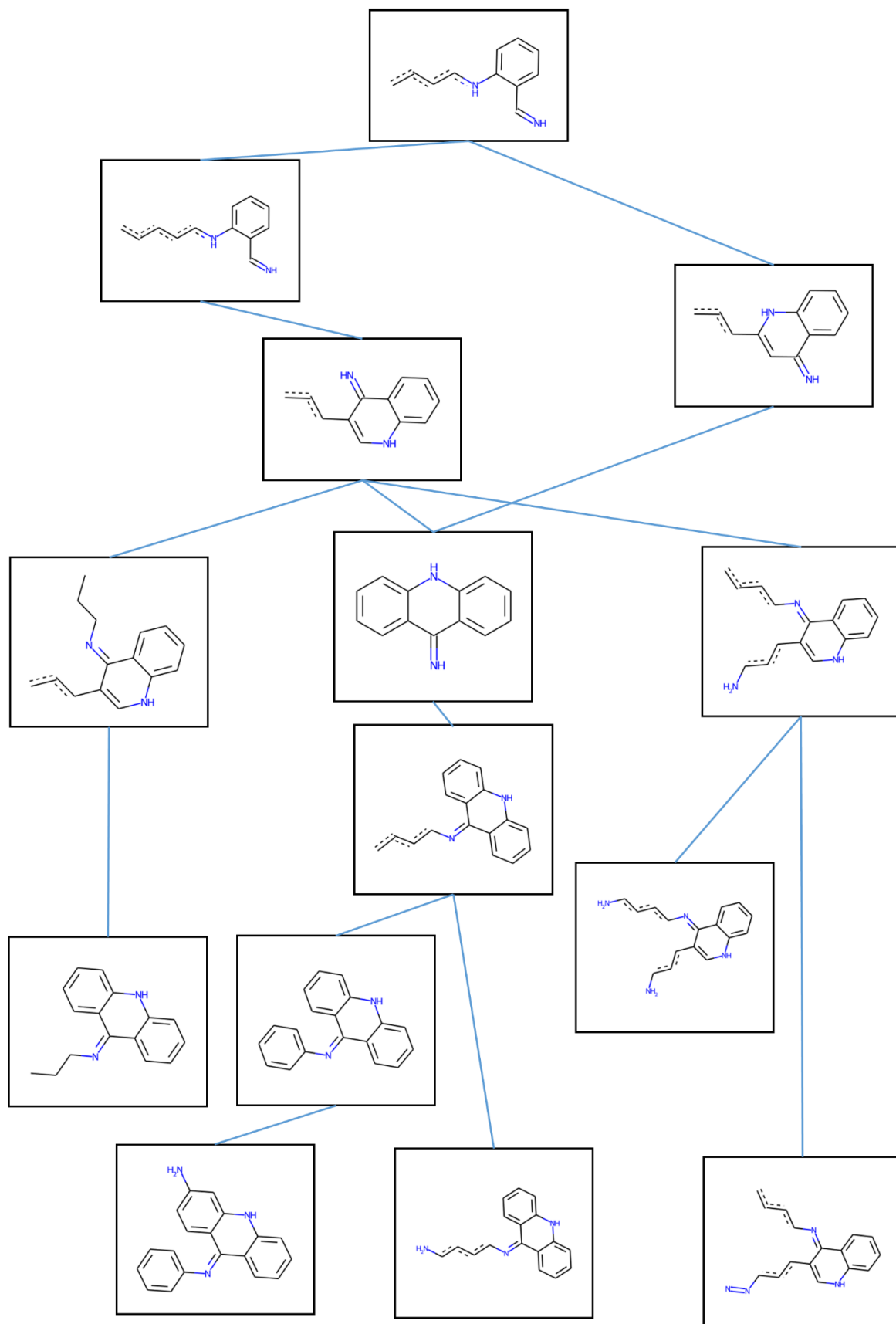


Figure 9-9 Substructures forming Subnetwork 1.

Subnetwork 2

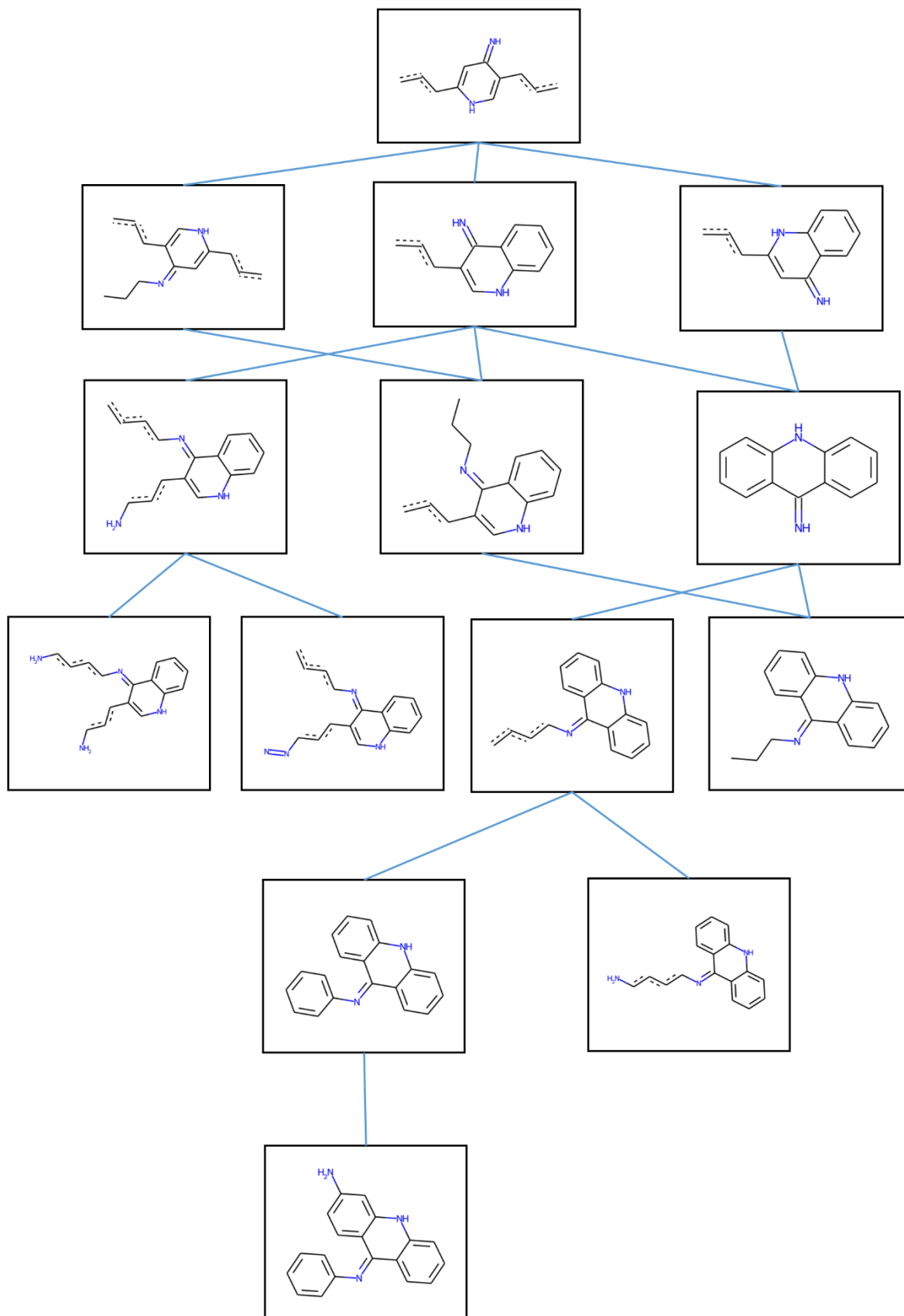


Figure 9-10 Fragments forming Subnetwork 2.

9.4 Conclusion

This chapter introduced a method to automatically extract chemical features in the form of substructures that are responsible for hidden neuron activation. The network structure supports the analysis of extracted substructures by storing them in an organised manner. Fragments within a subnetwork can be considered as more specific or generic versions of a particular chemical concept, whereas different subnetwork represent different (yet potentially closely related) chemical patterns. Furthermore, the network structure facilitates the matching of test compounds (to identify the cause for their neuron activation) with the fragments. If a test compound does not match the root structure of a subnetwork, it will not match any of the fragments in the subnetwork. More generally, if a test compound does not match a particular fragment within the network, it will not match any of its child nodes.

By focussing on neuron 1-69, it was demonstrated that this approach can yield chemical features responsible for mutagenicity (azide and acridine). The extracted chemical substructures may be analysed directly to discover toxicophores for the modelled endpoint or they can be mapped onto test compounds in order to explain predictions made by the model for those compounds. In the following chapter, the extent to which the extracted fragments correspond to known toxicophores for mutagenicity will be evaluated as well as the extent to which the explanations provided for predictions of test compounds match their known causes of toxicity. These evaluations are used to measure the quality of the method developed here for substructure extraction and to design modifications to the method in order to improve its quality. Modifications may include changes to the introduced parameters or more fundamental changes to individual steps.

Chapter 10 Evaluation and optimisation of the model explanation approach

10.1 Introduction

In Chapter 9, a technique to automatically extract chemical substructures responsible for the activation of hidden neurons was presented. The purpose of the technique is to enable interpretation of the neural network model on a global and on a local level. Global interpretability means understanding which chemical features are overall the most relevant ones for a model, whereas local interpretability refers to explaining individual predictions made by the model.

In the previous chapters, a neural network model trained on experimental mutagenicity data was analysed. However, the true reason for toxicity is not always known. To facilitate evaluation of local and global interpretability, in this chapter, a model is trained on a dataset that has been labelled using the Derek Nexus software (Marchant et al., 2008) which detects the presence of structural alerts for mutagenicity. Global interpretability is then evaluated by comparing extracted substructures to chemical substructures recorded in Derek as being responsible for mutagenicity (i.e. toxicophores). Local interpretation requires the extracted substructures to be mapped to test compounds in order to explain the predictions. In this case, the atoms responsible for a positive label are known (defined by the Derek alerts), the concordance of the model's explanation to the Derek alerts can be evaluated.

First, the approach for substructure extraction presented in the previous chapter was evaluated globally and locally. Based on gained insights, changes were then made to the extraction workflow with the aim to improve its performance measured as global and local interpretability. Next, the modified workflow was evaluated on test compounds not previously used for model training, substructure extraction or validation of the explanation model. Finally, the optimised method for extracting substructures and explaining predictions was applied and evaluated on a model trained on experimental Ames labels, as it is experimental toxicity labels that are relevant in practical settings.

10.2 Methodology

10.2.1 Dataset

The Ames dataset investigated in Chapters 8 and 9 was used. The experimental labels were replaced by labels determined using the Derek Nexus software and any structural alerts identified by Derek were appended to the compound record. Of the 7662 unique original structures (the union of the training, validation and test sets), 7336 could be processed in Derek Nexus. The remaining structures were discarded. No changes were made to the membership of compounds in the different splits of the data (training, validation, test). The Derek Nexus software returned a SDF (structure-data file) containing each compound in a MolFile (connection table) format, any alerts matched by the compound (more than one possible) and which atoms were responsible for the alert(s) being matched. The Derek Nexus software added explicit hydrogen atoms to the chemical graphs and these were removed using RDKit prior to model building. According to its internal rules, the Derek Nexus software labels compounds as “INACTIVE”, “EQUIVOCAL”, “PLAUSIBLE” or “PROBABLE”. The latter three categories indicate the presence of one or more alerts and those compounds were labelled as the “positive” (i.e. toxic) class, with the others labelled as “negative” (i.e. non-toxic). Across the whole dataset, 3789 (0.516) compounds form the toxic class and 3547 (0.484) compounds form the non-toxic class. 105 different Derek alerts were set across the whole dataset, 102 of these were set at least once in the training set. To refer to specific alerts, they were assigned IDs (Alert1 to Alert105; these IDs are different from those used in the Derek Nexus software).

10.2.2 Model training

A neural network model was trained using the Derek derived labels indicating whether or not a compound activated a Derek alert for mutagenicity. As for the model using experimental Ames labels, the model contained one hidden layer with 512 neurons. To find optimal values for other hyperparameters, a grid search was conducted. All considered hyperparameters are shown in Table 10-1.

A model was trained for a maximum of 10 epochs on the training set and evaluated after each epoch on the validation set. For a given hyperparameter combination, the model instance (after a particular training epoch) with the best performance (early stopping) was compared to models obtained for other hyperparameters. Among those models, the best performing model according to ROC-AUC

(*best_model* in Table 10-1) was selected to be used in this chapter. In addition, a second model instance using dropout during training was selected (*dropout_model* in Table 10-1). This was motivated by a conceptual advantage of dropout models relevant to the study. Compared to models not using dropout during training, individual neurons are expected to learn more meaningful features as the emergence of co-adaptations between neurons is restricted. A neuron with clearer defined features should facilitate the extraction of relevant chemical fragments. The model instance was selected by choosing the instance with the best performance among those using dropout.

Table 10-1 Model instances used in the chapter. Shown are the hyperparameters of the model along with all tested values.

Hyperparameter	Tested values	<i>best_model</i>	<i>dropout_model</i>
Batch size for optimisation	16, 32, 64	16	32
L2 regularisation of neuron weights	0, 0.00001, 0.001	0.001	0.001
Dropout	0, 0.2, 0.5	0	0.5
Learning rate	0.0001, 0.00033, 0.001	0.001	0.001

10.2.3 Local evaluation

10.2.3.1 Attribution methods

A common strategy to interpret QSAR models is to highlight atoms in the chemical graph of a test compound according to their importance for the compound's prediction (Jiménez-Luna et al., 2020). For example, different techniques have been developed to highlight atoms based on the importance of input features to the model. Integrated gradients (IG) applied to Morgan Fingerprint is one such established technique and was used within this study as a baseline (referred to as *IG_input*). The method developed here is called *IG_hidden* and is based on considering the importance of hidden neurons and the substructures extracted for those neurons. The results obtained using the *IG_hidden* method were compared to the baseline.

10.2.3.2 *IG_input*

IG determines an importance (positive or negative) for each input feature towards the prediction of a given test compound. In this project, the implementation of the method (*IntegratedGradient* class) provided in the Python library Captum (version 0.4.0) was used. Notably, the attributions for each

individual feature sum to the prediction (precisely, this is the difference in predicted probability relative to the empty bit vector baseline) made for the test compound (Sundararajan et al., 2017). This follows from the fact that gradients of features are integrated along the path from a baseline to a compound's bit vector as described in Chapter 7. The attributions obtained for features (i.e. bits of the Morgan FP) were mapped to the atoms in a procedure comparable to a previous study (Preuer et al., 2019). Firstly, all atom environments belonging to a given fingerprint bit were collected. Multiple environments for a given bit may exist due to multiple occurrences of identical environments in a compound or due to bit collisions (i.e. different environments map to the same position in the bit vector). The total attribution for a given bit was shared equally between all associated atom environments (i.e. if two different environments map to the same bit, each environment receives half of the total attribution for the bit). Then, the total attribution for a given environment was shared equally by all atoms being part of the environment. Atoms may receive positive or negative attributions from different bits of the fingerprint. All attributions for a given atom were summed to obtain the final attribution for that atom. To simplify the calculations, only fingerprint bits with an attribution of at least 1% of the most important feature (positive or negative) were considered. An illustration of the method is provided in Figure 10-1.

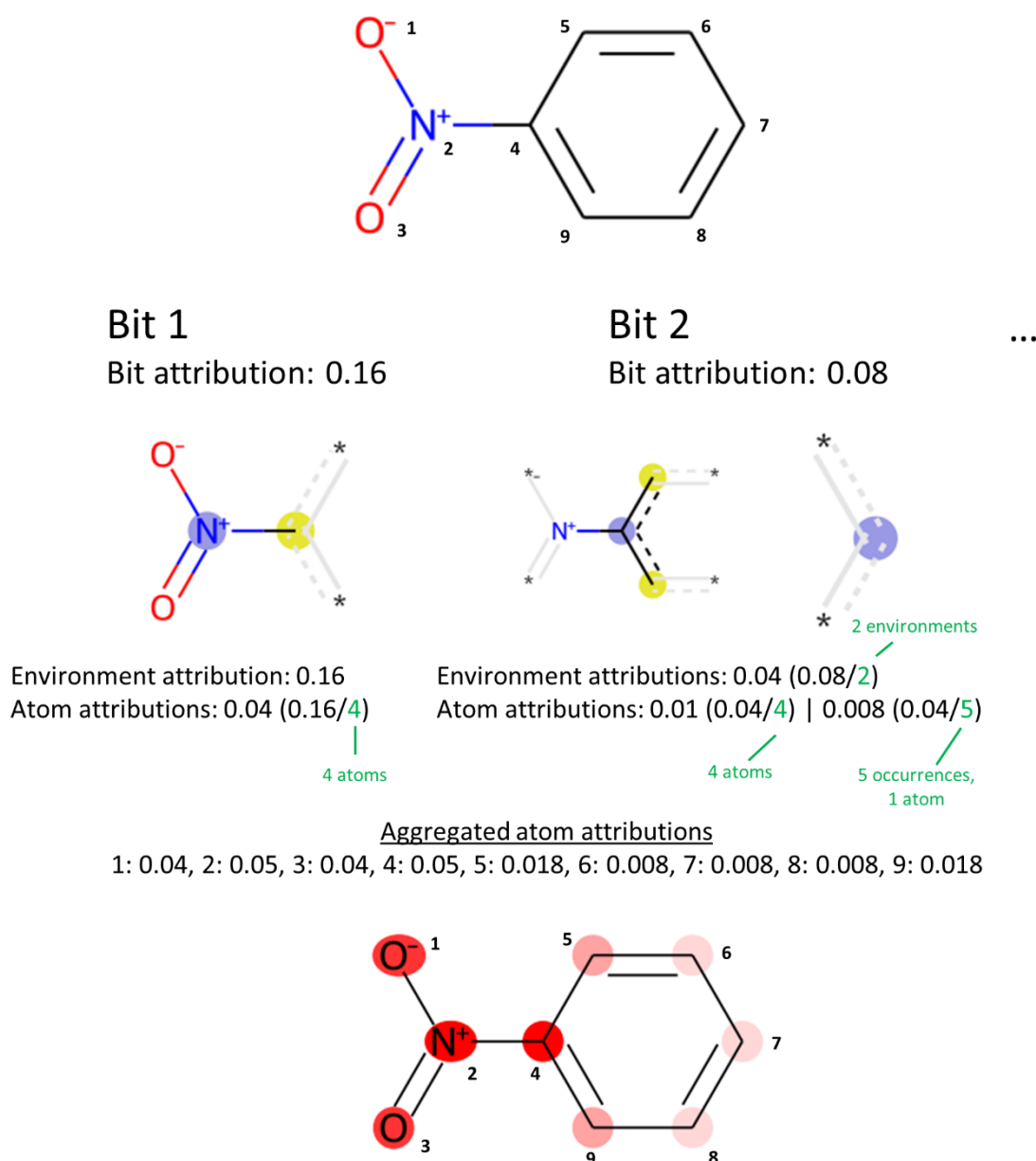


Figure 10-1 Illustration of IG_input. In this simplified illustration, only attributions for two bits (Bit 1 and Bit 2) are depicted. The mapping between bits and atom environments was selected for illustration purposes and does not correspond to RDKit's implementation of Morgan fingerprints. Bit 1 belongs to a single atom environment. Therefore, the full attribution for Bit 1 (0.16) was assigned to the environment. Then, the environment attribution was shared equally between all atoms in the environment (1, 2, 3 and 4). A rare case of bit collision occurred for Bit 2: two different atom environments map to the same bit. Therefore, the bit attribution is shared equally between both environments. The first of the two environments contains four atoms and the environment attribution is shared among the respective atoms (2, 4, 5 and 9). The second of the two environments contains just a single atom, but has five occurrences in the compound. The environment attribution is shared between those five atoms (5, 6, 7, 8, 9). To obtain the depiction, the atom attributions obtained from all bits are aggregated. Details on how the highlight colours are obtained, are described in a section 10.2.3.6 below.

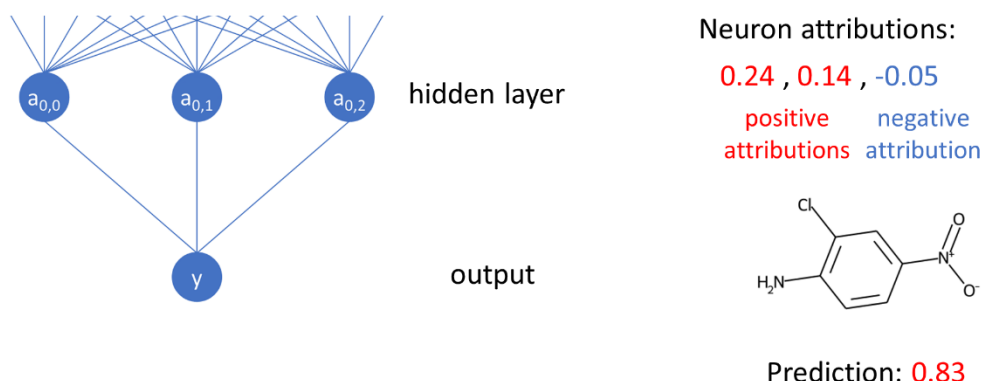
10.2.3.3 IG_hidden

In the IG_hidden method, atom attributions for the chemical substructures activating hidden neurons are obtained in a comparable manner to IG_input. As for input features, IG can be used to assign an importance (i.e. attribution) to each neuron of a hidden layer and the attributions of a compound for all neurons in a network layer sum to the prediction made by the model. The activations of neurons in the hidden layer are treated the same way as input neurons were in IG_input to find attributions. As for IG_input, the Python library Captum was used (here: *LayerIntegratedGradient* class). The attributions found for a given neuron were mapped to the chemical structure of a test compound using the extracted substructures for that neuron. As described in the previous chapter, multiple chemical substructures organised in a hierarchical network (and separated into subnetworks) were extracted for each hidden neuron. The subnetworks were utilised to find substructures that match the test compound. If a test compound does not match a given substructure in a subnetwork, none of its (more specific) child substructures will match. The attribution for a neuron (obtained from the IG method) was shared between the set of most specific substructures matching the test compound. If a test compound matched none of the substructures extracted for a neuron, the attribution for that neuron was ignored. This means that the attribution for some of the neurons may not be used to explain the prediction (i.e. will not contribute to the atom colouring). The proportion of total positive and negative attribution accounted for is reported alongside the explanation. Two different schemes were investigated to map the attributions to individual atoms. In the first, the attribution for a given fragment was shared equally by all atoms of the fragment (as is done for environments in the IG_input method). In the second case, in addition to sharing the attribution among the atoms, different weights were considered for all atoms forming a fragment with the weights derived from the weight individual fingerprint bits have for the neuron, as described in the previous chapter (section 9.2.2).

The general principle of the IG_hidden method is illustrated in Figure 10-2 for a simplified case of three hidden neurons. An attribution (positive or negative) was assigned to each neuron and for each neuron the matching substructures for the test compound were found and the neuron attribution value was shared across all atoms of the substructure. This was repeated for each neuron and atom attributions were summed to give a final model explanation. In the case shown, no weights were used for the atoms of extracted substructures.

1 Determine attribution of neurons for individual prediction

- Integrated gradients (IG) on hidden neurons



2 Map neuron attributions onto structure

- Find most specific matching substructure(s) in subnetworks
- Share attribution between atoms of substructure (here: unweighted)

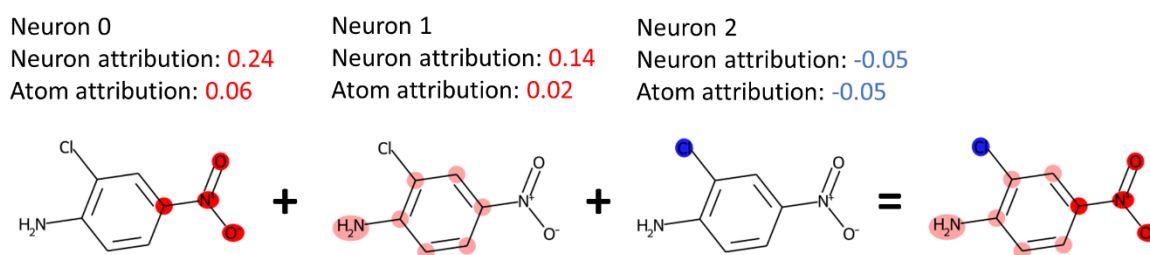


Figure 10-2 Illustration of IG_hidden. In the first step, an attribution (positive or negative) was determined for each hidden neuron for the test compound. Then, the neuron attributions were converted to atom attributions using matching substructures. In this case, one matching substructure was found for each neuron. The substructure for the first neuron was a nitro group with an aromatic carbon (4 heavy atoms). Therefore, the neuron attribution was divided by 4 and an atom attribution of 0.06 was obtained. The same procedure was applied to the other neurons' attributions. Notably, the attribution for Neuron 2 was negative, hence the blue colouring. Details for the atom colouring are provided in the section 10.2.3.6 below. In this case, no weighting was applied to the atoms of a substructure. The rightmost structure contains atom colourings aggregated from the individual neurons' atom attributions.

10.2.3.4 Evaluation of attributions

If the reason for a compound's toxicity is known, the concordance of the model's explanation with the true cause of toxicity can be evaluated. As described above, the output from Derek Nexus reports which atoms were responsible for an alert being fired. For a given compound, the ground truth is defined as the union of all atoms responsible for all alerts that are fired. Attribution ROC-AUC (McCloskey et al., 2019) was used to measure the concordance of atom attributions to the ground truth for a given positive compound. This metric is the same as the ROC-AUC score used to evaluate performance of binary classification models. That means that atoms are ranked according to their

attribution and at each threshold the TPR (true positive rate, i.e. recall) and FPR (false positive rate, i.e. 1-specificity) were recorded and then the area under the obtained ROC curve was determined. Note that the attribution ROC-AUC cannot be computed for compounds where all atoms form the ground truth of toxicity because no FPR can be computed.

Naturally, attribution AUC values can only be determined for compounds matching an alert (actual positives). These cases can be further discriminated into TPs (correctly predicted as positive) and FNs (incorrectly predicted as negative). For a FN compound, the explanation cannot be expected to match the true cause of toxicity, since the model did not predict the compound as toxic. This is a mistake of the model. However, the primary objective of this analysis is to evaluate the performance of the attributions obtained from extracted substructures. Therefore, attribution AUC scores were only computed for TP compounds. Initially, IG_hidden was evaluated using the substructure extraction as described in the previous chapter. Following this evaluation, various changes were made to the extraction process in an attempt to achieve better explanation performance. These changes are described below. In all cases, the obtained AUC scores were compared to those obtained using IG_input. For both attribution methods, the distribution of attribution AUCs obtained for the set of compounds can be summarised using the median value for a simple comparison.

10.2.3.5 Alert-specific attribution scores

A deeper understanding of the performance of attribution methods can be gained by analysing attribution AUC scores obtained for specific alerts. It may be that an attribution method performs very well for some alerts, but poorly for others. For this analysis only compounds matching a single alert were considered. Two alerts (Alert39 and Alert87) were almost always found co-occurring with the alert for alkylating agents (Alert53) and in this case they were added to the support set for Alert53 to be included in the analysis. Then, for each alert the mean attribution AUC across compounds matching this alert was computed.

10.2.3.6 Depiction of atom attributions

Atom attributions for individual compounds were depicted using a colour map. Positive attributions (contributing to a toxic prediction) were highlighted red, while negative attributions (contributing to a non-toxic prediction) were highlighted blue. Neutral atoms (attribution = 0) were not highlighted (white 'highlight'). To make the colouring between different compounds comparable, colours were

scaled according to the maximum atom attribution observed in a dataset, which may be positive or negative. The maximum atom attribution received full colour intensity and all atoms of the compounds in the dataset were assigned colours relative to this maximum. The colour intensity for individual atoms was assigned by interpolating in RGB colour space. To obtain a better discrimination of atoms in the lower range of attributions, the maximum colour intensity was assigned to all atoms with attributions at least 70% of the maximum, which has also been done in a previous study (Harren et al., 2021). Separate scales were used for IG_input and IG_hidden due to the observation that larger atom attributions were generally obtained for IG_input. The reason for this mostly seems to be that attributions for IG_hidden were ignored when no matches were found for a given neuron. As a result, colour intensities between IG_input and IG_hidden are not directly comparable.

10.2.4 Global evaluation

The global analysis is focussed on the entirety of extracted substructures and how well they match the chemical substructures associated with Derek alerts. Whether or not an extracted substructure matches an alert is evaluated by checking if the (extracted) substructure completely includes (i.e. is a superstructure of) a substructure belonging to a Derek alert (multiple substructures are possible). The set of substructures belonging to a given Derek alert was derived from the training compounds. In particular, all substructures (that occurred in the training set) belonging to a given alert were collected.

Based on this basic property of extracted substructures (being a superstructure of at least one alert structure), various analyses were conducted for each respective set of extracted substructures (several different workflows for substructure extraction were tested as described in the following section). Firstly, the proportion of alerts for which superstructures were extracted was determined as well as the number of distinct superstructures extracted for each alert. This is to determine how well the extracted substructures cover the set of Derek alerts occurring in the training set.

Also, the proportion of extracted structures that are superstructures of any alert structure was determined. This is to determine whether extracted structures are relevant with respect to toxic predictions. Notably, an extracted substructure may be relevant whilst not being associated with a Derek alert if it is related to negative predictions. To account for that, it was also determined what proportion of substructures extracted only in neurons associated with toxic predictions (confidence of toxic prediction >0.667) are 'relevant'.

Finally, the proportion of neurons associated with toxic predictions with at least one superstructure of alert structures among extracted substructures was determined. This was done to estimate

whether or not for relevant neurons (to toxicity) any meaningful substructures were extracted. Otherwise it might be helpful to analyse why no relevant substructures were extracted for affected neurons.

10.2.5 Modifications to automatic substructure extraction

The approach to automatically find chemical substructures activating hidden neurons was described in the previous chapter. In an attempt to improve the explanatory performance achieved, various modifications were made to the original approach as described below. The variations were all evaluated on the validation set using both model instances (*best_model* and *dropout_model*). Finally, one model instance and one approach was chosen for final evaluation on the test set.

Compared to the original workflow, the aim was to improve the extraction of substructures associated with Derek alerts of low frequency in the training set. The first change (Variation 1) was to allow the inclusion of FCs where the support of the extent is below *ThreshSupport* (the proportion of compounds in the extent of a FC among all compounds selected for a neuron). However, a stricter threshold for the corresponding weight was applied to these FCs. Previously, *ThreshWeightFC* (the threshold applied to the sum of the FP bit weights in the intent of a FC). To have a stricter threshold for FCs with lower frequency, *ThreshWeightLowSupp* was introduced as additional parameter. It is a multiplier on the *ThreshWeightFC* and was selected to be 1.5 meaning that FCs not meeting *ThreshSupport* need to have a sum of weights in the intent of at least $1.5 \times \text{ThreshWeightFC}$.

In Variation 2, an increased number of compounds and FP bits were considered by changing *ThreshCompound* from 3 to 2 and changing *ThreshBits* from 0.05 to 0.1. This means that training compounds with an activation of mean+2xSD and the 90th percentile of bits according to their weight were considered. This led to a much larger number of FCs for each neuron. To compensate for this, not all FCs were considered for substructure extraction. Each FC was checked whether it is novel compared to previously considered FCs for a neuron. This was done by keeping count of how often a given fingerprint bit has been included in previously considered FCs. A FC was considered novel if each of the bits in the intent had been included fewer than *ThreshNoveltyFC* (a newly introduced parameter) times in the intent of previously considered FCs. In variation 2, *ThreshNoveltyFC* was selected to be 1; *ThreshWeightLowSupp* was selected to be 1; and *ThreshSupport* to be 0. The latter means that no threshold on support was used. However, FCs were considered in decreasing order of support and the maximum number of extracted substructures per neuron was limited, meaning that FCs with low support might not be considered.

Variation 3 was introduced to limit the extraction of substructures not actually activating a neuron very strongly. This was motivated by the fact that only fingerprint bits with (high) positive weights are used to find chemical substructures. However, it may happen that an extracted substructure also contains fingerprint bits with strong negative weight which would lead to a lower true activation of the neuron than suggested by merely considering bits with positive weights. To account for this, the true neuron activation caused by a given substructure was estimated. This is difficult because a chemical substructure cannot be perfectly expressed as a Morgan Fingerprint due to the fact that some chemical environments of a compound are fully included in a substructure, whereas others are partially included (see illustration for an exemplary substructure in Figure 10-3).

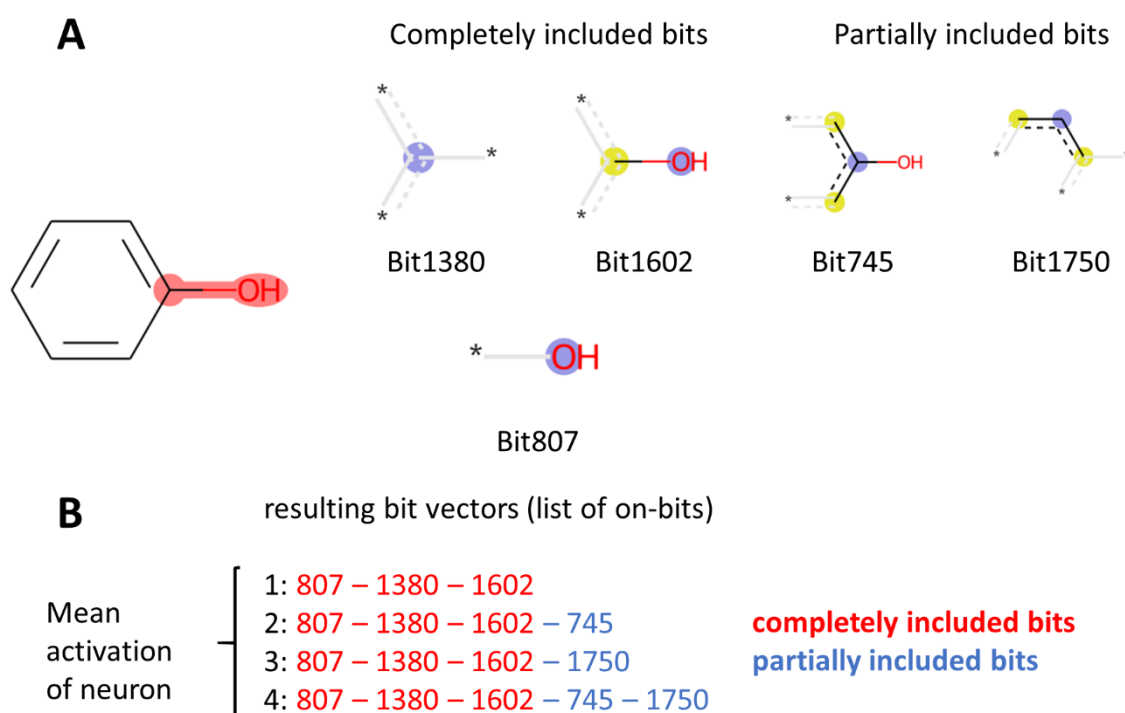


Figure 10-3 Scheme for estimating the neuron activation of a substructure. In this example the extracted substructure was a hydroxyl group attached to a single aromatic carbon atom and phenol was a compound matching this substructure in the extent of a given FC. The goal is to estimate how strongly the substructure activates the respective neuron. **A:** the atom environments that are completely and partially included in the substructure are shown. **B:** To estimate neuron activation for the substructure, various bit vectors were considered. All bit vectors contain the bits completely included in the substructure (Bit807, Bit1380, Bit1602; shown in red). The set of different bit vectors was generated by considering all potential combinations of bits partially included in the substructure (Bit745 and Bit1750; shown in blue). The estimated activation for the substructure (using phenol as associated compound) was the mean activation for all generated bit vectors. In practice, the substructure may match different compounds in the extent of a FC. In that case, the described procedure was applied to all the different compounds matching the substructure and the final estimate for the substructure's neuron activation was the mean of activations obtained for different compounds.

To account for this, a set of fingerprint bit vectors was generated for a substructure of a compound. Each of these contains the bits fully included in the substructure with distinct bit vectors generated by

considering all possible combinations of the bits partially included in the substructure. To estimate the neuron activation for a given compound-substructure pair, the mean activation of all bit vectors was calculated. In a given FC, a substructure may be supported by multiple compounds. The final estimate for the neuron activation of a substructure was the mean across all compound-substructure pairs within the FC. Only substructures whose estimated neuron activation exceeded *ThreshWeightFC* were retained. A new parameter with the name *use_all_bits_check* (whether or not the approach is applied) was introduced. Variation 3 used this approach with otherwise identical parameters as Variation 2.

In Variation 4, *ThreshNoveltyFC* was selected to be 3. Otherwise identical parameters were used as in Variation 2. The motivation for this was to check if a less restrictive threshold for novelty of FCs would lead to more relevant substructures being extracted.

An overview of all tested modifications is provided in Table 10-2. For all variations, global and local evaluation was conducted as described in the previous sections. In the local evaluation, the results are compared to atom attributions obtained when using *IG_input*.

Table 10-2 Varied parameters for automatic substructure extraction. Shown are the parameters of the original approach as described in the previous chapter and Variations 1-4. *use_all_bits_check* refers to the procedure described in Figure 10-3. *Using the value 1000 for *ThreshWeightLowSupport* effectively means that no FCs with support below *ThreshSupport* are considered. **Using the value 1000 for *ThreshNoveltyFC* effectively means that no FCs are excluded due to that criterion.

	Original	Var1	Var2	Var3	Var4
<i>ThreshCompound</i>	3	3	2	2	2
<i>ThreshBits</i>	0.05	0.05	0.1	0.1	0.1
<i>ThreshSupport</i>	0.2	0.2	0	0	0
<i>ThreshWeightFC</i>	0.5	0.5	1	1	1
<i>ThreshWeightLowSupp</i>	1000*	1.5	1	1	1
<i>ThreshNoveltyFC</i>	1000**	1000**	1	1	3
<i>use_all_bits_check</i>	No	No	No	Yes	No

10.2.6 Final evaluation on the test set

Different neural network model instances (*best_model* and *dropout_model*) as well as different approaches for substructure extraction were compared on the validation set. Among these, one model instance and one extraction approach were selected for final evaluation on the test set. This was done to investigate if the observed performances generalise beyond the validation set on a set of compounds not used during training or validation of the models. The explanations provided by the extracted substructures were compared to those obtained when using *IG_input*.

The final evaluation was conducted in the same manner as the validation. For TP compounds (not having all atoms as ground truth), attribution AUC values were obtained by comparing atom attributions to atoms being part of Derek alerts as ground truth. Furthermore, average alert AUCs were computed in the same manner as described above.

10.2.7 Analysis of proportion of attributions accounted for in IG_hidden

As described above, when applying IG_hidden it may happen that no substructure matching the test compound has been extracted for a neuron. In that case, the attribution for this neuron was not used to colour atoms of the test compound. The extent to which this was the case was investigated for the studied dataset. In particular, for each TP compound in the validation set the proportion of (a) positive attributions, (b) negative attributions and (c) total attributions (i.e. sum of absolute positive and negative attributions) that could be considered for the atom colouring (i.e. a neuron had a substructure match for the test compound) were calculated. Moreover, for exemplary compounds the most relevant neurons (i.e. the neurons with the highest attributions) were considered and the presence or absence of substructure matches was analysed.

10.2.8 Analysis of predictions for a model based on experimental Ames labels

In addition to analysing model instances trained on Derek labels as alerts, a model trained on experimental Ames labels was analysed. The same model instance as in the two preceding chapters was used. For extracting substructures activating hidden neurons, the protocol selected for final evaluation in the previous section was used and the extent to which the model attributions correspond to atoms flagged for mutagenicity by the Derek expert system was investigated. This was done for TP compounds which are also labelled as positive by Derek. In addition to comparing performances for individual compounds, average attribution AUCs for Derek alerts were determined. To enable a more robust evaluation, the validation and the test set were pooled for these analyses. Notably, the validation set was not used to optimise the explanation method in conjunction with the model trained on experimental Ames data.

Moreover, TP compounds not labelled by the Derek system were inspected. This analysis may provide insights on how Derek alerts may be refined to increase their coverage. Notably, compounds are also labelled as negative if they match an alert but the Derek software has knowledge of a negative

experimental result for a compound. In that case the experimental label in Derek is in conflict with the one found in the dataset for the present study and that information may be used to curate the data.

Finally, TN compounds were analysed in order to understand how well IG_input and IG_hidden can explain negative predictions made by a model.

10.3 Results

10.3.1 Model evaluation

The performance of both the models (*best_model* and *dropout_model*) was evaluated on the training and validation set using various metrics, as reported in Table 10-3. Both models achieved very high scores on the validation set (AUC > 0.97, accuracy > 0.9 and MCC > 0.8). For all metrics except recall, *best_model* performs slightly better than *dropout_model*. Notably, both models performed better than the one analysed in the previous chapters which was trained on experimental labels. This may be due to the fact that Derek labels are clearly defined by rules and are not prone to experimental uncertainty.

Table 10-3 Classification metrics for models. Various metrics for *best_model* and *dropout_model* are compared on both training and validation sets.

	<i>best_model</i> training	<i>dropout_model</i> training	<i>best_model</i> validation	<i>dropout_model</i> validation
ROC-AUC	0.997	0.993	0.974	0.970
Accuracy	0.978	0.960	0.914	0.903
Balanced accuracy	0.978	0.960	0.914	0.902
Precision	0.976	0.951	0.918	0.889
Recall	0.981	0.973	0.915	0.929
Specificity	0.974	0.947	0.913	0.876
F1	0.978	0.962	0.917	0.908
MCC	0.956	0.920	0.828	0.807

Furthermore, recall scores for specific alerts were determined for both models. If a model does not correctly predict compounds matching a particular alert as toxic, the model likely failed to find this alert as a reason for toxicity. The number of alerts with recall of at least 0.9 was recorded. On the training set (102 different alerts), this was the case for 88 alerts (0.862) with *best_model* and 78 alerts

(0.765) with *dropout_model*. The validation set contains 68 alerts. A recall of at least 0.9 was achieved for 49 (0.721) of them with *best_model* and 50 (0.735) with *dropout_model*, respectively. Overall, *best_model* performed slightly better than *dropout_model* with respect to most classification metrics. Nevertheless, it may be that the chemical features relevant to *dropout_model* prove to be more interpretable.

10.3.2 Local evaluation of IG_input

The capability of IG_input to explain predictions was evaluated on TP compounds in the validation set. For those compounds where not all atoms are part of the alert attribution, AUC scores are reported in Figure 10-4. The results for *best_model* are described in detail before briefly comparing these with the *dropout_model*.

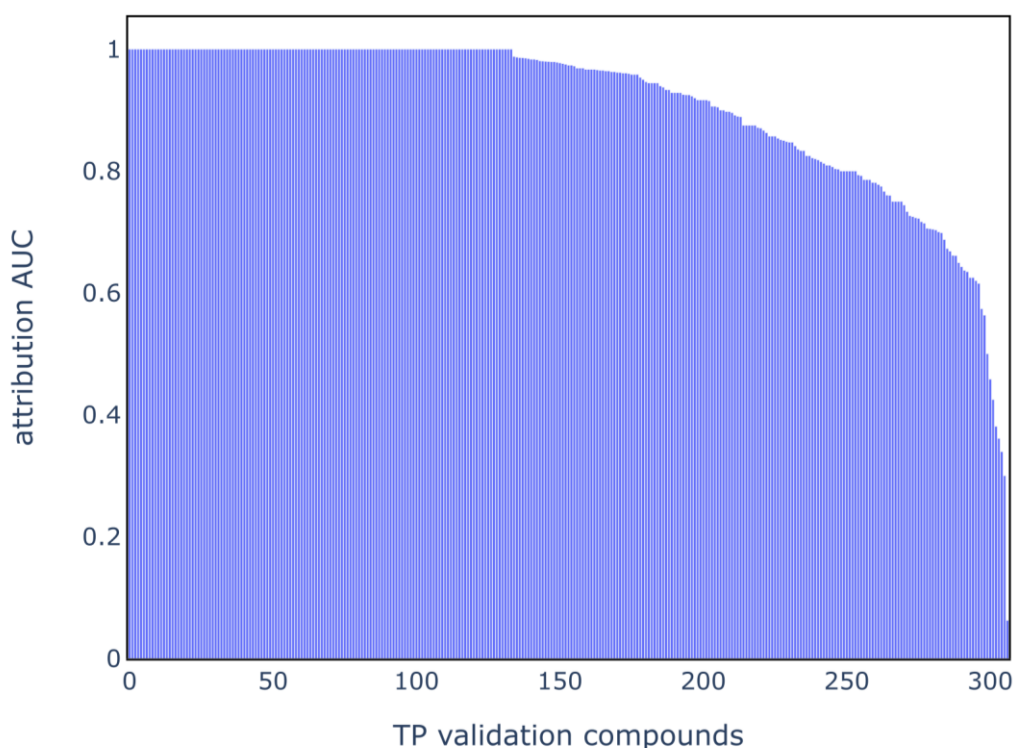


Figure 10-4 Attribution AUC scores for IG_input. Attribution AUC scores were computed for all TP compounds in the validation set (excluding compounds where all atoms are part of the ground truth).

The median score for those compounds was 0.974. For 134 (out of 307) compounds a perfect score of 1 was measured, while for 254 a score of at least 0.8 was achieved. This means that the IG_input atom attributions strongly match the ground truth for the majority of the compounds. However, for a few compounds the attributions did not agree well with the ground truth. For eight compounds, the

attribution AUC score was less than 0.5 and the minimum measured score was 0.06. Overall, IG_input worked very well to explain the predictions. For compounds with low AUC scores it is unclear whether those are because the model did not recognise the true cause of the toxic label or because the attribution method failed. Attributions for exemplary compounds along with the ground truth of the alerts and the respective attribution AUC scores (if defined) are shown in Figure 10-5.

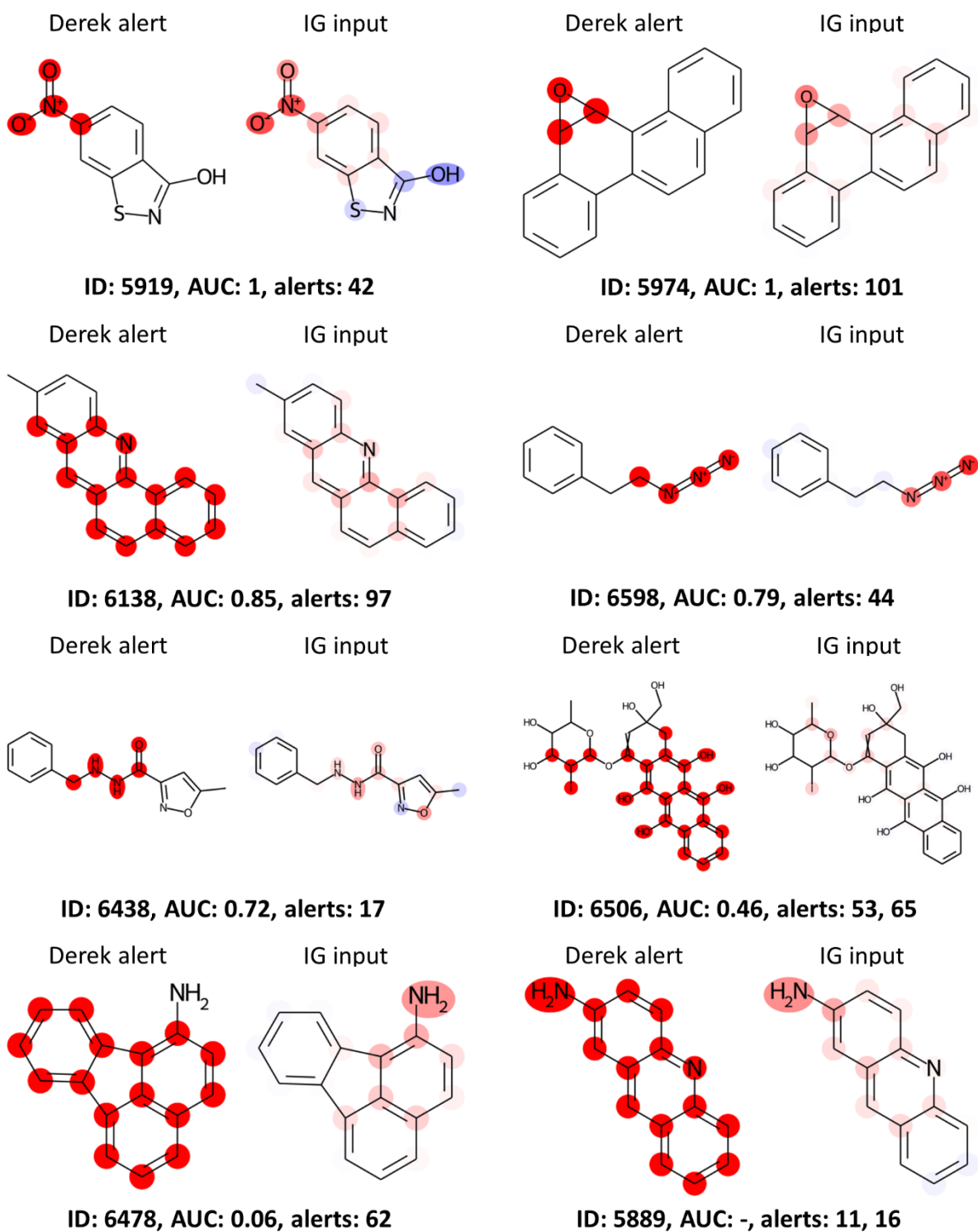


Figure 10-5 Atom attributions using IG_input. Atom attributions were obtained from attributions of input features (i.e. fingerprint bits) as described in Methodology. Atom attributions for compounds are compared to atoms matching Derek alerts and measured as attribution AUC (if possible). Alerts in this Figure: 11: aromatic amine, 16: PAH, 17: hydrazine, 42: aromatic nitro, 44: azide, 53: alkylating agent, 62: PAH, 65: PAH, 97: PAH, 101: epoxide.

Perfect AUC scores (=1) were observed for two of the depicted compounds (aromatic nitro: 5919, epoxide: 5974). A perfect AUC score corresponds to all atoms belonging to the alert receiving

attributions larger than all the remaining atoms. High AUC scores were also obtained for compounds 6138 (PAH) and 6598 (azide). In the first case, attributions largely agree with the alert, yet not perfectly. Some of the alert atoms did not receive a high attribution, whereas some atoms not belonging to the alert also received high attributions. It is worth noting that the model explanation in essence captures a polycyclic aromatic system and it seems that aromatic atoms that are part of more than one ring received high attribution values. In the azide compound, the alert was also well explained with all three nitrogen atoms having received high attributions, although the connected carbon atom did not. For compound 6438 (hydrazine) an AUC of 0.72 was achieved and in this case the explanation did not match the ground truth very well. All of the alert atoms received positive attribution values, however, atoms in the isoxazole ring (especially the oxygen atom) also received strong positive attributions.

While all considered compounds were correctly classified as toxic, in a few cases the attributions did not match the ground truth. An AUC score of 0.46 was found for compound 6506. This compound matches two alerts. The alert 'alkylating agent' is due to the iodine atom attached to the left ring. The iodine atom received a strong attribution, but the attached carbon atoms did not. Arguably, the most important atom for this alert was highlighted. Concerning the second alert (PAH type alert), some of the aromatic carbons received slightly positive attributions, but not the hydroxyl groups. This alert is not in good agreement with the respective atom attributions. The lowest AUC score of the dataset was found for compound 6478. A high attribution was assigned to the amine group which is very plausible, given that aromatic amine is an alert frequently occurring in this dataset. However, for this molecule the aromatic amine group is not part of the respective Derek alert, hence the very low AUC score. No AUC value could be computed for compound 5889 (since all of its atoms are part of the alert). This compound matched two Derek alerts (aromatic amine and PAH type alert) and most of the atoms for both alerts received positive attributions. Only two carbon atoms received slightly negative attributions. Even though not all atoms received positive attributions, the provided explanation matches both alerts fairly well.

Since an absolute colour scale was used across the whole dataset, atom attributions can be compared across different compounds. It seems that the strongest atom attributions (i.e. colour intensities) were assigned to atoms belonging to small functional groups which are well represented in the training set and for which the compounds can be predicted as toxic very accurately (e.g. aromatic nitro, epoxide, azide). When the alert structures consist of larger parts of the compound (e.g. PAH alert type), the colour intensities for individual atoms naturally will be lower.

In addition to analysing individual compounds, alert-specific attribution scores were computed, as reported in Figure 10-6. Only 50 alerts (out of 68 with at least one occurrence in the validation set) were considered. The remaining alerts either did not have any TP compounds or there were no compounds where that alert is the only alert. Of the 307 TP validation compounds, 36 were not included in this analysis as they matched multiple alerts. The reported scores are mean scores for a given alert, yet in several cases only a single compound represents an alert. For 40 out of the 50 alerts, a mean score of at least 0.8 was obtained (see Figure 10-6A). In Figure 10-6B, the performance of the alerts is depicted together with the frequencies the alerts occur in the training set. It can be seen that for all alerts with at least 80 occurrences (~1.4%) in the training set, a mean AUC of more than 0.8 was achieved. For alerts with fewer occurrences, there are some cases where lower AUCs were obtained, however, no clear trend exists between occurrences and AUC. Also for alerts with fewer than five occurrences in the training set, high AUC scores were obtained. Notably, the analysis of AUC distributions across the validation set is dominated by compounds belonging to the most frequent alerts. Of the 307 TP compounds analysed in Figure 10-4, 220 match one of the most frequent alerts (>80 occurrences in the training set).

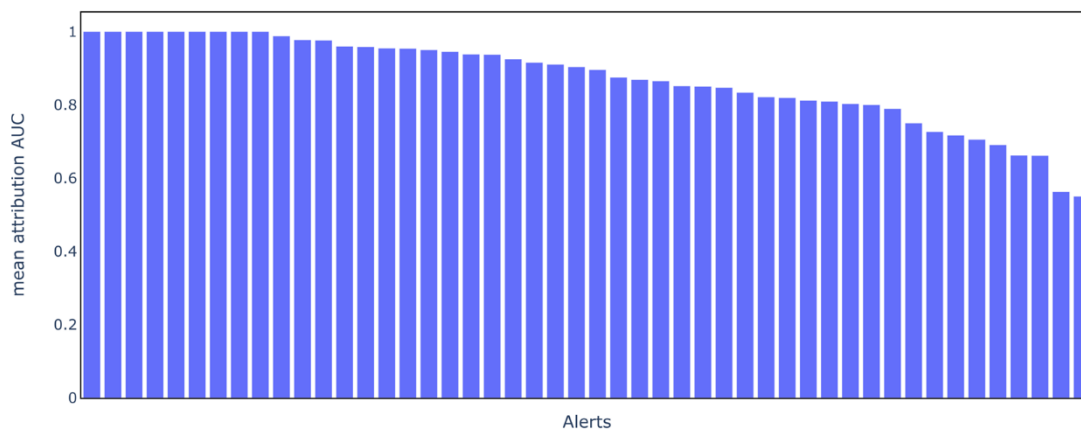
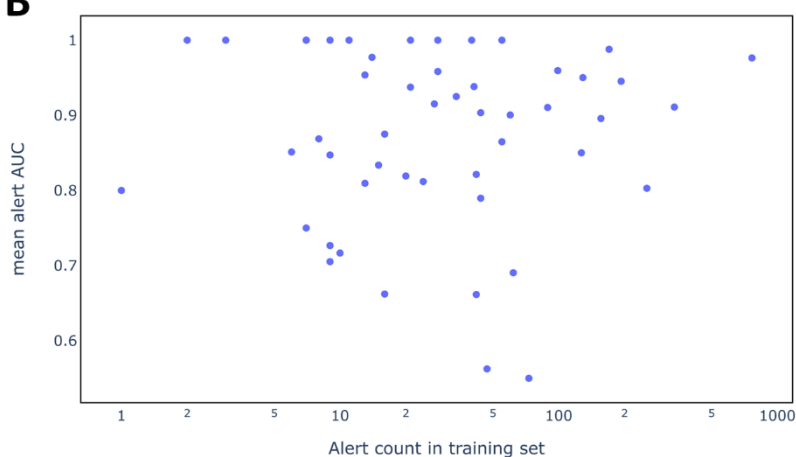
A**B**

Figure 10-6 Alert-specific AUC scores for IG_input. A: Mean attribution AUC scores for the alerts in descending order. B: Mean attribution AUC scores and occurrence of alerts in training set (logarithmic scale).

The results presented here demonstrate a good explanatory performance using IG_input for *best_model*. Table 10-4 shows the corresponding performance metrics for *dropout_model*. Notably, a direct comparison is of limited meaning, as the set of TP compounds and also the set of alerts covered in TP compounds differs between the two models. Nonetheless, comparable explanatory performance was observed for the two models. In section 10.3.4 below, the performance achieved using IG_hidden will be compared to those shown here.

Table 10-4 Explanatory performance of *best_model* and *dropout_model*.

	<i>best_model</i>	<i>dropout_model</i>
Median AUC	0.974	0.964
AUC ≥ 0.8	254/307 (0.827)	254/311 (0.817)
Median alert AUC	0.900	0.894
Alert AUC ≥ 0.8	38/48 (0.792)	36/52 (0.692)

10.3.3 Global evaluation of extracted substructures

For each extraction method and model, various numbers are reported in Table 10-5 to evaluate the quality of extracted substructures globally. The objective was to estimate how well the extracted substructures correspond to known Derek alert structures.

Table 10-5 Global evaluation for various substructure extraction workflows. Reported are various figures characterising the total set of extracted substructures for both *best_model* and *dropout_model*. These are the number of distinct substructures extracted for all neurons, the number of Derek alerts for which superstructures were extracted, the number of neurons that are associated with toxic predictions for which superstructures of Derek alerts were extracted, the proportion of the extracted substructures that represents superstructures of Derek alert structures ('relevant'), and the proportion of relevant substructures extracted from neurons associated with toxic predictions. *Neurons associated with toxic predictions are neurons for which the confidence of a positive prediction is at least 0.667 upon activation. The threshold used for all neurons here was selected to be the 20th percentile of maximal neuron activations across the dataset. For more details on confidence of neurons, see chapter 8.

		Distinct substructures	Extracted alerts	Relevant neurons among toxic neurons*	Relevant substructures	Relevant substructures among toxic neurons*
<i>best_model</i>	Original	3213	49/102	169/205	0.286	0.312
	Variation1	29676	88/102	204/205	0.396	0.491
	Variation 2	33387	96/102	204/205	0.47	0.577
	Variation 3	33427	97/102	204/205	0.483	0.596
	Variation 4	35947	95/102	204/205	0.464	0.584
<i>dropout_model</i>	Original	3511	50/102	180/226	0.393	0.516
	Variation 1	37911	93/102	225/226	0.471	0.63
	Variation 2	39164	101/102	225/226	0.577	0.753
	Variation 3	39164	101/102	225/226	0.577	0.753
	Variation 4	49233	99/102	225/226	0.546	0.727

Similar trends were observed for *best_model* and *dropout_model*. In both cases, the original extraction method yielded more than 3000 distinct substructures and the different variations tested resulted in much higher numbers of substructures. Notably, the larger number of distinct substructures corresponded to superstructures for a wider range of the Derek alerts. While the original workflow retrieved superstructures for only about half of all alerts in the training set, the

Variations 2, 3 and 4 resulted in superstructures for almost all of the alerts. Figure 10-7 shows the proportion of alerts for which superstructures were extracted using the original workflow for different frequencies of occurrence of the alerts in the training set. It can be seen that this method failed to extract superstructures for many of the less frequent alerts.

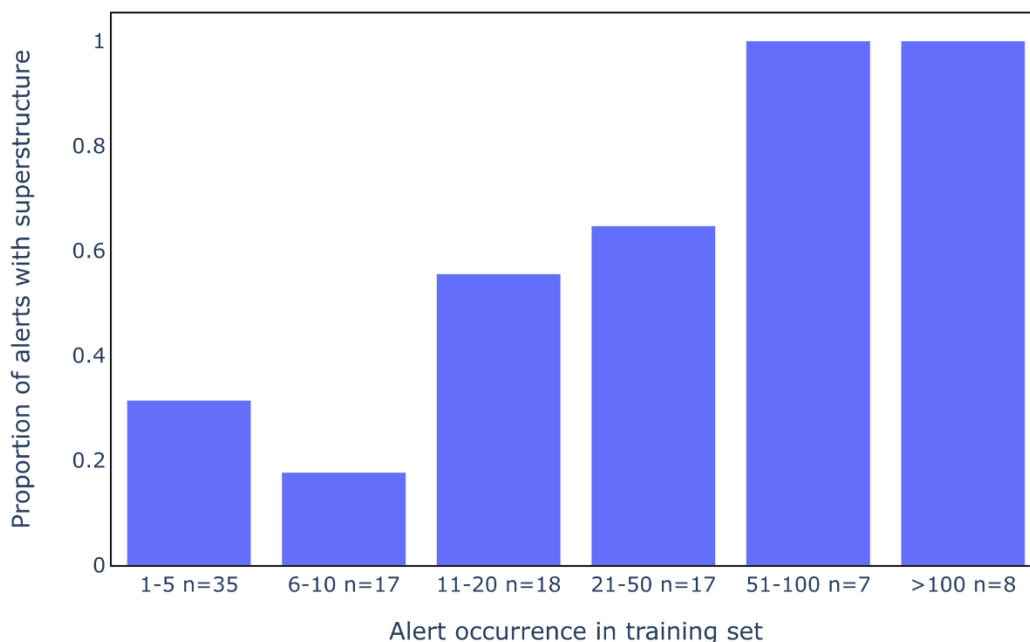


Figure 10-7 Coverage of Derek alerts for original extraction workflow. Alerts were assigned to bins according to frequency of occurrence in the training set. The figure reports the proportion of alerts for which at least one superstructure was extracted in any of the neurons. The plot reports the figures for *dropout_model*.

Also investigated was the number of neurons associated with toxic predictions for which superstructures of Derek alerts were among the extracted fragments. For the original workflow this was the case for the majority of neurons for both *best_model* (169/205) and *dropout_model* (180/226). The different variations of the extraction method increased these numbers further so that at least one superstructure was extracted for nearly all of those neurons.

Finally, the proportion of the retrieved substructures that correspond to any Derek alert and hence can be considered 'relevant' was investigated. This was analysed for all substructures (extracted in any neuron) and also for those extracted from neurons linked to toxic predictions. As expected, a higher proportion was always found among substructures extracted from toxic neurons. Interestingly, the variations of the extraction process led to higher proportions of relevant structures, although also much higher numbers of substructures were retrieved for those. Moreover, consistently higher proportions were found for *dropout_model* compared to *best_model*. The retrieval of substructures belonging to more of the alerts as well as the larger proportion of relevant substructures being

extracted suggest that *dropout_model* may be better suited to extract meaningful substructures from its hidden neurons.

In the following, exemplary fragments are shown for two different alerts. The substructures were taken from the Variation 2 approach applied to *dropout_model*. Firstly, substructures related to aromatic nitro compounds, the most frequent alert in the dataset, are shown in Figure 10-8 (labelled 42-1 – 42-9). In total, 4999 distinct substructures related to the alert were retrieved (33387 substructures in total). Substructures related to this alert were retrieved in 213 different neurons and in 179 neurons related to toxic predictions indicating a vast distribution across neurons in the neural network. It is worth noting that many of the structures are superstructures of different alerts at the same time.

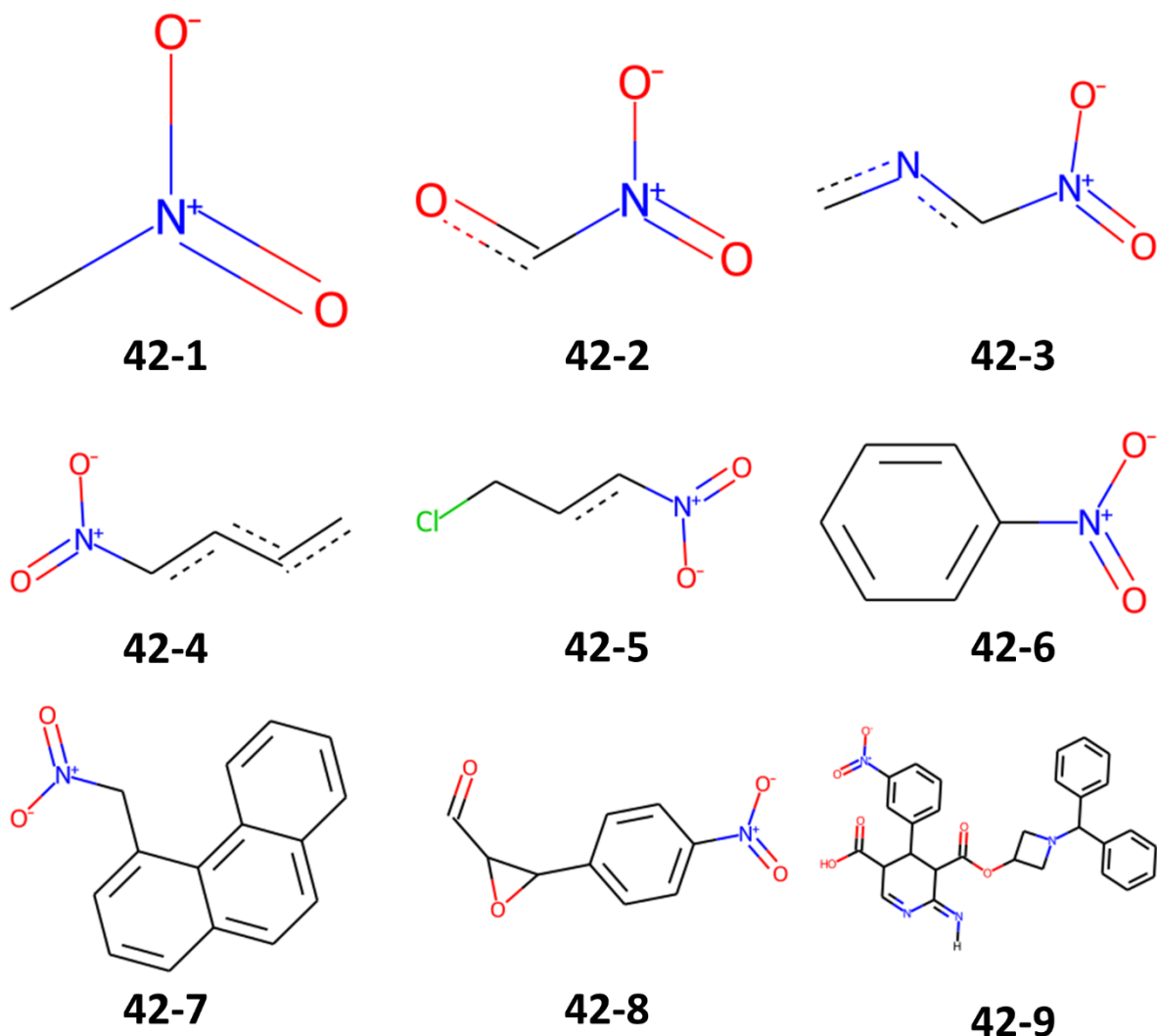


Figure 10-8 Exemplary substructures related to aromatic nitro (Alert42). The substructures were extracted in various neurons and selected to represent the variety of superstructures found for this alert.

The first substructure (42-1) matches all atoms marked by the Derek alert: a nitro group attached to a single aromatic carbon atom. Substructures 42-2 and 42-3 match an oxygen or nitrogen atom within the aromatic ring, respectively, and hence are more specific than 42-1. 42-4 includes 4 aromatic carbons and 42-6 a full phenyl ring. The remaining substructures include different chemical features and alerts. 42-5 includes a chlorine attached to an aliphatic carbon. 42-7 is also a superstructure of the polycyclic aromatic hydrocarbon alert. 42-8 includes even two further alerts: an epoxide and an aldehyde. 42-9 is an example of a very large and specific fragment. Such fragments were extracted rarely.

Secondly, some selected superstructures for the less frequent alert 'nitrogen or sulphur mustard' (Alert39) are shown in Figure 10-9. In total, 327 substructures belonging to this alert were extracted in 166 different neurons and in 120 neurons linked to toxic predictions. This shows that less frequent alerts also seem to be detected in a large number of neurons.

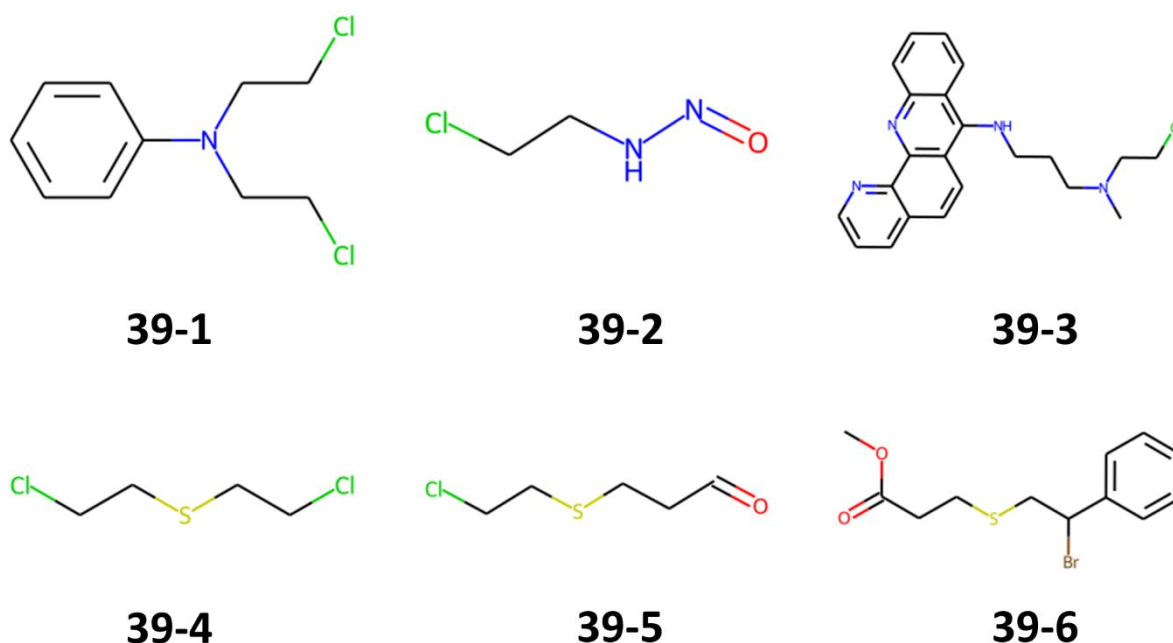


Figure 10-9 Exemplary substructures related to nitrogen or sulphur mustard (Alert39). The substructures were extracted in various neurons and selected to represent the variety of superstructures found for this alert.

This alert structure consists of a nitrogen or sulphur atom which is connected to one or two halogen atoms via an ethyl group. The nitrogen or sulphur atom can activate the carbon next to the halogen in an intramolecular mechanism to make it a strong electrophilic. The first row of structures shows nitrogen mustard compounds. As with aromatic nitro compounds, other alerts may be included such as nitrosamine in 39-2 or a PAH and aromatic amine in 39-3. The second row shows sulphur mustard

compounds. Structure 39-6 is an example of a bromine as halogen atom. This shows that the extracted substructures related to Alert39 cover different chemical variations of this alert.

10.3.4 Local evaluation of IG_hidden

10.3.4.1 Analysis of overall and alert-specific performances

The explanatory performance of IG_hidden was evaluated both for *best_model* and *dropout_model* by mapping extracted substructures onto test compounds and compared to the IG_input attribution method. A direct comparison between the two models (*best_model* and *dropout_model*) is not straightforward, as different compounds may have been predicted correctly and hence the set of TP compounds (and alerts) is not identical. A summary of the various evaluation scores is presented in Table 10-6. Each modelling method using IG_hidden is evaluated with and without weighting applied to the atoms of the substructures (see Methodology).

Table 10-6 Local evaluation metrics for different model instances and substructure extraction workflows. Reported are the median AUC (for TP compounds), the proportion of compounds with an AUC ≥ 0.8 , the median alert AUC and the proportion of alerts with an alert AUC ≥ 0.8 . The best attribution method for each model instance is in bold.

		Median AUC	AUC ≥ 0.8	Median alert AUC	Alert AUC ≥ 0.8
<i>best_model</i>	IG_input	0.974	0.827	0.900	0.792
	Original	0.867	0.576	0.683	0.396
	Original we	0.875	0.576	0.683	0.417
	Var1	0.917	0.645	0.828	0.563
	Var1 we	0.938	0.645	0.850	0.542
	Var2	0.867	0.619	0.868	0.583
	Var2 we	0.903	0.642	0.866	0.583
	Var3	0.867	0.606	0.787	0.5
	Var3 we	0.900	0.635	0.815	0.521
	Var4	0.833	0.547	0.788	0.479
	Var4 we	0.889	0.611	0.781	0.5
<i>dropout_model</i>	IG_input	0.964	0.817	0.894	0.692
	Original	0.933	0.637	0.729	0.404
	Original we	0.917	0.627	0.686	0.404
	Var1	0.967	0.717	0.866	0.615
	Var1 we	0.958	0.711	0.873	0.615
	Var2	0.917	0.669	0.881	0.692
	Var2 we	0.935	0.727	0.903	0.712
	Var3	0.917	0.669	0.881	0.692
	Var3 we	0.935	0.727	0.903	0.712
	Var4	0.893	0.678	0.864	0.654
	Var4 we	0.939	0.736	0.895	0.673

For *best_model*, IG_input achieved the best scores when evaluating both individual compounds and average values for different alerts. The IG_hidden attribution method closest to it when evaluating single compounds was Variation 1 with weights which achieved a median AUC of 0.938 (vs. 0.974 for IG_input) and a proportion of 0.645 (vs. 0.827) compounds with an AUC of at least 0.8. Variation 2 showed the highest median alert AUC (0.868 vs. 0.900 for IG_input) and Variation 2 (both with and without weights) showed the highest proportion of median AUCs above 0.8 (0.583 vs. 0.792 for IG_input).

For *dropout_model*, IG_input was the best attribution approach overall, however, the scores for the IG_hidden methods were very similar or even slightly higher in some cases. Variation 1 achieved a median AUC score of 0.967 (vs. 0.964 for IG_input). Variation 4 with weights achieved the highest proportion of compounds with an AUC of at least 0.8 (0.736 vs. 0.817 for IG_input). Variation 2 and Variation 3 (each with weights) achieved a slightly higher median alert AUC higher than IG_input

(0.903 vs 0.894). These IG_hidden methods also achieved a larger proportion of alerts with an AUC of at least 0.8 compared to IG_input (0.712 vs 0.692). Clearly, the explanations obtained with IG_hidden are more competitive with IG_input for *dropout_model* as opposed to *best_model*. From this it can be concluded that a network using dropout may simplify the extraction of meaningful molecular fragments to interpret the model. Conceptually, models using dropout should lead to single neurons representing more meaningful features in the model.

Another observation, made for both *best_model* and *dropout_model*, is that the original workflow performed relatively well in terms of median AUC across all compounds, but the various modifications to the workflow led to higher average alert AUC scores. This suggests that good explanations are provided for a wider range of different substructure types.

The AUC scores for individual compounds, obtained from various attributions methods for *dropout_model*, are depicted in Figure 10-10. It can be seen that IG_input clearly achieved the highest scores overall, although the highest median score was found for IG_hidden Variation 1. Among the different IG_hidden methods, Variation 2 with weights and Variation 4 with weights achieved the best scores. A more detailed comparison of these two approaches is shown in Figure 10-11A with a scatter plot comparing AUCs for individual compounds and Figure 10-11B with a scatter plot comparing mean AUC values for the different Derek alerts.

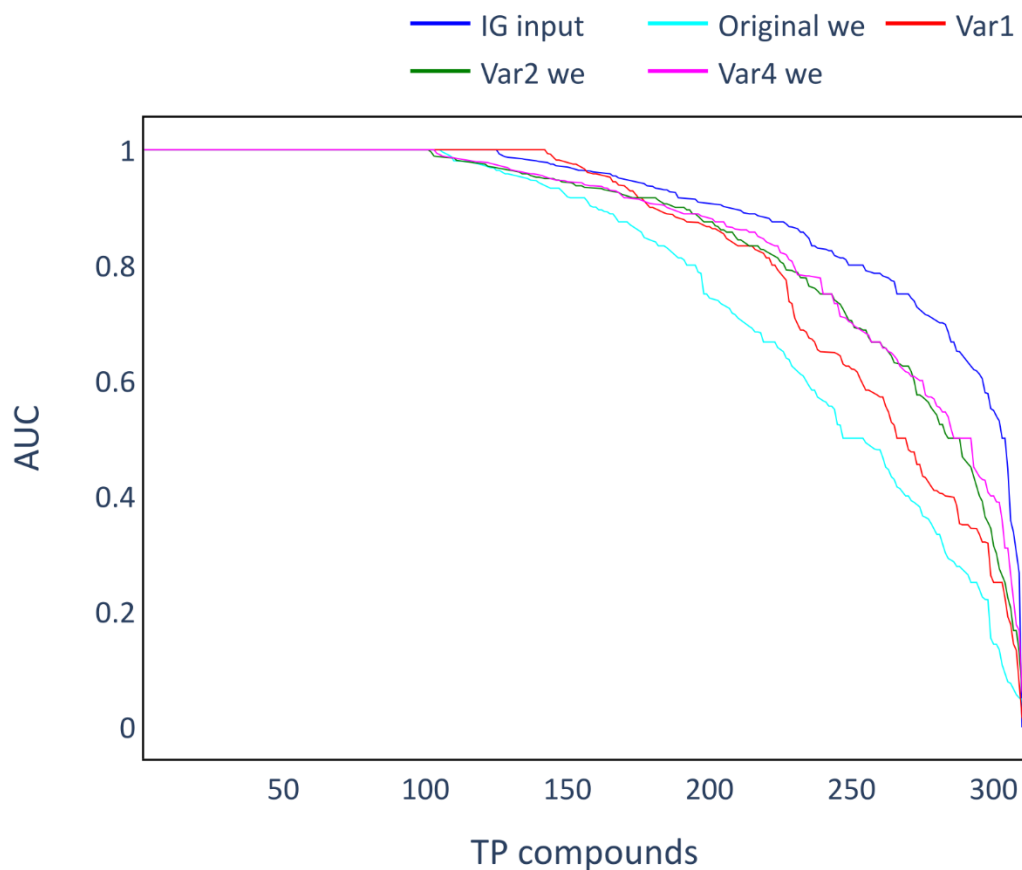


Figure 10-10 Comparison of attribution AUC scores for different attribution methods. AUC scores for individual compounds are reported for selected IG_hidden variations and IG_input. For each attribution method, AUCs for individual compounds were sorted in descending order and plotted as lines. The data is for *dropout_model*. The included IG_hidden variations were: original workflow with weights (Original we), Variation 1 (Var1), Variation 2 with weights (Var2 we) and Variation 4 with weights (Var4 we).

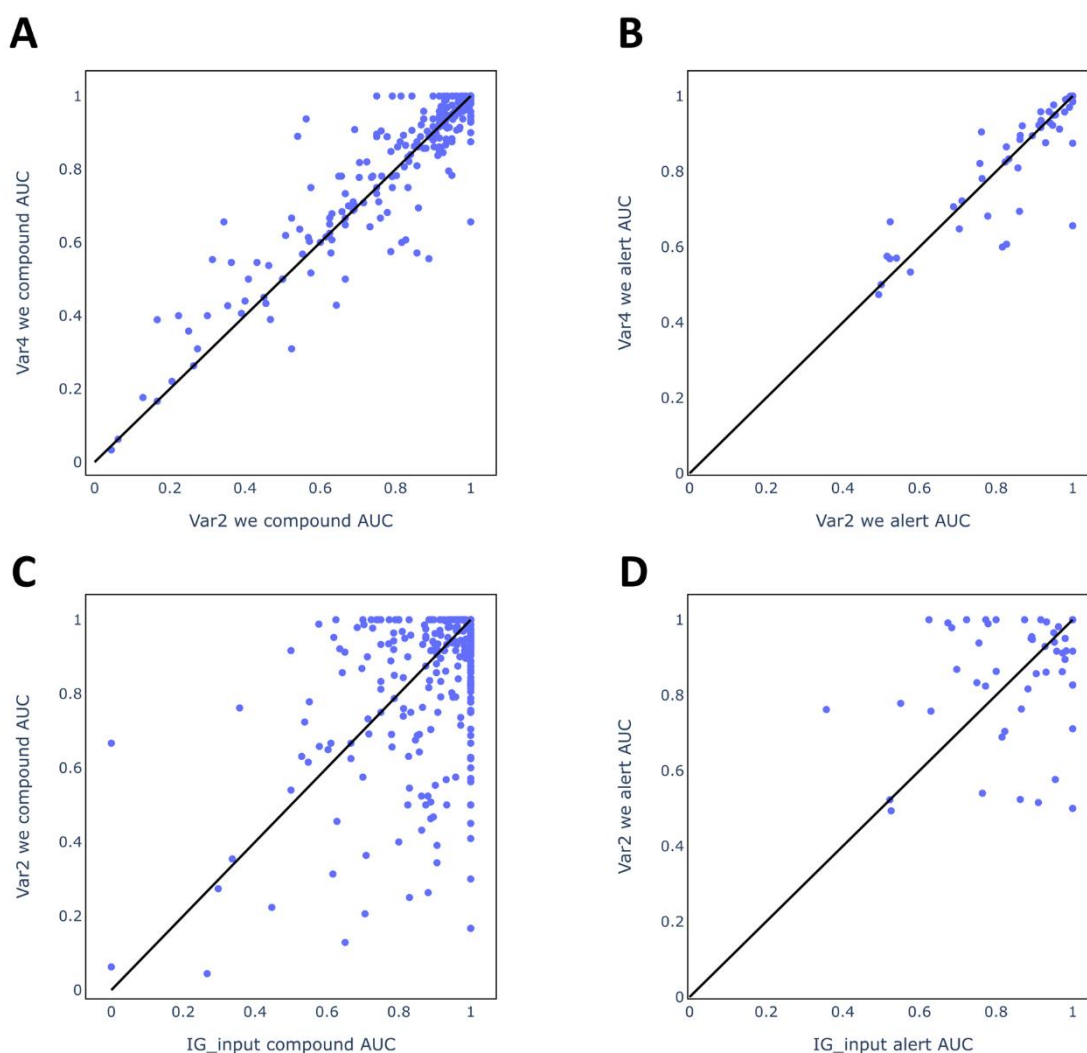


Figure 10-11 Comparisons of compound AUCs and alert AUCs. Comparisons were made between Variation 2 with weights (Var2 we) and Variation 4 with weights (Var4 we) as well as between Variation 2 with weights and IG_input. **A:** Scatter plot for individual compound AUCs (Var2 we vs. Var4 we), **B:** Scatter plot for Derek alert AUCs (Var2 we vs. Var4 we), **C:** Scatter plot for individual compound AUC2 (Var2 we vs. IG_input), **D:** Scatter plot for alert AUCs (Var2 we vs. IG_input)

When analysing the scatter plot for individual compounds (10-11A), the scores were very similar for the two methods with a large number of compounds in the upper right corner (AUC close to 1 for both methods). At lower AUC scores there were several compounds with relatively large differences in scores, however, these were a minority. In contrast, Figure 10-11B shows that IG_hidden Variation 2 (with weights) outperformed Variation 4 (with weights) when analysing mean AUC scores for Derek alerts. Therefore, Variation 2 (with weights) was selected for final evaluation on the test set.

In Figure 10-11C and 10-11D, Variation 2 (with weights) is compared to IG_input. While IG_input achieved higher scores overall, there were several compounds and alerts for which IG_hidden led to better explanatory performances. Table 10-7 provides an overview of mean AUC scores for some

selected alerts (frequent alerts and alerts with large differences in performance. A direct comparison of explanations for several individual compounds is provided in the following section.

Table 10-7 Average alert AUCs. The table reports the average AUC scores for the 10 most frequent alerts in the training set as well as a selection of alerts where one of the attribution methods clearly outperforms the other.

Alert ID	Alert Name	Proportion train set	IG input	IG_hidden (Variation 2 (with weights))
42	Aromatic nitro	0.130	0.983	0.918
53	Alkylating agent	0.058	0.98	0.895
97	PAH	0.043	0.764	0.54
101	Epoxide	0.033	0.974	0.912
31	N-Nitro or N-nitroso	0.029	0.98	0.95
21	Aromatic amine or amide	0.027	0.891	0.95
89	Aromatic azo	0.022	0.952	0.941
28	Quinoline	0.022	0.8	0.863
37	Aromatic hydroxylamine/ester	0.017	0.973	0.862
11	Aromatic amine or amide	0.015	0.895	0.948
80	Isocyanate or isothiocyanate	0.002	1	0.5
34	Aromatic nitroso compound	0.007	1	0.711
61	Hydroperoxide	0.002	1	0.5
65	PAH type alert	0.007	0.63	0.758
40	Quinolone derivatives	0.003	0.674	0.992
92	Halogenated alkene	0.011	0.357	0.762

When considering the most frequent alerts, in almost all cases both attribution methods achieved high average AUC scores (>0.8). An exception was alert 339 with a score of 0.764 for IG_input and 0.54 for IG_hidden (Variation 2 with weights). This is further discussed below when analysing atom attributions for individual compounds. Overall, these findings confirm that both methods provide accurate explanations for the most frequent alerts.

Inaccurate explanations for at least one of the attribution methods occurred mostly in rarer alerts. However, as can also be seen in Figure 10-11D, IG_hidden performed better than IG_input for some alerts, while the opposite is the case for other alerts. Some of these alerts are listed in the table and examples of individual compounds are provided below.

10.3.4.2 Analysis of individual compounds

In this section, model explanations (i.e. atom colourings) obtained from IG_input and IG_hidden are compared. As mentioned above, the colour intensity between the two methods is not directly comparable as different scales had to be used. It is of note that when using IG_hidden, some of the hidden neurons may not lead to substructure matches which means that the attributions for these neurons cannot be used to explain the prediction. Table 10-8 reports the proportion of total attribution that is accounted for by substructure matches as well as separate values for neurons with positive and negative attribution, respectively, for the compounds in Figures 10-12 to 10-15. These values are helpful to contextualise the explanations presented for IG_hidden in this section and a more detailed analysis of this issue is presented in a later section (10.3.6).

Table 10-8 Overview of atom attributions accounted for by substructure matches for the IG_hidden method. If no substructure match is found for the fragments extracted from a neuron and the test compound, the neuron attribution is not used to explain the model prediction. Shown are the proportions of summed neuron attributions for which substructure matches exist for total (positive and negative), positive and negative attributions.

ID	Proportion total neuron attributions accounted for	Proportion positive neuron attributions accounted for	Proportion negative neuron attributions accounted for
5919	0.253	0.271	0.124
6005	0.36	0.447	0.016
6561	0.157	0.178	0.074
6083	0.058	0.06	0.007
6155	0.011	0.013	0
6520	0.049	0.055	0.024
6033	0.063	0.078	0.032
6097	0.18	0.189	0.162
6153	0.353	0.419	0.077
6207	0.141	0.159	0.042
6116	0.078	0.107	0.009
6219	0.067	0.069	0.051
6476	0.082	0.097	0.007
6533	0.285	0.223	0.403

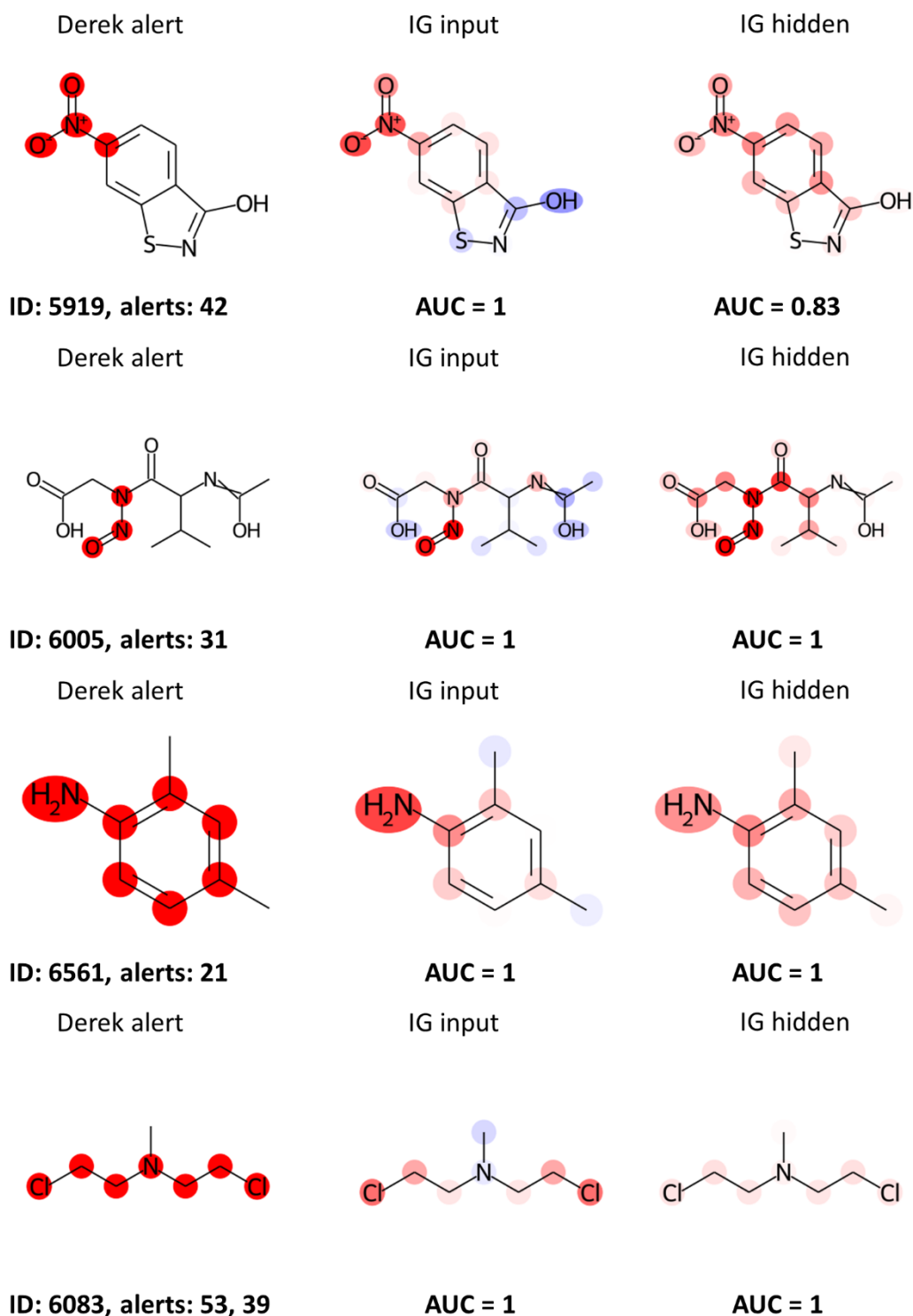


Figure 10-12 Comparison of atom attributions for individual compounds (Part 1). Shown are compounds with Derek alerts highlighted (first column), atom attributions along with the corresponding attribution AUC for IG_input (second column), and IG_hidden (Variation 2 with weights) for *dropout_model*. Alert42: aromatic nitro, Alert31: N-nitro or N-nitroso, Alert21: aromatic amine, Alert53: alkylating agent, Alert39: nitrogen or sulphur mustard.

The first set of compounds (Fig 10-12) provides examples where both IG_input and IG_hidden led to high attribution AUC scores. For compound 5919, IG_hidden highlighted the full aromatic ring which

resulted in a lower AUC of 0.83, as in the Derek alert only the aromatic carbon next to the nitro group belongs to the ground truth. The AUC score depends on the definition of the alert and in this case highlighting the complete phenyl ring may still be considered a correct explanation. In contrast, the alert 352 (aromatic amine) is defined to comprise the complete phenyl ring (compound 6561). Both methods here led to a perfect AUC score and again IG_hidden highlighted the full ring, whereas IG_input mostly highlighted carbon atoms in close range to the amine group. For compound 6005, the nitrosamine group correctly received the highest attributions. Again, atoms next to the nitrosamine received positive attributions. Both methods obtained a perfect AUC score for compound 6083, yet only IG_hidden highlighted all atoms belonging to the nitrogen mustard motif (albeit some very weakly). IG_input assigned a negative attribution to the nitrogen atom which is essential to the electrophilic properties of the compound. A perfect AUC score was only achieved because the remaining carbon atom received a stronger negative attribution. In all these examples, IG_hidden assigned positive attributions to larger parts of the compounds compared to IG_input. This can be explained by the fact that the explanations given were derived from substructures potentially much larger than the atom environments of radius 1 used for IG_input. Similar observations can be made for the compounds depicted in the following figures.

The next set of compounds (Figure 10-13) includes PAHs (Alert97) and related structures (Alert20 and Alert14) Alert97 is one of the most frequent alerts and one where IG_hidden achieved quite poor AUC scores (see Table 10-7). As reported in Table 10-8, the proportion positive attribution accounted for is low for all compounds in this set resulting in faint atom colourings for IG_hidden. While not visible, IG_hidden assigned (weak) positive attributions to all atoms in compound 6155, yet not all the rings are part of the ground truth according to rules defined in the Derek software. In other cases, IG_hidden assigned positive attributions to groups attached to the aromatic system, as, for instance, the sulphate group in compound 6520 (again very pale colouring). The attributions from IG_input were focussed on carbon atoms belonging to multiple rings. This is plausible as these atoms are part of atom environments indicative of polycyclic aromatic systems. Overall, IG_input achieved higher AUC scores for Alert97. Quite different attributions were obtained for compound 6033. While IG_input assigned positive attributions to all atoms, the focus is on the pyrrole-like ring within the polycyclic system. This led to a relatively high AUC score (0.863), yet it clearly did not provide the entire relevant polycyclic moiety as explanation. IG_hidden highlighted atoms only weakly and did not highlight the alert structure resulting in a poor AUC score (0.524). Similar as before, IG_input only highlighted parts of the complete ground truth fragment for compound 6097. Here the attributions provided by IG_hidden were mostly concordant with the alert structure.

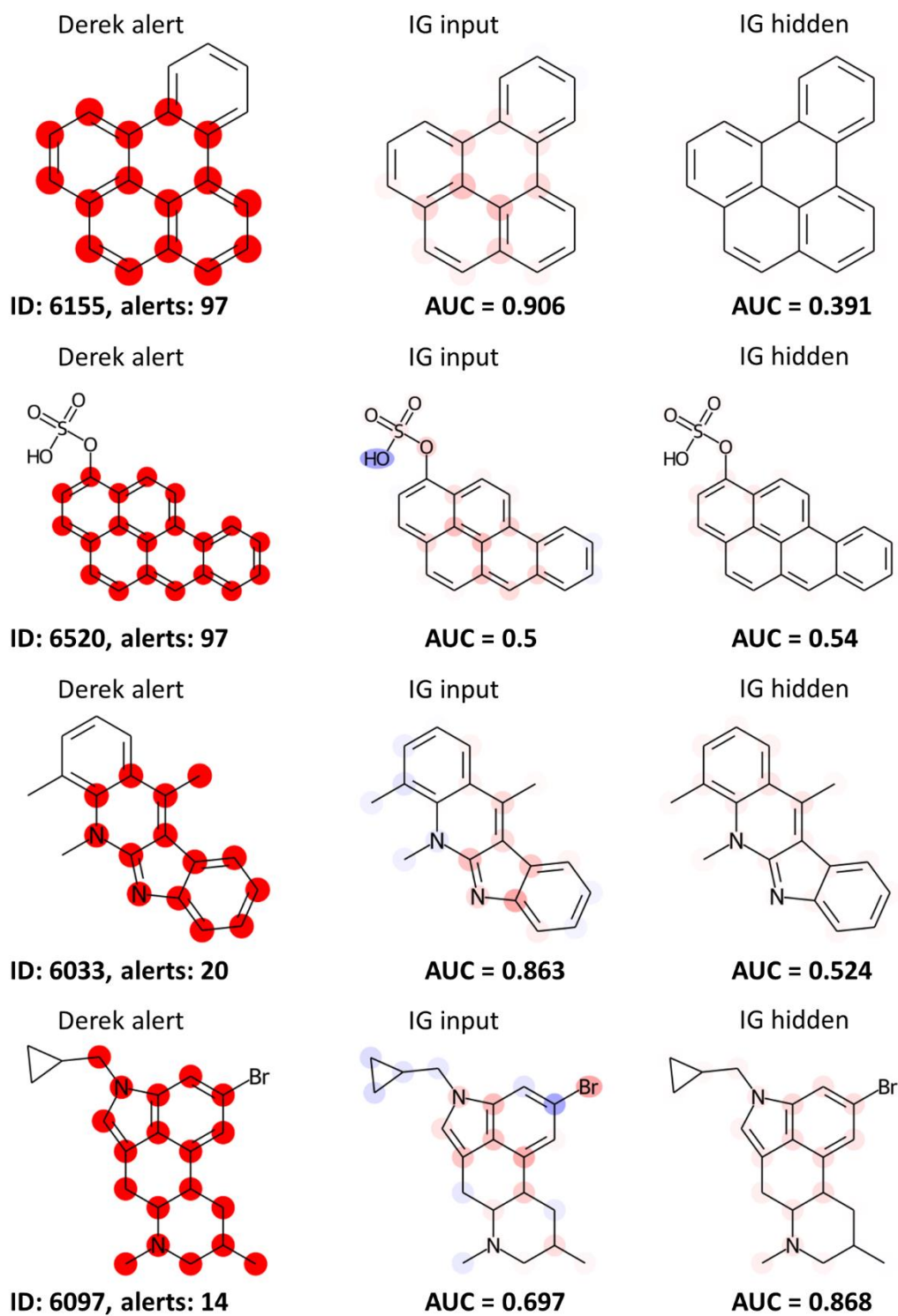


Figure 10-13 Comparison of atom attributions for individual compounds (Part 2). Shown are compounds with Derek alerts highlighted (first column), atom attributions along with the corresponding attribution AUC for IG_input (second column), and IG_hidden (Variation 2 with weights) for *dropout_model*. Alert 97: PAH, Alert20 and Alert 14: PAH type alerts.

In the next set (Figure 10-14), three examples where IG_hidden outperformed IG_input are provided. A good agreement between the attributions of IG_hidden and the ground truth exists for compound

6153 (quinolone derivative). As observed for other compounds, atoms not belonging to the ground truth also received (here weak) positive attributions. In contrast, the explanation provided by IG_input did not agree well with the ground truth. A perfect score was achieved by IG_hidden for compound 6207. However, clearly the strongest attributions were given to the ring with the two amine groups. This may be since the aromatic amine group is also recognised by the network as an alert for mutagenicity. Also, IG_input mostly highlighted atoms of the PAH type system. However, the hydroxyl groups at the central ring received negative attributions which means that the attributions did not match the entire system. The thiophosphate alkyl ester was perfectly explained by IG_hidden for compound 6116. IG_input highlighted only parts of this group. When IG_input was outperformed by IG_hidden, this was mostly due to the fact that IG_input failed to recognise all components of large alert structures.

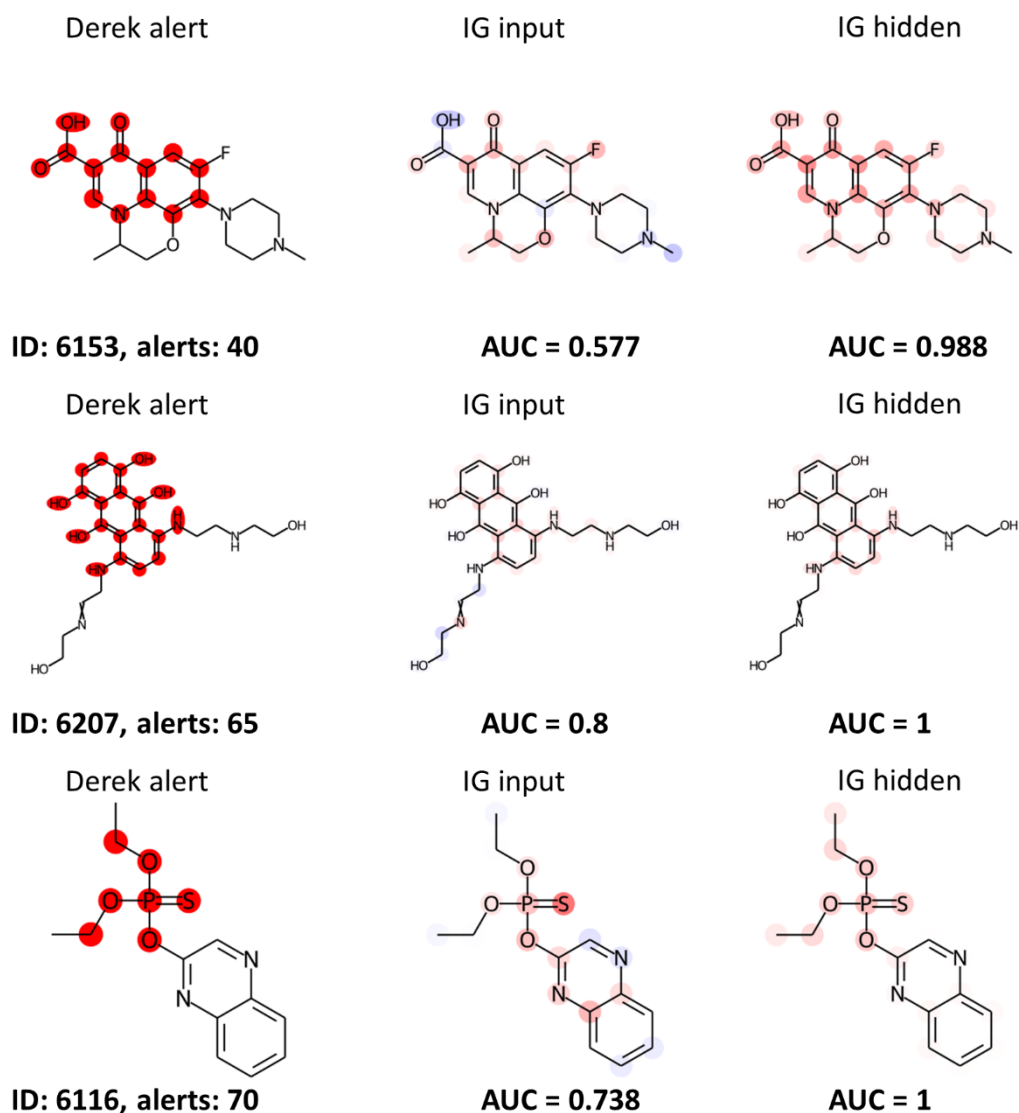


Figure 10-14 Comparison of atom attributions for individual compounds (Part 3). Shown are compounds with Derek alerts highlighted (first column), atom attributions along with the corresponding attribution AUC for IG_input (second column), and IG_hidden (Variation 2 with weights) for *dropout_model*. Alert40: quinolone derivative, Alert65: PAH type alert, Alert70: thiophosphate alkyl ester.

The last set of compounds considered in this section (Figure 10-15) contains compounds for which IG_input provided better explanations. IG_input clearly highlighted the nitroso group in compound 6219, whereas the explanation provided by IG_hidden resembles an aromatic amine. A similar observation can be made for compound 6476. Only IG_input correctly identified the isocyanate, while IG_hidden highlighted the nitrogen atoms attached to the respective phenyl ring. For compound 6533, IG_hidden failed to recognise the aziridine ring as a cause for mutagenicity. IG_hidden sometimes failed to provide accurate explanations for compounds containing alerts that are infrequent in the training dataset.

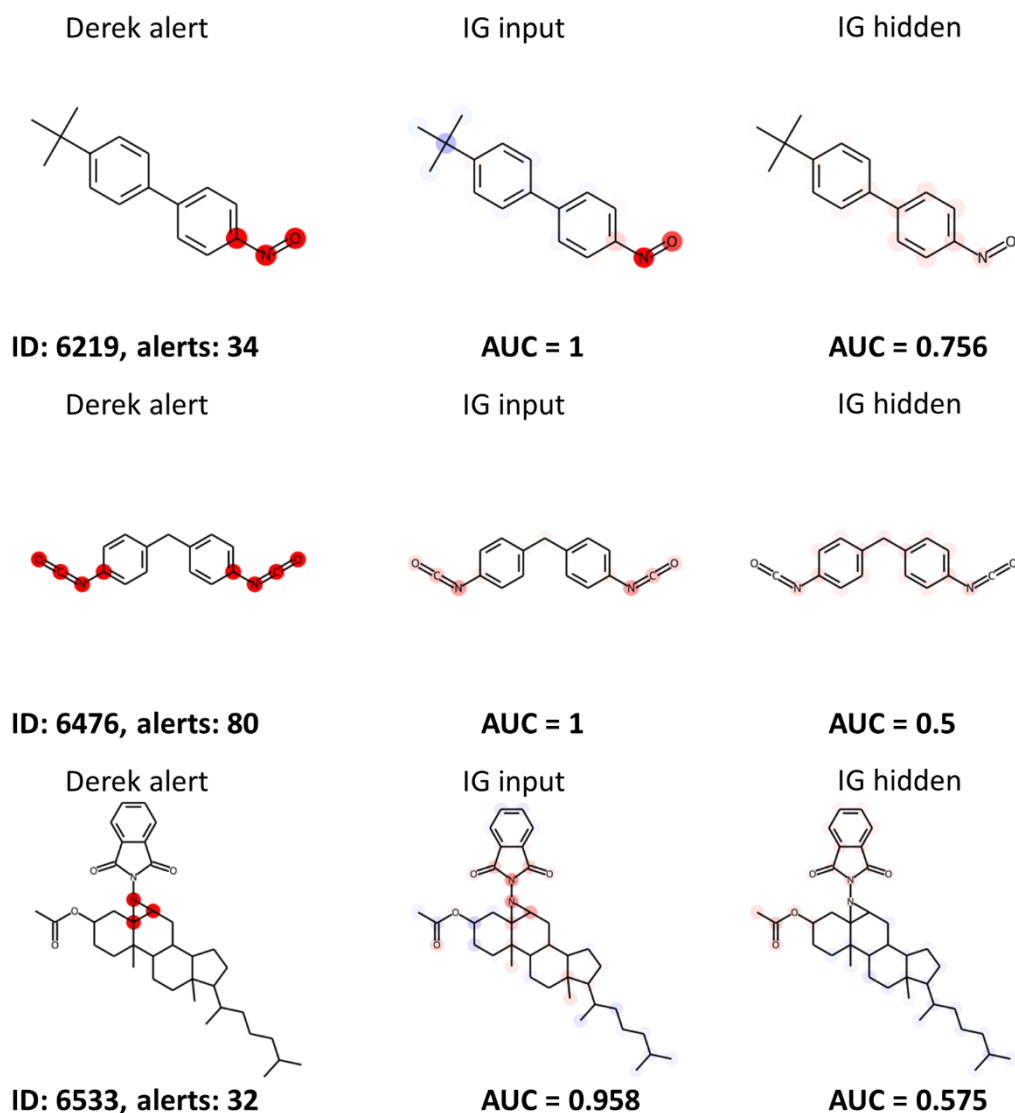


Figure 10-15 Comparison of atom attributions for individual compounds (Part 4). Shown are compounds with Derek alerts highlighted (first column), atom attributions along with the corresponding attribution AUC for IG_input (second column), and IG_hidden (Variation 2 with weights) for *dropout_model*. Alert34: aromatic nitroso, Alert 80: isocyanate or isothiocyanate, Alert32: azirine or aziridine.

10.3.5 Final evaluation on the test set

Table 10-9 reports the various evaluation scores obtained for the test set. For individual compounds IG_input achieved both a higher median AUC score and a higher proportion of compounds with a score of at least 0.8. IG_hidden achieved a slightly higher median alert AUC, yet for a larger proportion of alerts an average AUC of at least 0.8 was obtained when using IG_input. Figure 10-16 reports a compound-wise comparison (10-16A) as well as a scatter plot contrasting average alert AUC scores for IG_input and IG_hidden (10-16B).

It can be seen in Figure 10-16A that while IG_input overall achieved higher AUC scores on individual compounds, for some compounds IG_hidden achieved higher scores. Similar observations were made for the validation set.

It can be seen in Figure 10-16B that for some alerts IG_hidden achieved very low scores compared to IG_input.

Table 10-9 Evaluation of explanations on the test set. The model instance was *dropout_model* and IG_hidden used the parameters from Variation 2 with weights. Shown are median AUC, number of compounds with AUC at least 0.8, median average alert AUC, and number of alerts with average AUC of at least 0.8.

	Median AUC	AUC \geq 0.8	Median alert AUC	Alert AUC \geq 0.8
IG input	0.965	0.765	0.838	0.702
IG hidden	0.938	0.725	0.852	0.532

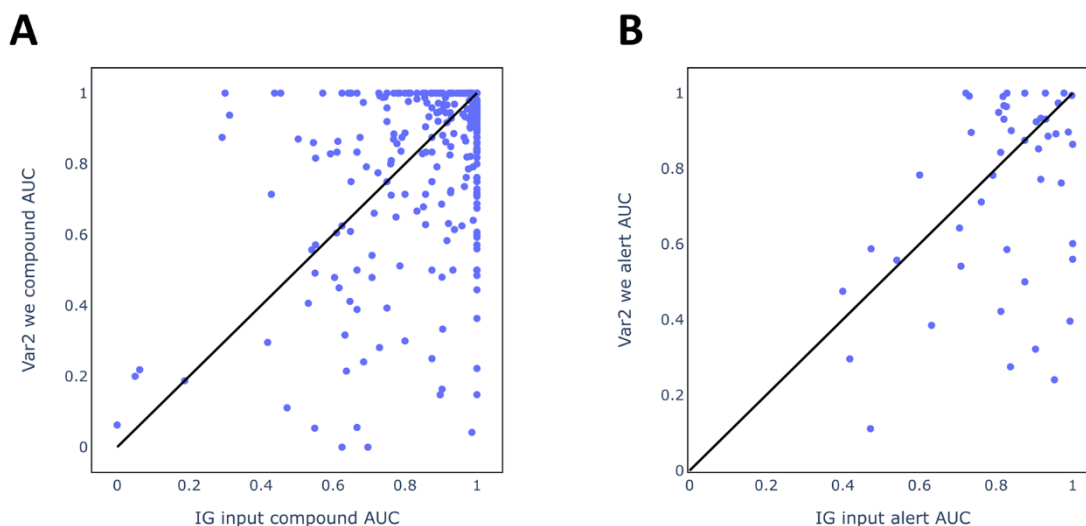


Figure 10-16 Comparisons of compound AUCs and alert AUCs between IG_input and IG_hidden on the test set. The model instance used was *dropout_model*. The IG_hidden method Variation 2 with weights (Var2 we) was used.

10.3.6 Analysis of the proportion of attributions accounted for in IG_hidden

As mentioned above, when explaining the prediction of a test compound using IG_hidden, it may be that the test compound does not match any of the fragments extracted for a given neuron. In that case the attribution for this neuron does not contribute to atom colouring. Proportions of attributions (total, positive and negative) that the model explanation accounts for were reported above for

exemplary compounds (Table 10-8). Overall, the proportions were relatively low with a maximum of 0.36 for total attributions, 0.447 for positive attributions and 0.403 for negative attributions. The most interesting value is arguably the one for positive attributions, as this corresponds to substructures used to explain toxic predictions. Conversely, we expect substructures indicative of non-toxicity to be much rarer, as non-toxicity often may be the result of the absence of toxic chemical features. Figure 10-17A presents the distribution of proportions of positive attributions accounted for in TP compounds as a histogram, while Figure 10-17B shows the proportions and attribution AUCs for those compounds as a scatter plot.

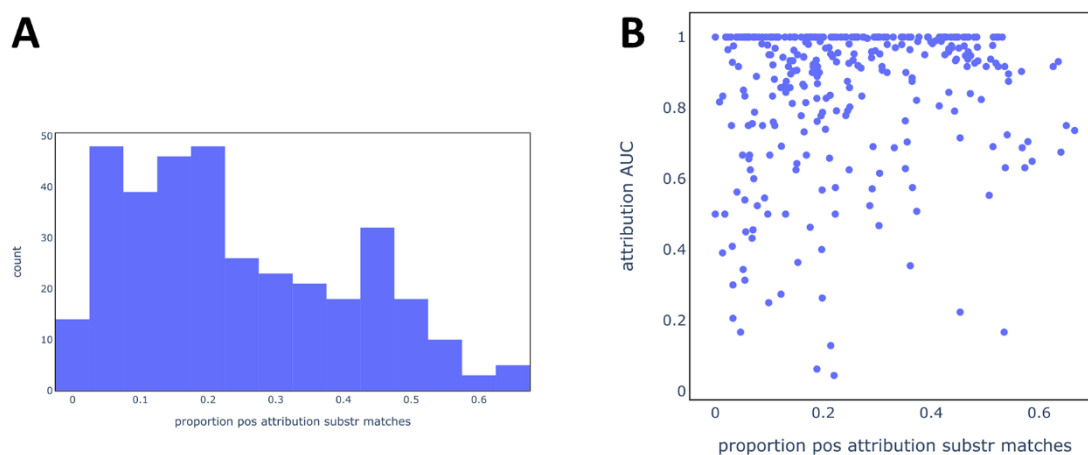


Figure 10-17 Analysis of positive attributions accounted for in model explanations by IG hidden. **A:** Histogram showing proportions of positive attributions accounted for in TP compounds of the validation set. **B:** Scatter plot showing proportions of positive attributions and attribution AUC values for individual TP compounds in the validation set.

It can be seen that for many TP compounds only a small proportion of positive attribution was accounted for in the obtained model explanations. For many compounds this value was below 0.2, while the highest observed proportion across all TP compounds was 0.666. However, the magnitude of the proportions is not correlated with the quality of model explanations. High AUC scores were obtained for low, medium and high proportions of accounted for attributions. To further understand this matter, relevant neurons and corresponding matching and non-matching substructures were analysed for four exemplary compounds. In particular, the three neurons most relevant for a positive prediction were considered. The following figures show either matches of the test compound with extracted fragments for this neuron used to explain the predictions, or, if no matches were found, extracted substructures that are chemically similar to the test compound. Four examples with different characteristics are presented:

- low proportion of positive attribution accounted for, high AUC (compound 6549, Figure 10-18)

- low proportion of positive attribution accounted for, low AUC (compound 5949, Figure 10-19)
- high proportion of positive attribution accounted for, high AUC (compound 6439, Figure 10-20)
- high proportion of positive attribution accounted for, low AUC (compound 6431, Figure 10-21)

In the first example (compound 6549), very few of the neurons relevant for the toxic prediction (and none of the Top-3) contained substructures matching the compound. This resulted in very pale atom colouring (invisible without increasing colour intensity). Nonetheless, all the atoms of the Derek alert received higher attributions than the other atoms which led to a perfect AUC score, while this was not the case for IG_input. When inspecting substructures extracted for the most relevant neurons, examples of aryl hydrazine groups and nitrogen heterocyclic structures can be found which suggests that those neurons did recognise the relevant chemical features, although no exact match for the test compound could be found. Finding more relevant substructures for those neurons by improving the substructure extraction method could improve the explanation by increasing the assigned colour intensities for relevant atoms.

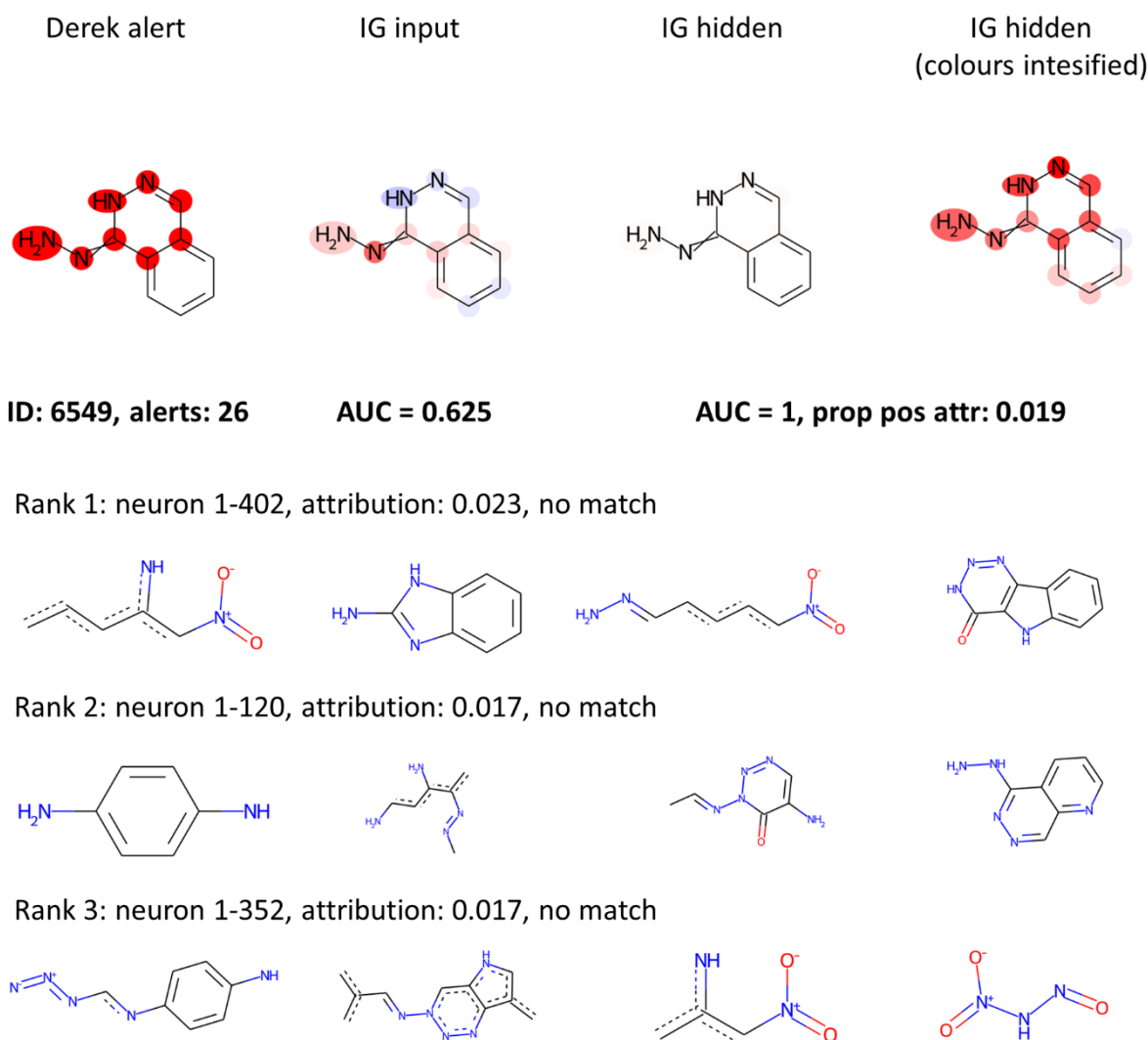


Figure 10-18 Analysis of positive attributions accounted for in compound 6549. Top part: Derek alert and explanations provided by IG_input and IG_hidden. On the right: the colours for IG_hidden are intensified so that the atom with the highest attribution is shown with full colour intensity. Bottom part: The three highest ranking neurons are shown based on attribution value. None of the substructures extracted for these neurons match the test compound. In each case, substructures that are close matches to the test compound are shown. Alert26: Aryldiazine.

As for compound 6549, the atom colourings for IG_hidden are very pale in this example (compound 5949) corresponding to a low number of neurons for which matches could be found. By inspecting substructures for the most important neurons, substructures containing nitroso or N-hydroxy groups can be found. As for the first example, this suggests that those neurons detect the relevant features to some extent, yet no matches for the specific test compound exist. The substructure extraction method would need to be modified to yield more generic substructures for the relevant chemical features (aromatic nitroso and N-hydroxy). In that way a better explanation of the model prediction could be achieved.

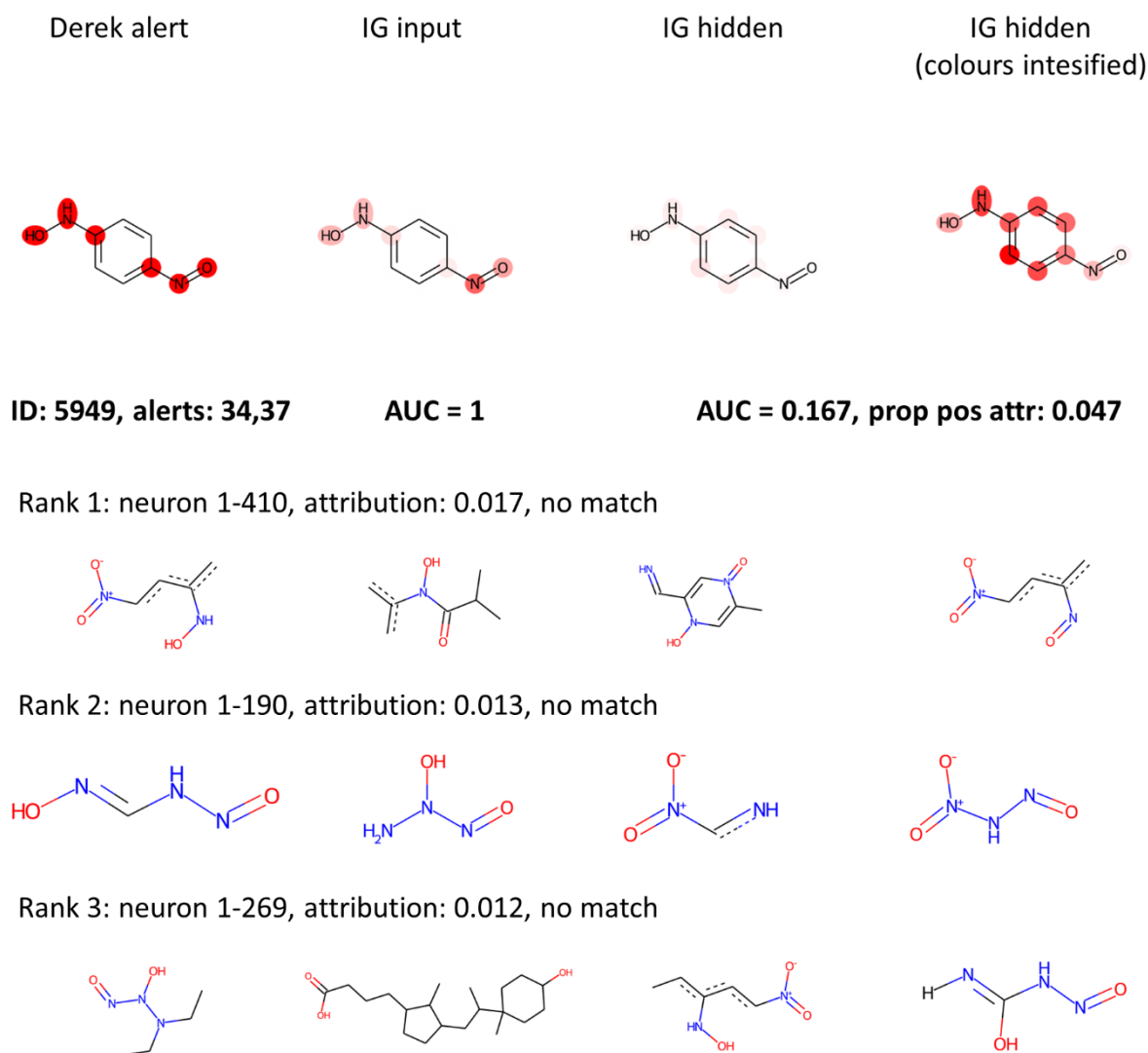


Figure 10-19 Analysis of positive attributions accounted for in compound 5949. Details in Figure 10-18. Alert34: Aromatic nitroso, Alert37: Aromatic hydroxylamine/ester.

For compound 6439, a relative high number of the neurons contributing to the positive prediction (all of the Top-3) had matches with the test compound and this resulted in a good explanation of the prediction. It is worth noting that the test compound (two nitro groups) also matches structures with other functional groups (e.g. nitro with amine in the meta position for neuron 1-236). This is because neither the charge of atoms nor the number of implied hydrogen atoms were considered for the substructure match. In this example around 0.6 of the positive attributions were accounted for and this seems to be sufficient to give a very good explanation.

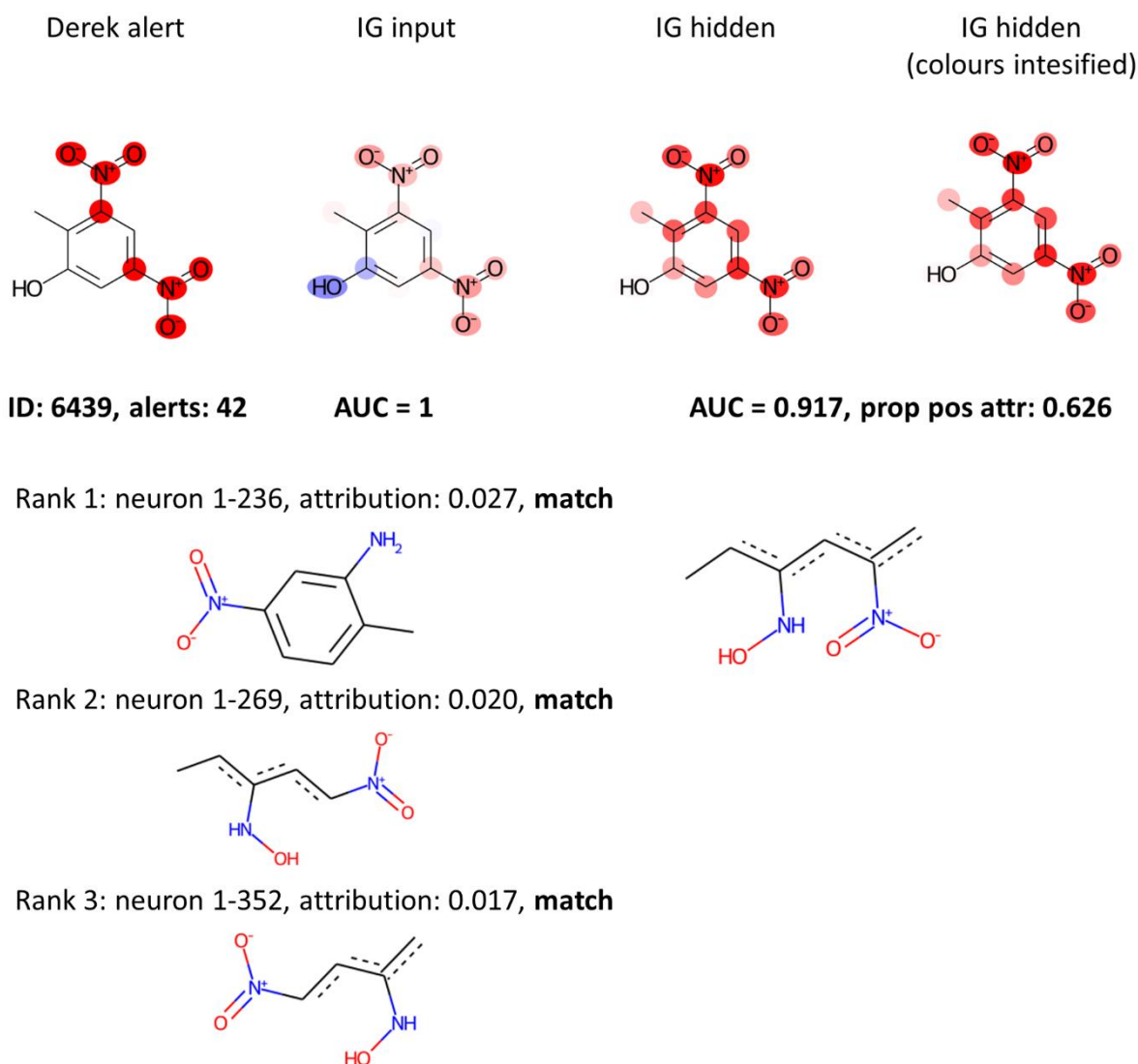
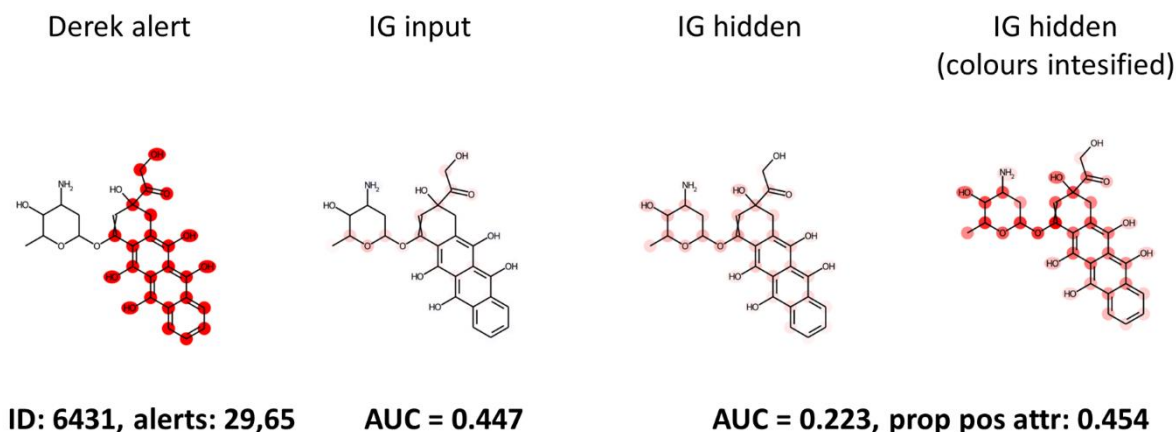


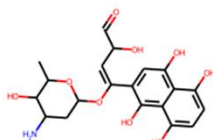
Figure 10-20 Analysis of positive attributions accounted for in compound 6439. Details in Figure 10-18. In this case, matches were found for the test compound for all three neurons. Alert42: Aromatic nitro.

Compound 6431 contains two Derek alert structures: the PAH type system and a precursor of 1,2-dicarbonyl. Both IG_input and IG_hidden partially highlighted these parts of the compound, but relatively low AUC scores resulted from high attributions to atoms of the substituted tetrahydropyran ring, which is not a Derek alert structure. For the most relevant neuron, a match was found which matches parts of the polycyclic system, the 1,2-dicarbonyl precursor and the substituted tetrahydrofuran ring. While there were no matches for the second neuron (1-108), substructures for this neuron include the relevant polycyclic system and the tetrahydrofuran ring. For the third neuron, three matches were found which to some extent match all of the three mentioned chemical features. It seems that the tetrahydrofuran ring is a relevant chemical feature that, at least to some extent, contributes to the prediction made by the model. This means that the incorrect explanation is due to

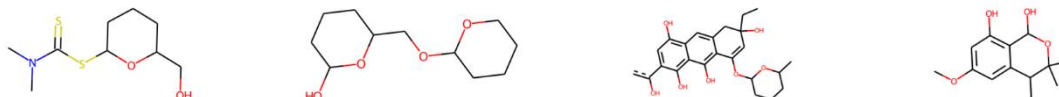
the model making the correct prediction for the wrong reason (the Clever Hans effect (Lapuschkin et al., 2019)) and not due to the explanation method insufficiently explaining the model.



Rank 1: neuron 1-224, attribution: 0.019, **match**



Rank 2: neuron 1-108, attribution: 0.015, no match



Rank 3: neuron 1-34, attribution: 0.014, **match**

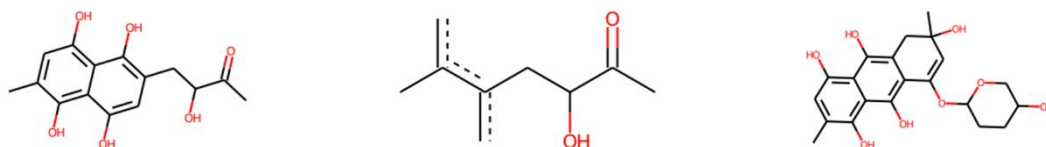


Figure 10-21 Analysis of positive attributions accounted for in compound 6431. Details in Figure 10-18. In this case, for two of the neurons a substructure match was found for the test compound (1-224 and 1-34). Alert29: PAH type alert, Alert65: 1,2-Dicarbonyl compound or precursor.

The observed low proportions of attributions accounted for when using IG_hidden are a limitation of the method. The considered examples suggested that an improved method to find substructures activating neurons may improve the model explanations either by providing a more accurate ranking of atoms (compound 5949) or by increasing colour intensity (compound 6549). In other cases

(e.g. compound 6431), incorrect explanations seem to be the result of the model rather than the explanation method.

10.3.7 Analysing predictions for a model based on experimental Ames labels

After validating IG_hidden using models trained on Derek alert labels, its usefulness was tested on a model trained on experimental Ames labels, as this is how QSAR models and explanation methods are used in practice. In this section, the following were investigated: (i) how well explanations for experimentally positive compounds correspond to Derek alerts, (ii) what can be learned from compounds that are experimentally positive but do not match a Derek alert, (iii) whether explanations for negative predictions can reveal deactivating features.

10.3.7.1 Analysis of TP compounds that are also positive in Derek

Firstly, it was evaluated how well explanations provided by IG_input and IG_hidden correspond to Derek alerts for TP compounds. In Table 10-10, performance on individual compounds (median AUC and $AUC \geq 0.8$) as well as average performance on individual Derek alerts (median alert AUC and alert $AUC \geq 0.8$) are reported. Similar to models trained on Derek labels, the performance on individual compounds seems to be slightly better when IG_input was used. The same is true for the average performance on Derek alerts, however, there were a number of compounds and alerts where IG_hidden outperformed IG_input (see Figure 10-22A and 10-22B). This is in agreement with observations from models trained on Derek labels whereby each method was found to have complementary capabilities. Overall, AUC scores for both methods were somewhat lower compared to models trained on Derek labels. This is plausible since the neural network model generally displayed somewhat lower performance in classifying compounds. Perhaps unsurprisingly, it seems that predicting and also explaining well defined rules such as Derek alerts is a simpler task compared to predicting and explaining experimental results which are prone to experimental uncertainty. Nonetheless, both explanation methods provided good explanations for the majority of the TP compounds.

Table 10-10 Evaluation of explanations extracted from the model trained on experimental Ames labels. Shown are median AUC, number of compounds with AUC at least 0.8, median average alert AUC, and number of alerts with average AUC of at least 0.8.

	Median AUC	AUC \geq 0.8	Median alert AUC	Alert AUC \geq 0.8
IG input	0.905	347/478	0.848	31/54
IG hidden	0.883	306/478	0.814	29/54

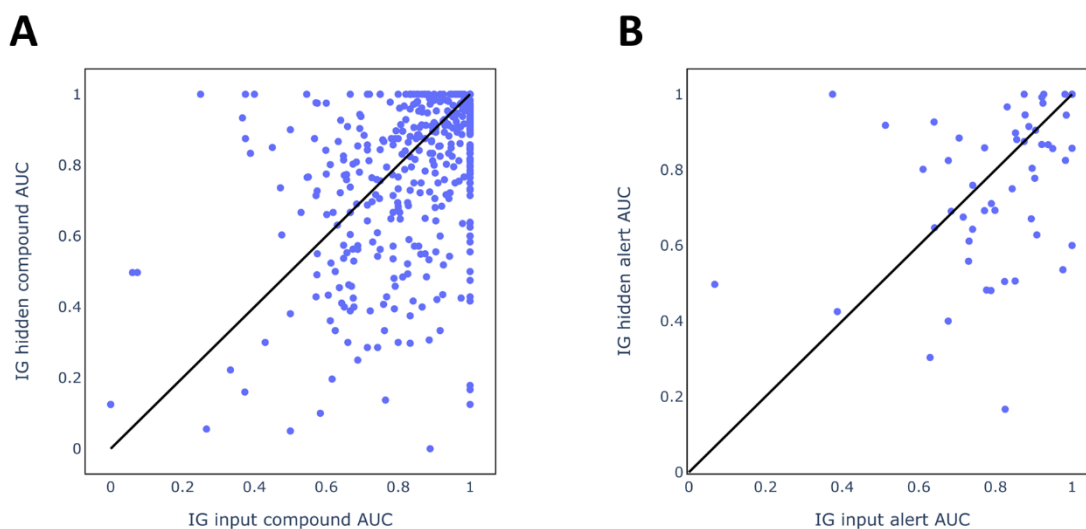


Figure 10-22 Evaluation of individual compounds and alerts for model trained on experimental Ames labels. A: Scatter plot showing attribution AUC scores for individual compounds for IG_input and IG_hidden. B: Scatter plot showing average attribution AUC scores for individual alerts.

10.3.7.2 Analysis of TP compounds that are negative in Derek

Next, TP compounds that do not match any of the Derek alerts were inspected. A total of 78 such compounds were found in the validation and test set. In Figure 10-23, six exemplary compounds are shown. These compounds are of interest as they may hint at potential refinements of Derek alerts to increase their sensitivity to flag Ames mutagenic compounds. Another possibility is that these compounds in principle do match Derek alerts but are still reported as negative due to the Derek system having access to the same compound being measured as negative. This would supersede the match and result in a negative Derek label. In this case, this information could be used to curate the data in the Derek system by comparing the label to the source dataset used in the present study.

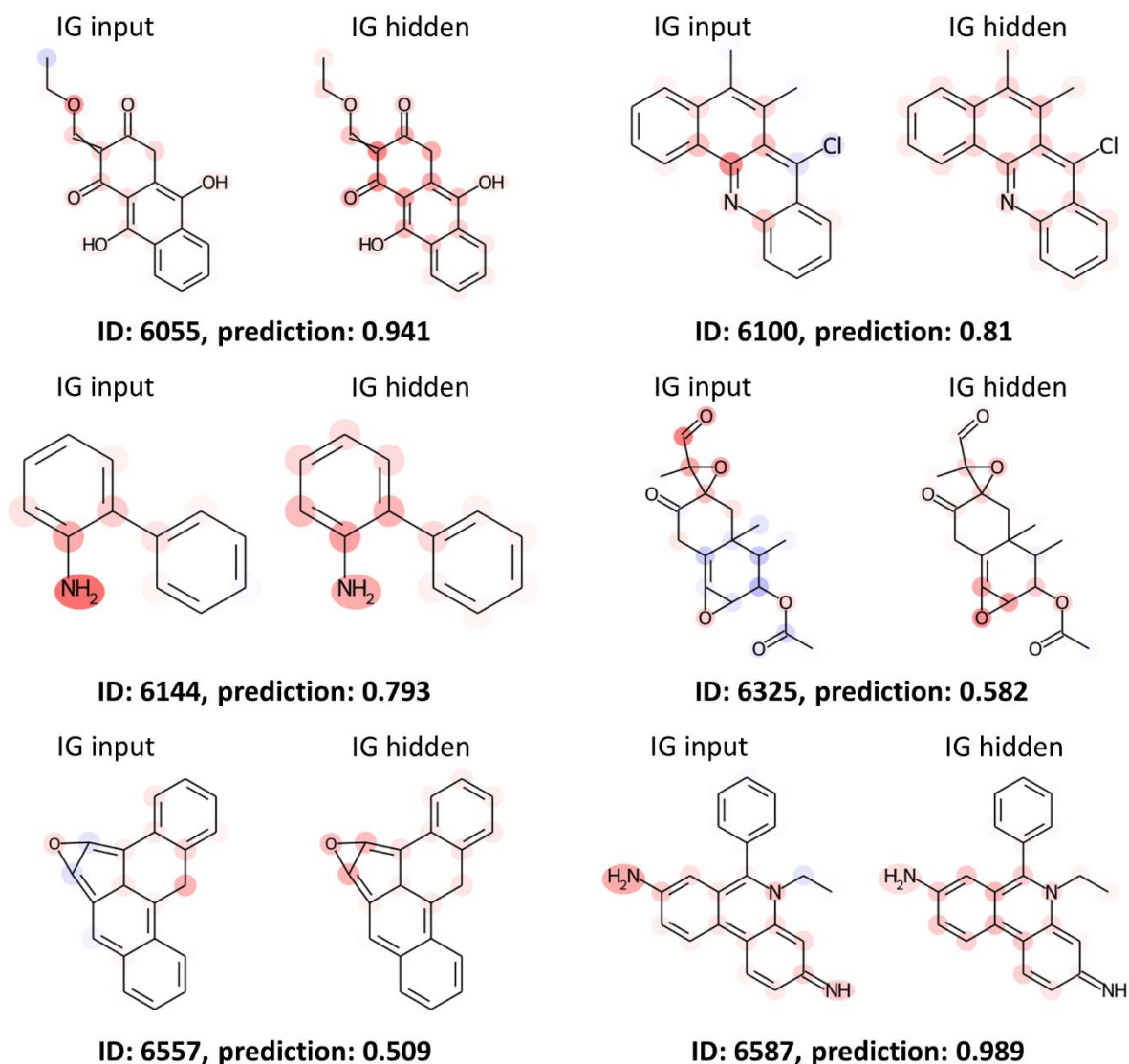


Figure 10-23 Explanations for TP compounds that are negative in Derek. Shown are generated model explanations from IG_input and IG_hidden for compounds correctly predicted as toxic in the Ames test, while being labelled as negative by the Derek software.

If a compound does not match a Derek alert, the explanation for a positive prediction may hint at substructures that might be good candidates for a new or refined Derek alert. Notably, explanations provided by IG_input and IG_hidden may provide complementary information. For compound 6055, IG_input most strongly highlighted an ether group next to a double bond, whereas IG_hidden highlighted the cyclodiketone moiety. Compound 6100 contains an acridine structure with an additional fused phenyl ring. Both methods highlighted certain parts of that polycyclic aromatic system. Compounds 6144 and 6587 contain aromatic amine groups which were highlighted. In addition, IG_input highlighted the imine in compound 6587, whereas IG_hidden highlighted large parts of the polycyclic system. In compound 6557, IG_hidden clearly highlighted the epoxide structure,

whereas an aliphatic ring carbon received the strongest attribution in IG_input. Compound 6325 contains two epoxide groups and both were highlighted by IG_hidden, whereas IG_input highlighted one of the two along with an aldehyde group.

Overall most of the shown compounds contain chemical groups covered in Derek alerts. Their negative label may stem from a negative experimental result in the Derek system or an unusual chemical environment in which the alerts occurs which might result in the compound not matching the alert. In the latter case, the model explanations for such compounds may aid to refine the definition of Derek alerts.

10.3.7.3 Analysis of TN compounds

In the final section, how well the attribution methods can explain negative predictions was investigated. Figure 10-24 presents explanations generated with IG_input and IG_hidden for some representative TN compounds. Table 10-11 reports the proportion of attributions accounted for with matches, as was also investigated above for the model trained on Derek labels.

Compound 5892 contains an aromatic amine, yet it was negative in the Ames test due to the effect of the sulfonate group and the chlorine which both withdraw electron density from the aromatic ring which reduces stability of the nitrenium ion intermediate essential for the compound's mutagenicity (Furukawa et al., 2022). Both methods assigned positive attributions to the amine group and IG_input assigned negative attributions to parts of the sulfonate group and the aromatic carbon next to the chlorine, but not the chlorine itself. With IG_hidden only a very pale negative attribution was assigned to the sulfonate group.

In compound 5965, atoms in the polycyclic system received positive attributions using both methods. IG_input assigned negative attributions to the aliphatic carbon and the attached hydroxyl group, whereas no negative attributions were obtained by IG_hidden. This can be explained by the fact that only around 0.02 of the negative attributions for this compound can be explained with matches (see Table 10-11). Overall, matches for hidden neurons with a negative attribution were less frequent than the ones with positive attributions. This means that IG_hidden can better explain tendencies of the model to make positive predictions as opposed to negative ones.

Compound 5979 contains a structural alert for mutagenicity (aromatic nitro) as well as a deactivating group (trifluoromethyl). Both IG_input and IG_hidden correctly assigned corresponding attributions,

but the intensity for the trifluoromethyl group in IG_input is more pronounced thus correctly reflecting the negative prediction.

Also compound 6337 contains both atoms with positive and negative attributions. The nitro group was assigned positive attribution while atoms in the aliphatic carbon chain were assigned negative attributions. This can be observed both for IG_input and IG_hidden.

No positive atom attributions can be found in the remaining two compounds (6443 and 6564). In 6443, IG_input assigned negative attributions to both the tert-butyl group and atoms in the 5-membered ring attached to the phenyl group. IG_hidden assigned (very weak) negative attributions to atoms in the 5-membered ring and attached methyl groups. Both IG_input and IG_hidden assigned negative attributions to atoms of the aliphatic carbon chain for compound 6564.

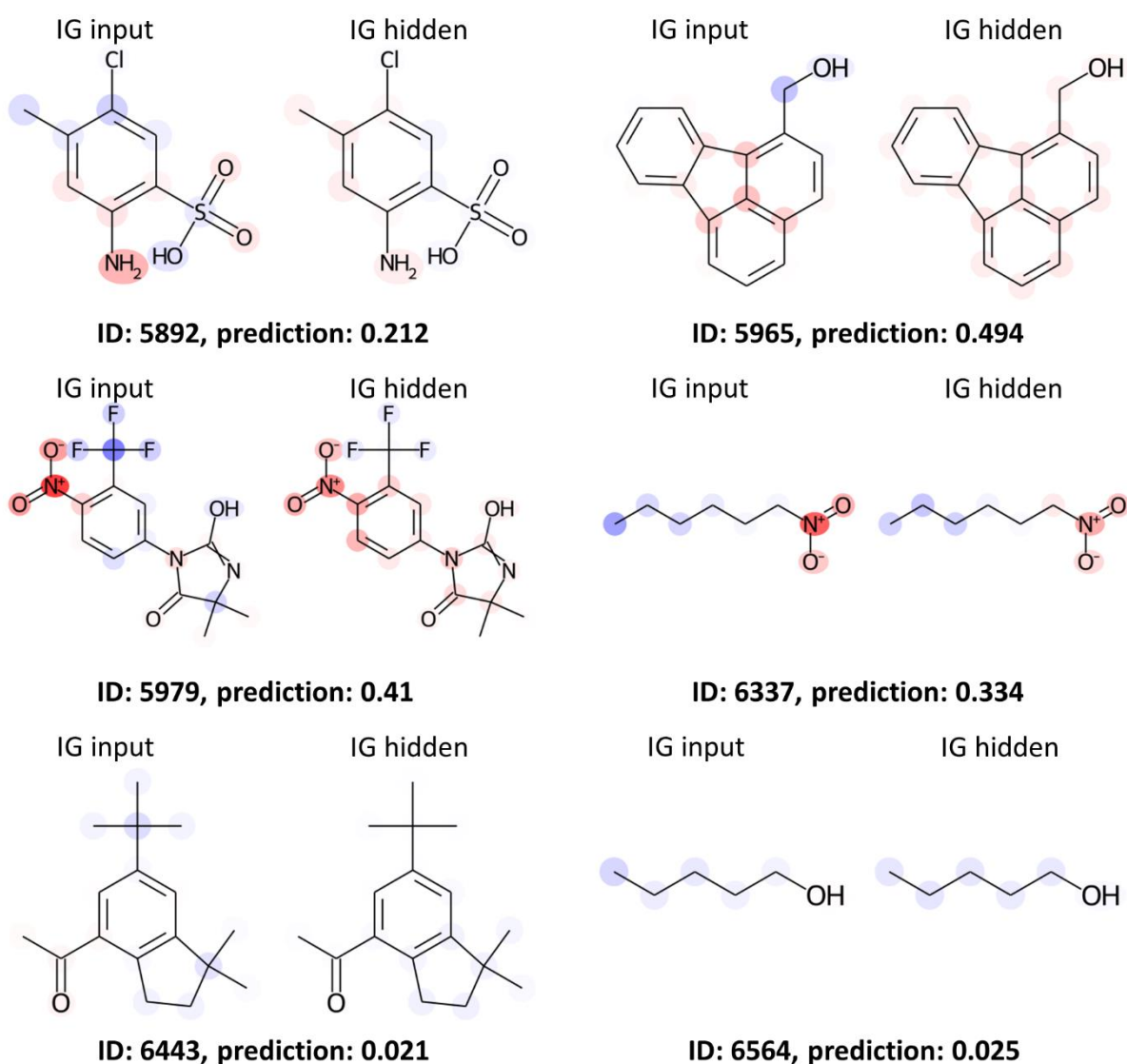


Figure 10-24 Explanations for TN compounds. Shown are generated model explanations from IG_input and IG_hidden for compounds correctly predicted as non-toxic in the Ames test.

Table 10-11 Overview of atom attributions accounted for by substructure matches in IG_hidden method for compounds shown in Figure 10-24. More details on how values were obtained in the caption of Table 10-8.

ID	Proportion absolute neuron attributions accounted for	Proportion positive neuron attributions accounted for	Proportion negative neuron attributions accounted for
5892	0.258	0.272	0.244
5965	0.161	0.223	0.024
5979	0.521	0.601	0.407
6337	0.303	0.275	0.341
6443	0.217	0.218	0.216
6564	0.203	0.137	0.222

For some of the above compounds parts of the molecule support a positive prediction, while other parts of the compound support a negative prediction. In other compounds structural features supporting a positive prediction are not present and the negative prediction may be attributed to certain parts of the compounds. In cases where there is no strong negative attribution, the negative prediction can be interpreted as resulting from the absence of mutagenic features.

It can be observed that the colour intensities are not always perfectly consistent with the prediction. For instance, for compound 5979 red colouring seems to be more prevalent across the molecule than blue colouring (for both IG_input and IG_hidden), yet the compound was predicted as negative. This is due to the fact that the attributions result from a comparison to a baseline (empty bit vector). The prediction for the baseline notably is not neutral (i.e. 0.5), but instead around 0.2 for the model trained on experimental Ames labels. This means for IG_input that red colour intensities will be stronger than blue ones if the predicted probability is higher than the one for the baseline. The same effect applies to IG_hidden. Moreover, for IG_hidden positive or negative attributions may not be mapped to the structure if no matching substructures have been extracted for the corresponding neuron which may also lead to inconsistencies in model prediction and atom colouring. Implications of those factors will be further discussed in Discussion.

10.4 Discussion

In this chapter, neural network models were explained by using chemical substructures found to activate individual neurons in the network. By evaluating the model explanations using the Derek

alerts, the workflow for extracting substructures could be optimised (discussed in 10.4.1). In section 10.4.2, the developed method to explain model predictions (IG_hidden) is compared IG_input as an established method to interpret neural networks. Moreover, it is reflected how explanation models may be best evaluated. After initially investigating models trained on the Derek alerts as labels, a model trained on experimental Ames labels was investigated to determine the usefulness of the developed technique in a more practical setting (discussed in 10.4.3). Finally, it is discussed how the obtained model explanations (in terms of atom attributions) can be best visualised to a user of the method.

10.4.1 Modifications to the substructure extraction workflow

The workflow for automatic extraction of substructures activating hidden neurons described in the previous chapter was evaluated both globally, by analysing the entirety of extracted substructures, and locally, by considering explanations provided for individual predictions. It was found that the original workflow extracted substructures relevant for frequent alerts but failed to extract substructures for many of the less frequent alerts. This is consistent with the observation of relatively low median alert scores which indicate a poor performance for many of the alerts.

To address this weakness, several modifications were made to the original workflow. First, the thresholds for compounds considered to be strongly activating and fingerprint bits considered to be significant were adjusted to increase the number of compounds and bits included in the FCA. This resulted in a much larger number of FCs, however, by introducing *ThreshNoveltyFC* (ensuring FCs are novel with respect to already considered FCs), the number of FCs that had to be analysed was limited. Another adaptation was to set *ThreshSupportFC* to zero. This means that FCs corresponding to chemical patterns infrequent in compounds activating the neuron may also be considered. The changes were effective in extracting fragments corresponding to rare Derek alerts as well as increasing explanatory performance for many of the alerts.

Overall, the changes resulted in a much larger number of extracted substructures across the neurons which may complicate a manual analysis of those substructures. However, since the substructures are organised in hierarchical networks, it is possible to start by only inspecting generic structures at the top of the subnetworks. In contrast, when interested in substructures causing very strong activation, the substructures may be filtered by this condition yielding typically more specific substructures. Furthermore, the running time of the modified workflow was increased compared to the original. Optimising the running time was beyond the scope of this project, however, the extraction process is,

in principle, very well suited to parallelisation as the extraction of substructures is done independently for each neuron. By distributing the programme across different computational cores, the running time could be significantly reduced.

Notably, all tested variations of the original workflow (Variation 1-4) resulted in marked increases in explanatory performance compared to the original workflow for both *best_model* and *dropout_model* (see Table 10-6). All variations included modifications to improve extraction of rare alerts. This suggests that the success of the workflow may not be very sensitive to the exact parameters used. However, the suitability of the workflow on other toxicity endpoints with other relevant features needs to be evaluated.

10.4.2 Comparison of IG_hidden with IG_input

Overall, IG_input and IG_hidden achieved comparable median AUC scores on individual compounds and averaged AUC scores for compounds matching specific Derek alerts. For both attribution methods, different compounds classes were explained with different degrees of success. Some of the alerts (e.g. aromatic nitro, epoxide) were very well explained by both methods. Other alerts were explained well only by one of the methods (e.g. IG_input was better for isocyanate whereas IG_hidden was better for halogenated alkene). Generally, very frequent alerts in the training set were well explained by both methods. IG_hidden struggled to provide good explanations for some of the rare alerts. In these cases, the workflow for extracting substructures may fail to find a number of appropriate substructures in all relevant neurons. Table 10-5 demonstrates that superstructures for almost all alerts were retrieved. However, to explain the predictions of individual compounds, the substructure must match the test compound. Having extracted a superstructure for a certain alert does not guarantee an accurate explanation for any compound matching that alert. In many such cases IG_input provided better explanations. On the other hand, IG_input sometimes failed to explain alerts constituted by large fragments for which IG_hidden provided more accurate explanations. IG_hidden in general highlights larger proportions of a compound in positive predictions.

Interestingly, the two attribution methods may provide quite different explanations for the same model predicting the same compound. This illustrates that each method is merely a model attempting to rationalise a prediction, yet the 'true' cause remains elusive. In some cases, both methods provide very similar explanations which also match the known cause of a Derek alert. In such cases, it would seem reasonable to assume that the attribution methods are well aligned with the actual model behaviour. In cases where only one of the attribution methods provides a correct explanation,

weaknesses of one attribution method may be found. If the other attribution method correctly explains the prediction, it can be suspected that the model gave the correct prediction for the right reason. Finally, a model may make a correct prediction but neither attribution method provides a correct explanation. This may be either because both attribution methods are not well aligned with the actual model in this case or because the model made the correct prediction for the wrong reasons (the Clever Hans effect) (Lapuschkin et al., 2019). To distinguish these cases, more in-depth analyses involving more different attribution methods may be required.

In general, more benchmark datasets are required where the ground truth is known. The benchmarks should cover a wide range of chemical patterns (e.g. small and large groups as the ground truth) to test how robust the methods are with respect to those different cases. Many previous studies reported successful (and in part unsuccessful) examples for various attribution methods without conducting a systematic analysis for the entire dataset (Harren et al., 2021; Jiménez-Luna et al., 2021; Preuer et al., 2019; Rodríguez-Pérez & Bajorath, 2020b). Other studies more systematically evaluated the quality of explanations by considering simple rules (such as atom counts, or the presence of certain functional groups) (Matveieva & Polishchuk, 2021; Sheridan, 2019) or activity cliffs (Jiménez-Luna et al., 2022) to obtain ground truths. In this work, toxicophores were found to be a suitable benchmark to evaluate the performance of attribution methods. Mutagenicity can be considered an insightful endpoint as a large number of diverse chemical features are known (Kazius et al., 2005) which all must be detected by a model. While Derek Nexus is a proprietary software, publicly available collections of toxicophores (e.g. ToxAlerts) (Sushko et al., 2012) may serve as a benchmark to the whole community.

In practice, different attribution methods may be used in a complementary fashion. However, to interpret these, good knowledge of the strengths and limitations of different approaches will be required. The approach introduced in this work (IG_hidden) is conceptually different to various approaches that focus on input features (IG_input, SHAP, LIME) and hence it may be particularly suited to provide unique insights.

10.4.3 Applying IG_hidden to a model trained on experimental Ames labels

It was shown that IG_hidden also provides meaningful explanations when applied to a model trained on experimental Ames data rather than the Derek alert labels. Similar observations as for the models trained on the Derek labels were made. Overall, IG_input seemed to provide more accurate explanations for TP compounds, yet for several compounds and alerts IG_hidden provided better explanations thus confirming the complementarity of the approaches.

For toxic compounds not matching Derek alerts, model explanations may be useful to refine Derek alerts or to propose novel structural alerts for a different endpoint. However, a prerequisite for meaningful model explanations is good performance of the respective model to predict toxicities (P. Polishchuk, 2017). While Ames mutagenicity is a well understood toxicity endpoint that can be fairly well predicted with QSAR models, this may not be the case for less well understood toxicity endpoints including in vivo toxicities. The suitability of model explanations to discover novel toxicophores for complex toxicity endpoints would need to be demonstrated in further studies.

Most focus in this work has been put on correctly explaining the cause of toxicity. However, making negative predictions is important as these indicate the safety of a chemical (Williams et al., 2016). Absence of toxicity may, in principle, be due to the lack of toxic features or due to features deactivating toxic features within the compound. For instance, moieties reducing the electron density in an aromatic system prevent mutagenicity of aromatic amines as the stability of nitrenium ions is reduced (Furukawa et al., 2022). QSAR models may correctly predict such effects and explanation methods may provide the correct mechanistic explanation. By inspecting compounds that match known structural alerts while being predicted as negative, novel deactivating moieties may be detected by model explanation methods. However, it has to be stated that in the present work IG_input more reliably highlighted negative features compared to IG_hidden. IG_hidden may be able to identify hidden neurons that contribute to negative predictions, yet in many cases the proposed method for substructure extraction did not find fragments matching test compounds. In order to improve the extraction workflow, focus might also be put on how well known deactivating features can be retrieved for relevant neurons in order to correctly explain negative predictions. However, deactivating features are less well known and understood than structural alerts (toxicophores) for mutagenicity, as well as other toxicity endpoints, which makes it more challenging to validate explanations made for negative predictions.

10.4.4 Depiction of atom attributions

The two attribution methods (IG_input and IG_hidden) were used to generate atom attributions indicating the importance of atoms to the model predictions for test compounds. As in previous studies, a colour scale was used to visualise the magnitude of atom attributions (Feldmann et al., 2021; Harren et al., 2021). The scale in this work used red to indicate atoms contributing to toxic predictions and blue to indicate atoms contributing to non-toxic predictions. To make explanations for different compounds comparable, an absolute colour scale was employed where colour intensities are relative

to the largest atom attributions found in the dataset. However, different scales were used for IG_input and IG_hidden due to the observation that generally larger atom attributions occurred for IG_input.

A significant observation is that the summed colour intensities are not always consistent with the predicted class. In particular, a user might expect that for a compound predicted as negative blue colour intensities would exceed red colour intensities, and vice versa for positive predictions. However, for the IG approach, attributions for input features or hidden neurons are obtained in relation to a baseline, for which an empty bit vector was selected. For the models considered in this work the predicted probability for an empty bit vector was not neutral, but rather a clearly negative prediction (e.g. ~ 0.2 for *dropout_model* where 0.5 is considered neutral). Compounds with a predicted probability larger than this will have a stronger intensity of red colour, yet may be predicted as negative if the predicted probability is below 0.5. This situation could be prevented if the difference in attributions between the baseline and a neural prediction were distributed equally among all atoms in the compound. In other words, the colour for each atom would be shifted towards blue to some degree so that blue colour intensity would be equal to red colour intensity if the predicted probability is equal to 0.5. For this work it was decided to not make this adjustment. Red colours in negative predictions can be interpreted as features supporting a positive prediction while not being enough to observe toxicity (according to the prediction). It follows that the meaning of colours needs to be made clear to any user of the model explanations to prevent misinterpretations.

The inconsistencies between atom colouring and model predictions described in the previous paragraph apply both to IG_input and IG_hidden. However, an additional factor contributes to this result for IG_hidden. If no substructure matches exist between a test compound and substructures extracted for a neuron, the attribution for this neuron is not used to colour atoms. It was observed for *dropout_model* that typically less than half of the total attribution is accounted for in atom colourings. This is the reason why lower atom attributions are observed for IG_hidden when compared to IG_input so that that different colour scales are necessary for visualisation (see above). The low proportion of attribution accounted for in model explanations can be considered a limitation of IG_hidden, but it has to be stated that the ranking of atoms nonetheless corresponds to good model explanations evaluated using attribution AUC scores. One consequence is that if the proportion of attribution accounted for is particularly low, then atom colourings will be very pale or even not visible to a user. Pale colours can be interpreted as if IG_hidden is not able to make a confident explanation of a prediction (due to the lack of matching substructures), although the proportion of attribution accounted for in the model explanation is not correlated to the quality of explanations. Either way, increasing the proportion of attribution would be desirable by, for example, modifying the workflow for extracting substructures. As was shown for some examples, the lack of matching substructures

may indeed lead to an incorrect explanation which could be remedied if a superior method for extracting substructures can be found in the future. This also shows that IG_hidden can be further optimised to potentially become an even more powerful tool to explain model predictions.

10.5 Conclusion

In this chapter, neural network models were trained on Derek labels in order to measure the explanatory performance of the proposed method for explaining model predictions. In the global evaluation, it was found that the substructure extraction covers most Derek alerts present in the dataset. Overall, good explanations were achieved for both IG_input and IG_hidden, while each method provided superior performances for certain compounds and Derek alerts making them, in principle, complementary. A limitation of IG_hidden was that substructure matches to the test compound were not found for all relevant neurons.

All models studied in this chapter were neural networks with one hidden layer. In the following chapter attempts are made to extend the method to networks with at least two hidden layers.

Chapter 11 Explaining predictions for deep neural networks

11.1 Introduction

The aim of this chapter was to explore the usefulness of the developed approach for interpretability (IG_hidden) when explaining predictions made by neural networks with more than one hidden layer (i.e. DNNs). This is of relevance as the neural networks used for toxicity prediction, and for most QSAR tasks, typically have more than one hidden layer and hence demonstrating the applicability of the developed approach for those models is necessary.

First, the Derek dataset (see Chapter 10) was used to train and analyse a DNN model with 2 hidden layers. Then the IG_hidden method was applied to the first hidden layer of the model to determine if the predictions could be explained. Finally, the neurons in the second hidden layer were explored for the Derek model as well as 2-layer networks trained on further toxicity and bioactivity datasets. This was done to determine whether IG_hidden could be applied to those neurons.

11.2 Methodology

11.2.1 Model training and evaluation

Neural networks containing two hidden layers were trained on a number of datasets in order to analyse whether DNNs may outperform neural networks which consist of a single hidden layer. The datasets are the one with Derek alerts as labels (see Chapter 10), and assays from the Tox21 dataset (see Chapter 5) and the ToxCast (see Chapter 6) dataset. In addition, a number of bioactivity datasets obtained from ChEMBL were investigated (Bosc et al., 2019). Only assays with at least 2000 labels and at least 10% actives were retained for the various datasets. This resulted in the inclusion of four assays for the Tox21 dataset, 43 assays for the ToxCast dataset and 52 assays for the ChEMBL dataset. For the Derek dataset, the training, validation and test set were unchanged. For the individual assays from the other datasets, the data was split into training (80%), validation (10%) and test (10%). The training data was used to train the model and the validation set was used find the best set of hyperparameters. The test data was used to evaluate the final model.

For the Derek dataset, the hyperparameters tested to find the best performing two-layer model are reported in Table 11-1. The best performing model was compared to the one layer models used in Chapter 10. Note that all models used dropout, due to the better performance for feature extraction.

Table 11-1 Parameters used for hyperparameter optimisation (Derek dataset). Selected hyperparameters in bold

Hyperparameter	Tested values
Neurons first layer	512
Neurons second layer	128, 256, 512
Batch size for optimisation	16 , 32, 64
L2 regularisation of neuron weights	0 , 0.00001, 0.001
Dropout first layer	0.2 , 0.5
Dropout second layer	0.2 , 0.5
Learning rate	0.0001 , 0.00033, 0.001

For the other datasets, the hyperparameters tested are reported in Table 11-2. In these cases, models were trained with one hidden layer and the performance was compared to models trained with two hidden layers. Note that the radius of the Morgan FP was also used as a hyperparameter for these datasets. Moreover, ‘weight’ was used as a hyperparameter to account for the imbalance of labels in some of the assays. When the option ‘balanced’ was used, the loss for data instances belonging to the minority class were scaled up according to the imbalance ratio using the function `compute_class_weight` in scikit-learn.

Table 11-2 Parameters used for hyperparameter optimisation (Tox21, ToxCast and ChEMBL dataset).

Hyperparameter	Tested values
Morgan FP radius	1, 2
Neurons first layer	512
Neurons second layer	0, 256, 512
Batch size for optimisation	32, 64
L2 regularisation of neuron weights	0, 0.00001, 0.001
Dropout (both hidden layers)	0.2, 0.5
Learning rate	0.0001, 0.00033, 0.001
weights	1, ‘balanced’

For all the models, early stopping was applied in the same way as described in Chapter 8 and Chapter 10. That is, the ROC-AUC score of the model was evaluated on the validation set after each epoch (10

epochs in total) and among these the best model was retained as the model instance which was compared to other hyperparameter configurations.

11.2.2 Interpreting DNN models by applying IG_hidden to the first hidden layer

The IG_hidden procedure, which was successfully applied to a neural network with a single hidden layer, was applied to the first hidden layer of a DNN to determine if substructures could be extracted that explain model predictions. This was done by using the optimised workflow for substructure extraction from Chapter 10 (Variation 2 with weights) for the DNN trained on Derek labels. As in the previous chapter, model explanations for IG_hidden and IG_input were compared to the ground truth (i.e. Derek alert atoms) and attribution AUC scores were computed for TP compounds in the validation set. Moreover, average alert AUC scores were determined.

11.2.3 Exploration of neurons in the second hidden layer

The relationship between various hidden neurons was analysed by calculating pairwise Pearson correlation coefficients between neurons, as described in Chapter 8. Each neuron was represented by a d-dimensional vector where each element represented the activation that the respective training compound caused.

Moreover, an attempt was made to identify neurons detecting chemical features that were not already detected in neurons of the first hidden layer. It was assumed that a neuron detecting novel chemical features would not have high correlation with any neuron in the first hidden layer. Also the weight of the neurons in the second layer to the output neuron were considered, as only neurons with a positive weight contribute to a toxic prediction. To understand the chemical features learned by the neuron, the compounds most strongly activating the neuron were inspected. In addition, it was analysed whether compounds matching certain alerts cause strong activation in relation to compounds matching no Derek alerts.

Finally, it was analysed whether individual neurons detect specific chemical features or rather detect toxicity in general. To that end, the ROC-AUC value was used as a metric. In particular, the activation of each compound in a neuron was considered as 'prediction' which yields a ranked list of compounds for each neuron. This ranked list was compared to Derek labels and allows the computation of a

ROC-AUC value. A high ROC-AUC value would be obtained, when a large number of toxic compounds activate the neuron more strongly compared to non-toxic compounds.

11.3 Results

11.3.1 Evaluation of Derek models

Neural networks with two hidden layers (i.e. DNNs) were trained on the Derek dataset and the best model was compared to the model instances analysed in Chapter 10. It can be seen in Table 3 that, across a range of different classification metrics, the DNN (*2layer_model*) outperformed both one-layer networks on the validation set, albeit by a small margin. *2layer_model* was only outperformed by *dropout_model* on recall and *best_model* achieved the same specificity. However, when considering the test set, *2layer_model* achieved the highest score only for precision and specificity. Overall, the observed differences were quite small between the models.

Table 11-3 Classification metrics for Derek models. Various metrics for *best_model*, *dropout_model* and *2layer_model* are compared on the validation and test set. In bold the highest score on validation or test set across the different models.

	<i>best_model</i> validation/test	<i>dropout_model</i> validation/test	<i>2layer_model</i> validation/test
ROC-AUC	0.974/ 0.966	0.970/0.965	0.977 /0.964
Accuracy	0.914/ 0.908	0.903/0.900	0.918 /0.906
Balanced Acc.	0.914/ 0.908	0.902/0.898	0.918 /0.907
Precision	0.918 /0.919	0.889/0.897	0.918 / 0.932
Recall	0.915/0.912	0.929 / 0.922	0.923/0.892
Specificity	0.913 /0.904	0.876/0.874	0.913 / 0.922
F1	0.917/ 0.916	0.908/0.910	0.921 /0.914
MCC	0.828/ 0.816	0.807/0.799	0.836 /0.812

11.3.2 Interpreting the DNN by applying IG_hidden to the first hidden layer

IG_input and IG_hidden (applied to neurons in the first hidden layer) were used to explain predictions of *2layer_model*. The median AUC on TP compounds, the proportion of compounds with an AUC of at least 0.8, the median alert AUC, and the proportion of alerts with an average AUC of at least 0.8 are

reported in Table 11-4 for each method. Attribution AUCs and alert AUCs for individual compounds and alerts are shown in Figure 11-1.

Table 11-4 Evaluation of explanations for the 2-layer Derek model. Comparisons were made for the model instance *2layer_model* on the validation set. IG_hidden used the parameters from Variation 2 with weights. Shown are median AUC, number of compounds with AUC at least 0.8, median average alert AUC, and number of alerts with average AUC of at least 0.8. The better approach for each metric is in bold.

	Median AUC	AUC \geq 0.8	Median alert AUC	Alert AUC \geq 0.8
IG_input	0.984	0.841	0.907	0.714
IG_hidden	0.938	0.725	0.906	0.735

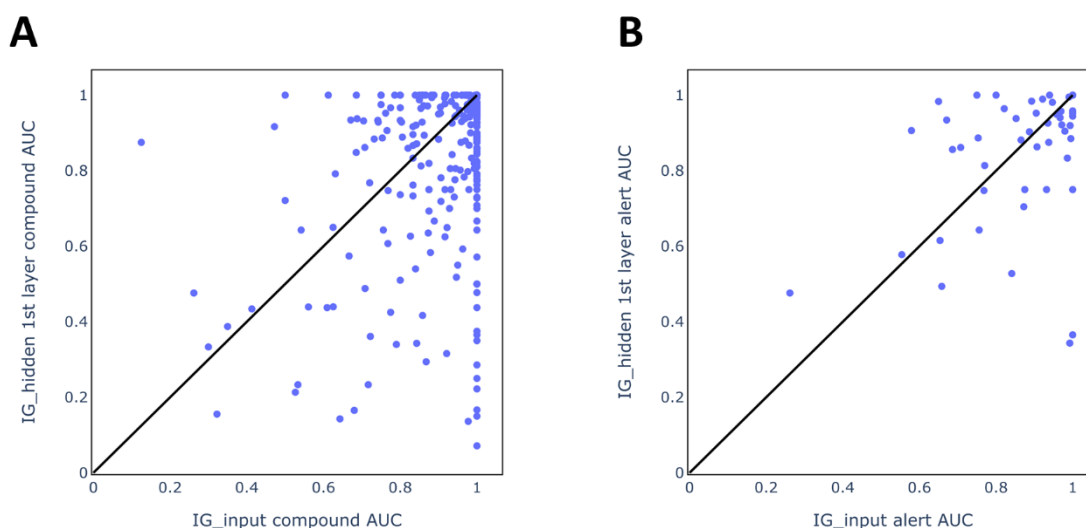


Figure 11-1 Comparisons of compound AUCs and alert AUCs between IG_input and IG_hidden on the validation set. The model instance used was *2layer_model*. The IG_hidden method Variation 2 with weights (see Chapter 10) was used and applied to the neurons of the first hidden layer.

Similar as seen in Chapter 10 for *dropout_model*, IG_input outperformed IG_hidden when evaluating individual compounds. However, when considering alert AUC scores both methods achieved comparable performance and IG_hidden had a slightly higher proportion of alerts with an average AUC of at least 0.8 (0.735 vs 0.714). As observed in Chapter 10, each approach was superior for a different set of compounds and alerts so that the approaches can be considered as complementary.

Overall, the observed AUC scores for IG_hidden were comparable to those recorded for *dropout_model*. This means that the proposed method may be applied successfully on the first hidden layer of a DNN. Notably, no further modifications were made to the approach and this provides evidence for the robustness of the approach on different model instances.

11.3.3 Exploration of neurons in the second hidden layer

A pairwise correlation analysis was conducted (as for neural networks consisting of a single hidden layer in Chapter 8) in order to understand the relations between different hidden neurons. All pairwise neuron correlations are shown in a heatmap in Figure 11-2A. The pairwise correlations recorded between neuron pairs within the first hidden layer, neuron pairs between the first and second hidden layer, and neuron pairs within the second hidden layer are visualised as histograms in Figure 11-2B.

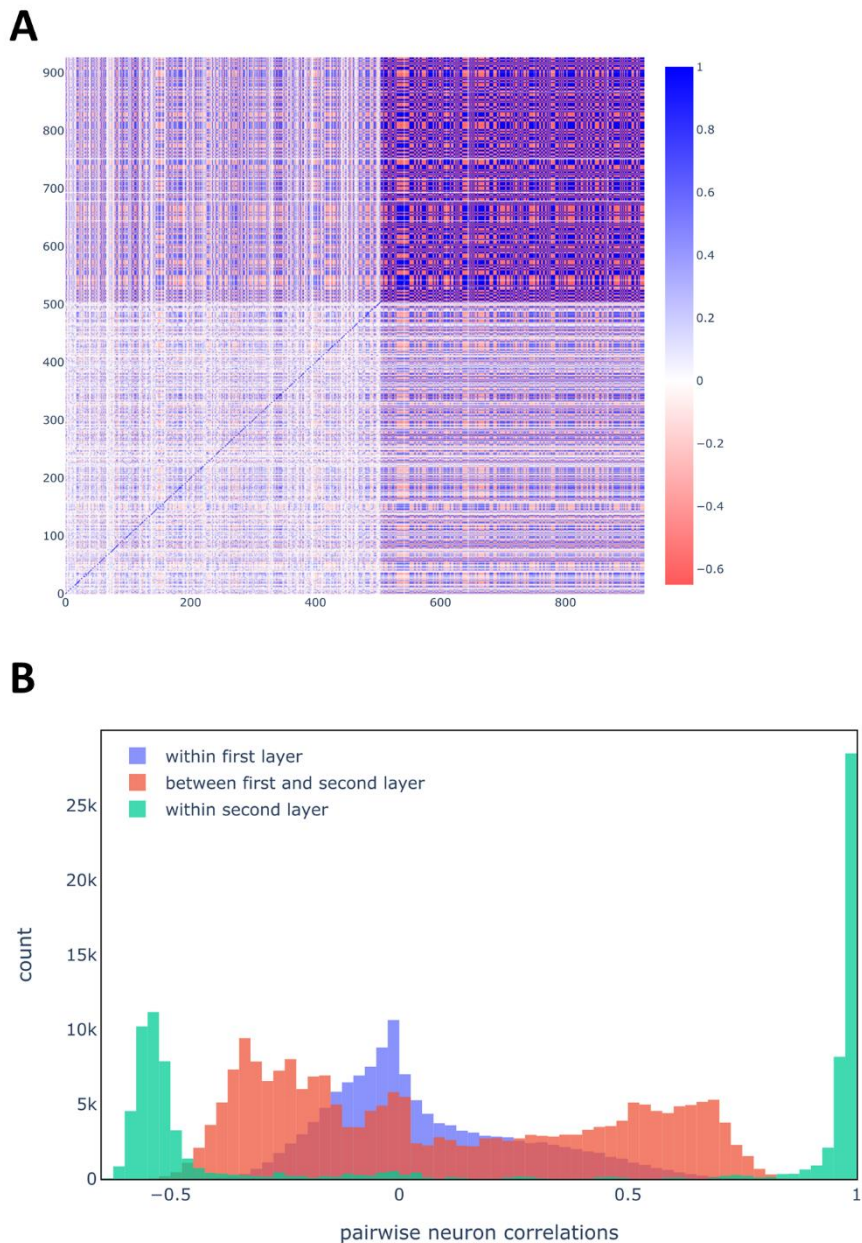


Figure 11-2 Correlation analysis of hidden neurons. **A:** pairwise correlations were calculated between all hidden neurons. Neurons with a maximum activation of <0.01 were excluded. Neurons in the 1st hidden layer: 0-505 (506 of the 512), Neurons in the 2nd hidden layer: 506-926 (421 of the 512). **B:** Pairwise correlations are grouped into pairs of neurons in the first layer ‘within first layer’, pairs of neurons in the first and second layer (‘between first and second layer’) and pairs of neurons in the second layer (‘within second layer’). The histogram of pairwise correlations for each group is shown. Correlations of neurons with themselves were ignored.

In the heatmap, the neurons 0-505 belong to the first hidden layer, while the neurons 506-926 belong to the second hidden layer. Although not labelled in the heatmap, the boundary between neurons in the first and second hidden layer is clearly visible, due to the largely different correlations occurring between the different groups of neuron pairs (as shown in Figure 11-2B). Neuron pairs within the first hidden layer have mostly no or very little correlation. A correlation of above +0.5 was found (blue

histogram) for a small number of those pairs only. In contrast, the vast majority of neuron pairs within the second hidden layer either have a strong positive correlation ($>+0.9$) or a moderately strong negative correlation (<-0.5). This means that many of the neurons in the second layer detect the same chemical features. Neuron pairs between the first and second hidden layer possess correlations across a wide range of values (from -0.5 to $+0.8$). A high correlation between a neuron in the second layer and a neuron in the first layer suggests that the neuron in the second layer detects similar features as those detected in the neuron in the first layer.

In the following an attempt was made to identify neurons in the second layer which detect novel chemical features (i.e. not already detected in the first layer). For all neurons in the second layer (excluding the ones with a maximum activation <0.01) the maximal correlation with any neuron in the first layer was determined as well as the weight assigned to the neuron in the output neuron (see Figure 11-3).

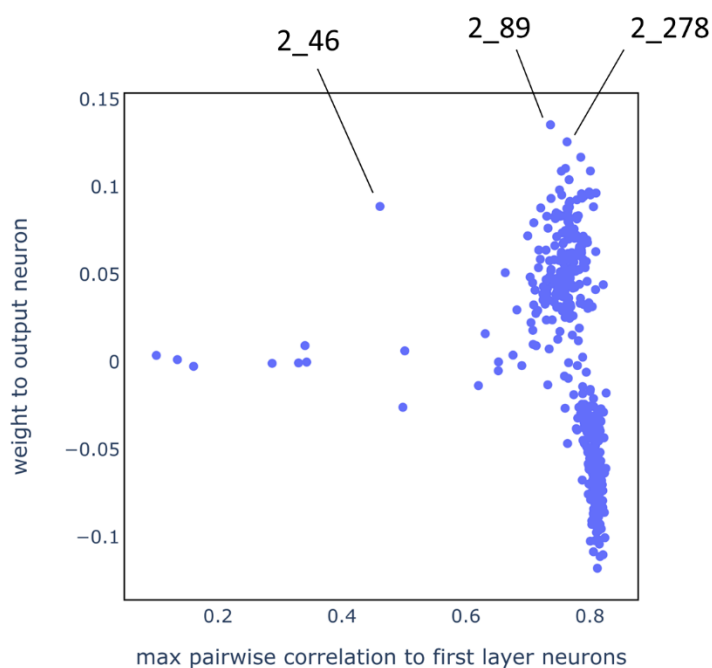


Figure 11-3 Search for 2nd layer neurons detecting novel chemical features. x-axis: maximal correlation observed with any of the neurons in the first hidden layer. The neurons selected for further analysis were labelled.

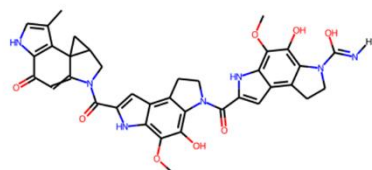
It can be seen that the majority of neurons in the second hidden layer had a fairly high correlation ($>+0.7$) with at least one neuron in the first hidden layer. This suggests that these neurons do not detect novel chemical features (at least not primarily) as their activations were similar to those of one or more neurons in the first hidden layer. Positive and negative weights of varying magnitude were

observed for the strongly correlated neurons. A few second-layer neurons were not strongly correlated to neurons in the first layer (max correlation $<+0.5$), but most of them had a weight to the output neuron of close to zero which means that they had very little relevance for the prediction made by the network. An exception was neuron 2_46 which had a relatively high weight to the output neuron (+0.089) and at the same time was not strongly correlated to any neurons in the first layer (max correlation +0.462). Hence this neuron was selected for further analysis as well as the two neurons with the highest weight to the output neuron (2_89 and 2_278).

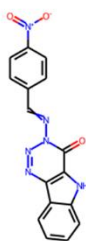
To find out which chemical features were detected in the respective neurons, it was analysed whether compounds matching a particular Derek alert on average activate the neuron more strongly than compounds matching no Derek alerts. The Derek alerts with the highest ratio between mean activation of matching compounds and mean activation of non-matching compounds are reported in Table 11-5 for each of the three neurons. Furthermore, the Top-3 training compounds most strongly activating each neuron are shown in Figure 11-4.

Table 11-5 Most relevant features detected in neurons. Shown are the Top-5 Derek alerts with the highest mean activation of the neuron (reported as the ratio to the mean activation of compounds matching no Derek alert). Shown is also the number of alerts with such a ratio of at least 5 and for reference the mean activation of compounds matching no Derek alert.

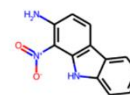
	Neuron 2_46	Neuron 2_89	Neuron 2_278
Rank 1	90: triazin tpye (4187)	31: N-nitro or N-nitroso (221)	31: N-nitro or N-nitroso (258)
Rank 2	22: cyclopropane type (2553)	44: azide (208)	44: azide (253)
Rank 3	84: quinone (2491)	87: dihalide type (206)	87: dihalide type (248)
Rank 4	46: pyran type (2152)	102: Aromatic azoxy (182)	102: Aromatic azoxy (243)
Rank 5	45: ketone or aldehyde type (2046)	78: aromatic N-oxide (172)	39: nitrogen or sulphur mustard (208)
Number of alerts with ratio >5	65/102	96/102	97/102
Mean activation of compounds matching no alert	8.5×10^{-5}	7.4×10^{-3}	6.2×10^{-3}

2_46

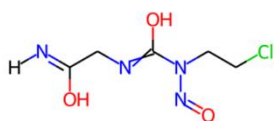
1666: 0.718



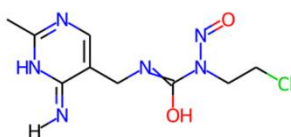
519: 0.579



2655: 0.543

2_89

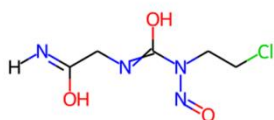
1712: 4.52



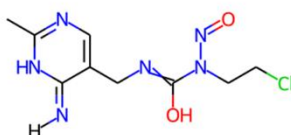
2804: 4.39



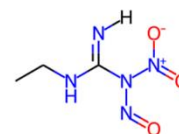
2664: 4.28

2_278

1712: 4.51



2804: 4.16



2664: 4.07

Figure 11-4 Top-3 compounds for neurons 2_46, 2_89, 2_278. Below each compound its index and neuron activation are reported.

Some of the Top-5 alerts for neuron 2_46 (Table 11-4) can be observed in the Top-3 compounds. Compound 1666 matches Alert22 (cyclopropane type) and compound 519 matches Alert90 (triazin type). A large number of alerts (65) has a mean activation of at least 5 times larger than the mean activation for compounds matching no Derek alert. This suggests that the neuron detected a large number of different chemical features. It was further analysed whether the features learned by this neuron provide *2layer_model* with a predictive advantage over *best_model* and *dropout_model*. To that end, the 30 validation compounds most strongly activating neuron 2_46 (all toxic) were inspected. All of them were predicted as toxic by *2layer_model*, yet the same is true for both *best_model* and

dropout_model. Hence it does not seem that neuron 2_46 has learned chemical features that cannot also be learned in the first layer of a model.

Neuron 2_89 and 2_246 seem to have learned very similar chemical features. The Top-3 compounds are identical and four of the Top-5 alerts are identical. In fact, the two neurons possess a very high pairwise correlation (0.987). Also note that in both neurons the mean activation for almost all alerts is at least five times larger than the mean activation for compounds matching no Derek alerts. This suggests that both neurons detect almost all relevant chemical features to predict compounds as toxic (i.e. matching any Derek alert). Next, it was therefore investigated how well individual neurons in the first and second hidden layer can classify toxic compounds (see Methodology).

For each neuron in the first and second hidden layer of *2layer_model* a ROC-AUC score was computed for the training and the validation set. 'Toxic' was considered as the true label and neurons that rank toxic compounds higher than non-toxic ones possess a score above 0.5. A score of 1 would mean that all toxic compounds are ranked higher than non-toxic ones. Conversely, if a neuron ranks all non-toxic compounds higher than toxic ones, a score of 0 would be achieved. Both a score of 0 and 1 hence would indicate perfect discrimination of toxic and non-toxic compounds. The distribution of AUC scores found for individual neurons in the first and second layer is shown for the training set in Figure 5A and for the validation set in Figure 11-5B.

It can be seen that the vast majority of neurons in the second layer were capable of distinguishing between toxic and non-toxic training compounds (i.e. stronger activation for one of the classes) (Figure 11-5A). Some of these neurons ranked toxic compounds higher (AUC>0.95), while others ranked non-toxic compounds higher (AUC<0.05). In contrast, an AUC of above 0.85 or below 0.15 was found for very few neurons in the first hidden layer. A moderately high AUC (0.6-0.85) would suggest that a neuron ranks some toxic compounds higher than others, but not all toxic ones higher than non-toxic ones. This is consistent with the finding that neurons in the first layer detect some of the relevant chemical features, but different neurons detect different features. Notably, neurons in *dropout_model* (one layer) neither possessed extreme AUC values (<0.1 or >0.9) (see Appendix Figure D1). Generally, the same observation was made for the validation compounds (Figure 11-5B), although as expected the neurons tended to discriminate slightly less well between toxic and non-toxic compounds (no AUC <0.025 or >0.975). It is worth mentioning that some of the neurons in the second layer of *2layer_model* discriminate between toxic and non-toxic with comparable performance to the model itself (AUC on training set: 0.999, AUC on validation set: 0.977).

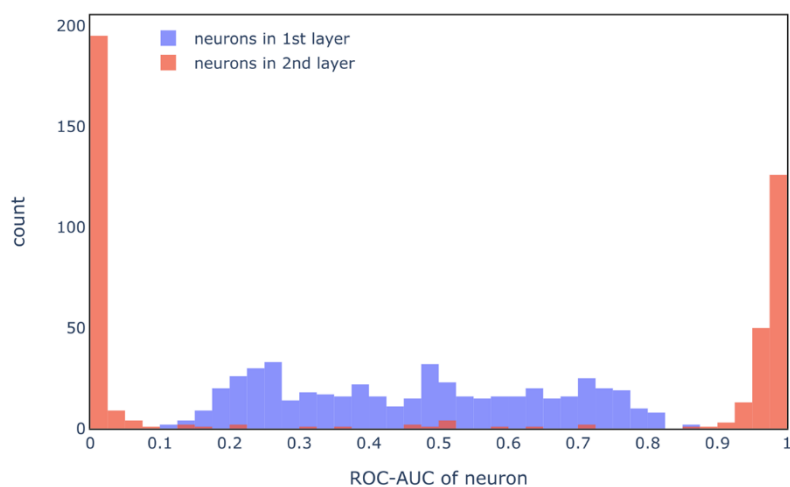
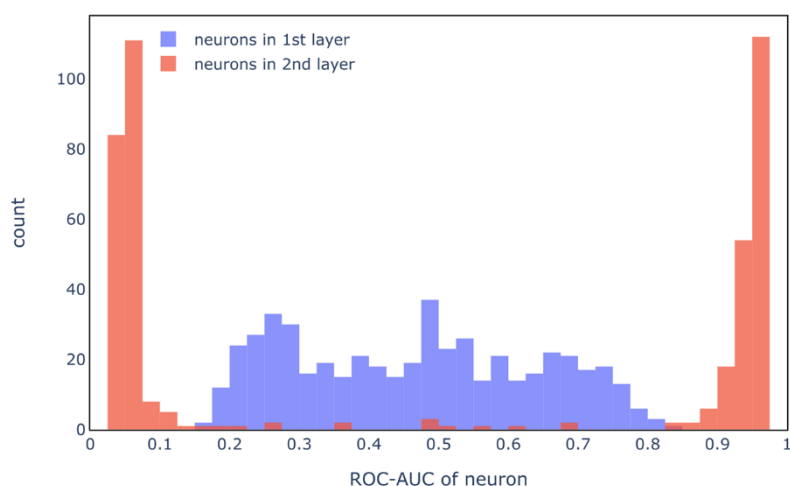
A: training set**B: validation set**

Figure 11-5 ROC_AUC scores for individual neurons. To compute the ROC-AUC scores, the neuron activations for neurons were considered as predictions. **A:** training set, **B:** validation set.

The AUC scores for the neurons analysed above are reported in Table 11-5. The scores are consistent with the findings in Table 11-4. The neurons 2_89 and 2_278 ranked nearly all toxic compounds higher than non-toxic ones and hence they possess AUC scores relatively close to 1. In contrast, neuron 2_46 seemed to detect only some of the chemical features relevant for toxicity which corresponds to a moderate AUC score.

Table 11-6 ROC_AUC scores for exemplary neurons. Reported are the scores for the same neurons as in Table 11-4.

Neuron ID	Training set	Validation set
2_46	0.643	0.606
2_89	0.965	0.934
2_278	0.977	0.977

It was reported above (Figure 11-2) that pairs of neurons in the second layer either had a very strong positive correlation ($> +0.95$) or a moderate negative correlation (~ -0.5). An obvious explanation for this finding is that those neurons with AUC scores close to 1 were strongly positively correlated and similarly for those with AUC scores close to 0, while pairs between the two groups were negatively correlated. To check this, the neurons with most extreme AUC values on the training set were considered ($AUC < 0.05$ or > 0.95) and all pairwise correlations within the respective group of neurons and between the groups were determined and visualised using histograms (Figure 11-6).

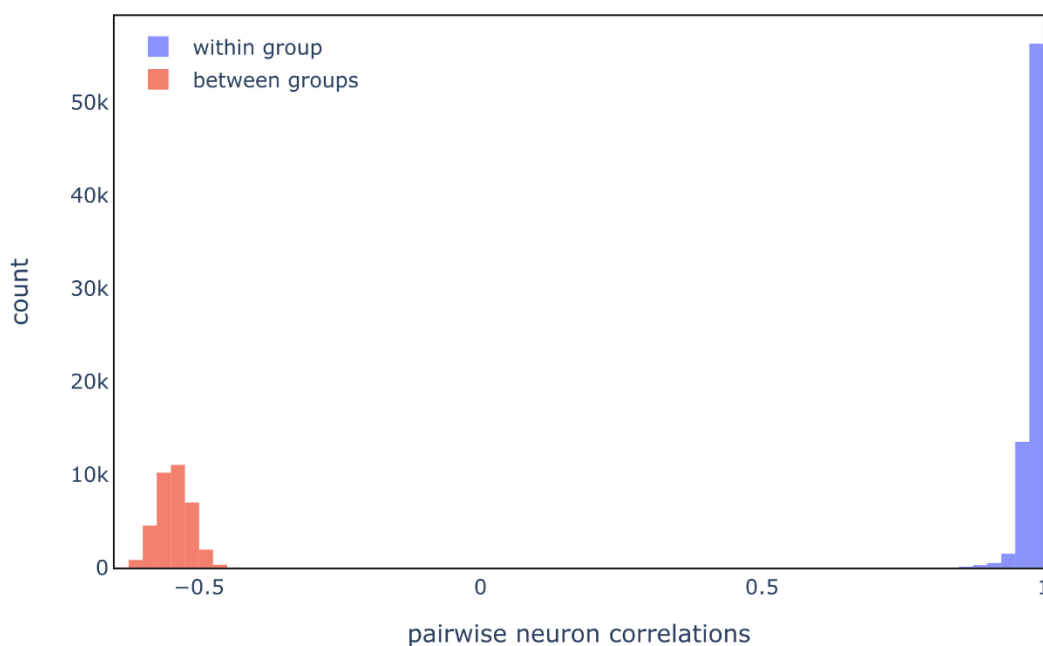


Figure 11-6 Pairwise correlations of selected neurons in Derek model. Among neurons in the second layer of *2layer_model*, those with an AUC of below 0.05 or above 0.95 on the training set were selected. Shown are pairwise correlations for neuron pairs within each group (group1: $AUC < 0.05$, group2: $AUC > 0.95$) and between the groups.

It can be seen that all pairs of neurons within a group ($AUC > 0.95$ or $AUC < 0.05$) are strongly positively correlated (vast majority $> +0.95$). This means that neurons within these groups are strongly activated by the same compounds and hence detect the same chemical features. For instance, the training compound activations for neuron 2_89 and 2_278 (both $AUC > 0.95$ for training compounds) are shown in Figure 11-7A. The strong positive correlation is clearly visible. In contrast, neuron pairs between the two groups were negatively correlated which means that different compounds activated the

respective neurons strongly. Also for this case an example is shown using the neurons 2_89 (AUC > 0.95) and neuron 2_32 (AUC < 0.05) (Figure 11-7B). It can be seen that compounds strongly activating neuron 2_89 have an activation close to zero for neuron 2_32 and vice versa.

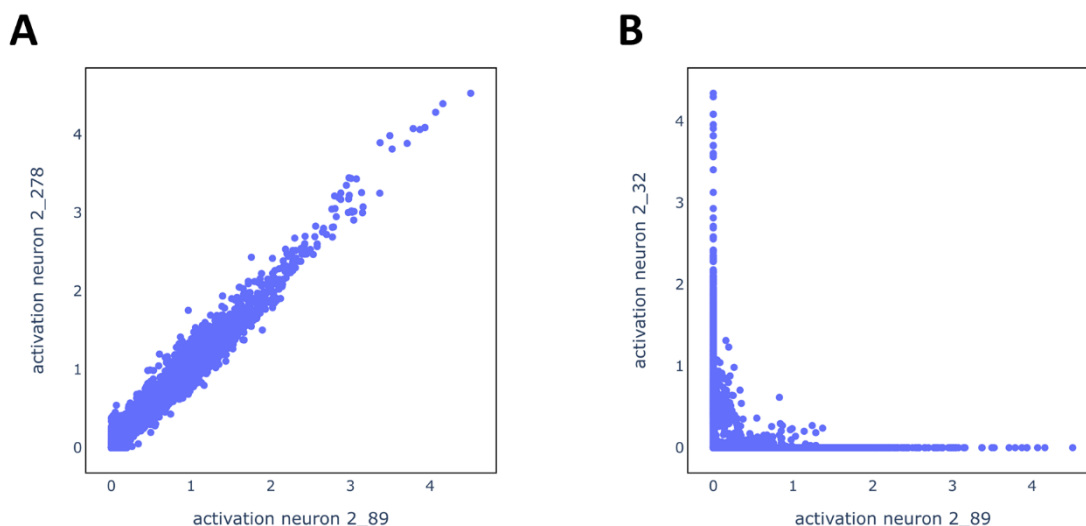


Figure 11-7 Training compound activations for neuron pairs. A: neurons 2_89 and 2_278. **B:** neurons 2_89 and 2_32.

It can be concluded that the majority of neurons in the second hidden layer did not recognise specific chemical features and rather were activated by virtually all toxic (or non-toxic) compounds. The assumption behind the approach developed to explain model predictions (IG_hidden) is that there is some diversity in which chemical features are detected in different neurons in a hidden layer. Here, it was demonstrated that this is not the case for neurons in the second hidden layer of *2layer_model* and hence no attempt was made to extract substructures for those neurons. Instead, different toxicity and bioactivity datasets were studied next to determine whether these findings are specific to the model obtained for the Derek dataset and are more generally valid.

11.3.4 Evaluation and exploration of additional toxicity and bioactivity datasets

Various toxicity and bioactivity datasets were considered to test whether the observations made for *2layers_model* with respect to the features detected in the second hidden layer also occur in other models. As described in Methodology, a grid search testing different hyperparameters was conducted

for each assay. The best score observed on the validation set for a model with 1 layer and a model with 2 layers for each dataset are reported in Figure 11-8.

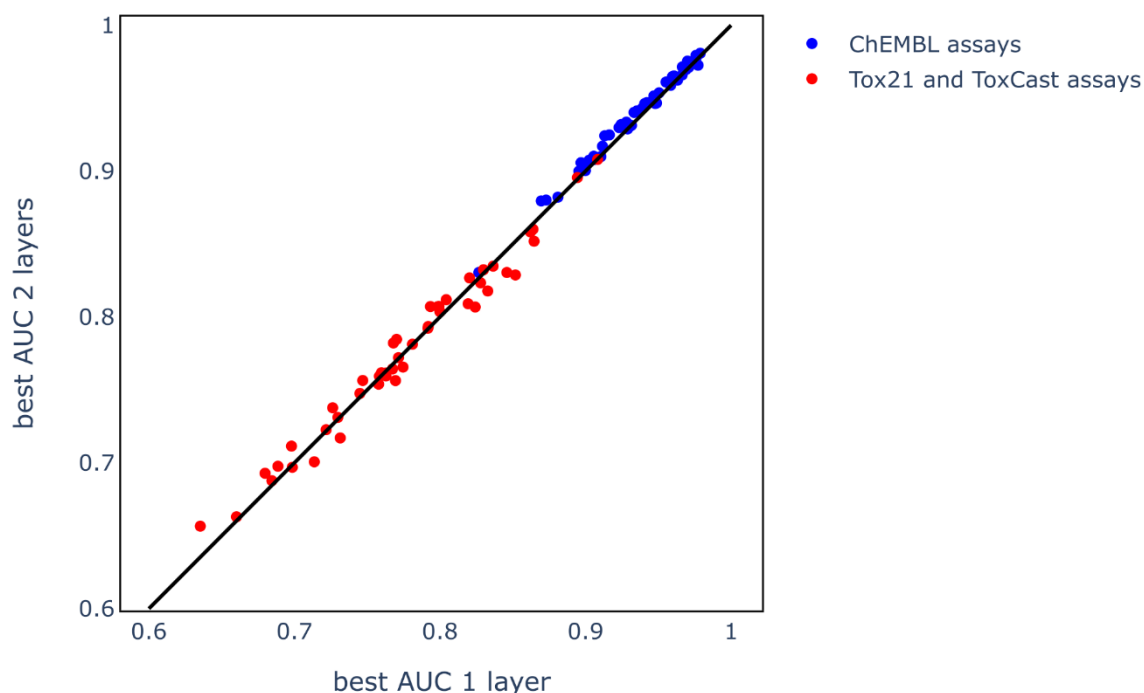


Figure 11-8 Comparison of best 1-layer and 2-layer model for Tox21, ToxCast and ChEMBL datasets. Shown are AUC scores of the best performing model with 1 and 2 layers on the validation set.

It is worth noting that AUC scores generally were higher for models trained on the ChEMBL assays when compared to the Tox21 and ToxCast assays. This might be due to there being a higher proportion of actives in the ChEMBL assays or due to the ChEMBL data containing series of chemically similar analogues. But this was not further investigated. For the Tox21 and ToxCast assays, the model with two layers performed better in some cases, but for approximately an equal number of assays models with one layer performed best. In contrast, for most ChEMBL assays, models with 2 layers performed better. However, in all cases the differences in AUC score between the best model with one layer and two layers were quite small.

In the following, the 2-layer models for some of the assays were further analysed. Assays for which the best 2-layer model outperformed the best 1-layer model were selected, as these would potentially be assays where the 2-layer model may have learned chemical features missed by the respective 1-layer model. The assays are reported in Table 11-7 along with the AUC scores for the best 1-layer and 2-layer models. A complete list of hyperparameters for the models is reported in Table D1 (Appendix D).

Chapter 11: Explaining predictions for deep neural networks

Table 11-7 Selected assays for further analysis. Selected were two assays from the ToxCast dataset and two assays from the ChEMBL dataset. Shown are AUC scores for the best performing 1-layer and 2-layer model.

Assay	Best AUC 1 layer	Best AUC 2 layers
ATG_ERa_TRANS_up	0.793	0.807
ATG_RXRb_TRANS_up	0.768	0.782
Adenosine A1 receptor (CHEMBL226)	0.913	0.924
Peroxisome proliferator activated receptor alpha (CHEMBL239)	0.870	0.880

For each 2-layer model, it was tested whether the neurons in the second layer detect specific chemical features not detected in the first hidden layer. To do that, pairwise correlations between neurons as well as AUC scores for individual neurons were determined (see above). These are reported for the individual assays in Figures 11-9 through 11-12.

It can be seen that pairs of second layer neurons of the model for ATG_ERa_TRANS_up were either strongly positively correlated or had a negative correlation of moderate strength (Figure 11-9A). The neurons in the second layer had AUC scores close to 0 or 1 on the training data (Figure 11-9B). These observations are very similar to those made for *2layer_model* (Derek data, see above). The same is true for the assays CHEMBL226 (Figure 11-11) and CHEMBL239 (Figure 11-12). This suggests that the neurons in the second layer for these models also detected toxic or bioactive compounds rather than specific chemical features. Different observations were made for the model on ATG_RXRb_TRANS_up. In this case, none of the neuron pairs (within or between layers) were strongly correlated (Figure 11-10A). Moreover, the neurons in the second hidden layer were not found to have AUC values close to 0 or 1. Instead the AUC values for those neurons ranged between 0.3 and 0.65, with most being very close to 0.5.

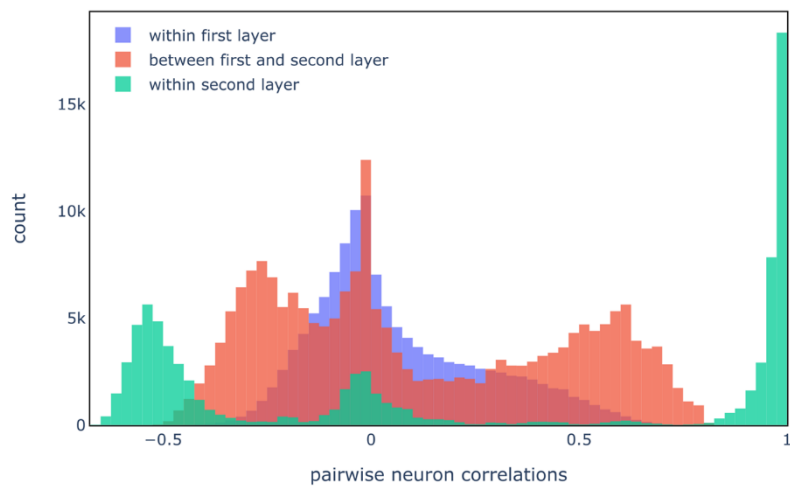
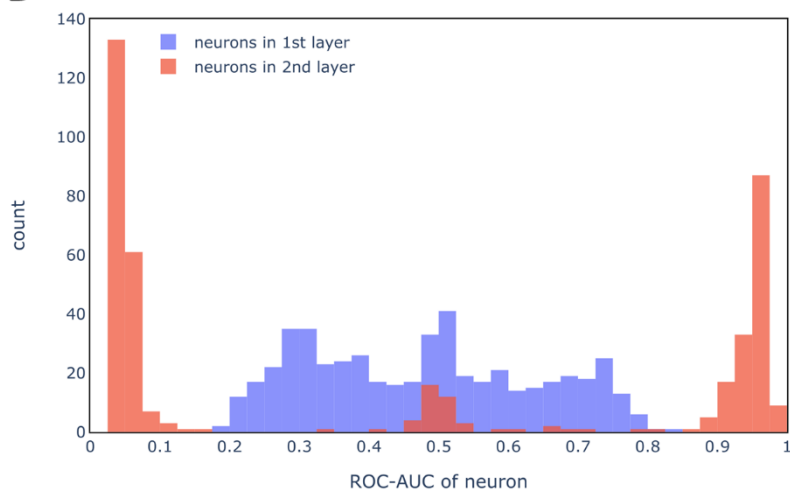
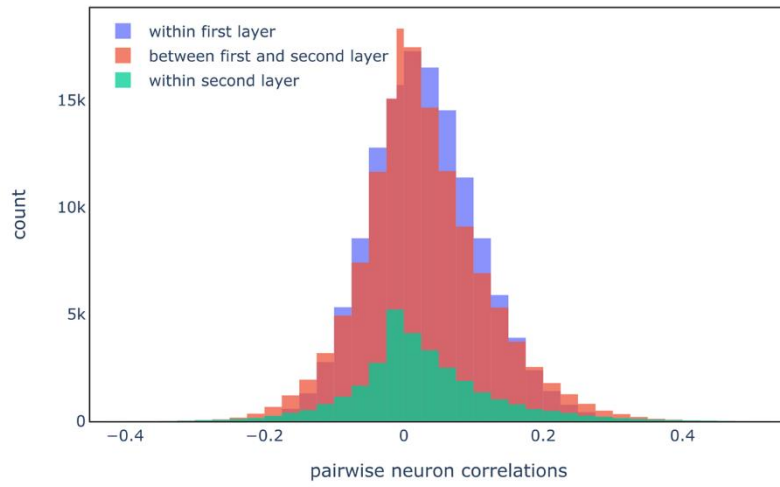
A**B**

Figure 11-9 Analysis of second layer neurons: ATG_ERa_TRANS_up. **A:** pairwise correlations of neurons (using training compounds). For details see caption of Figure 11-2. **B:** AUC scores of neurons on training compounds. For details see caption of Figure 11-5.

A



B

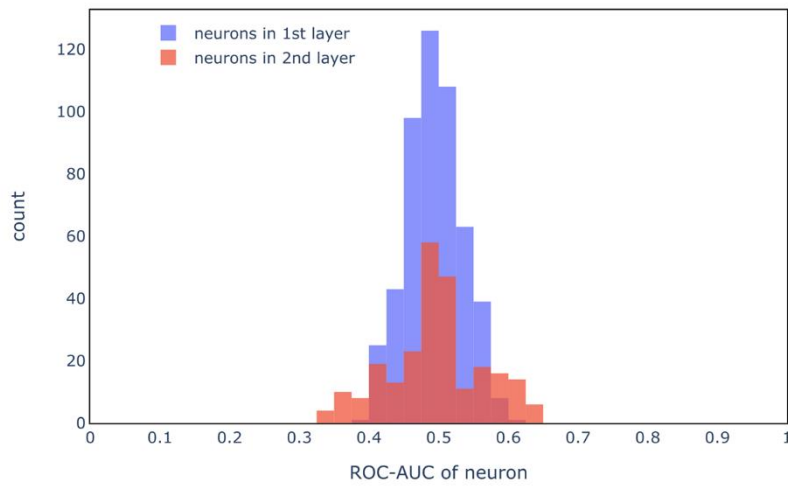
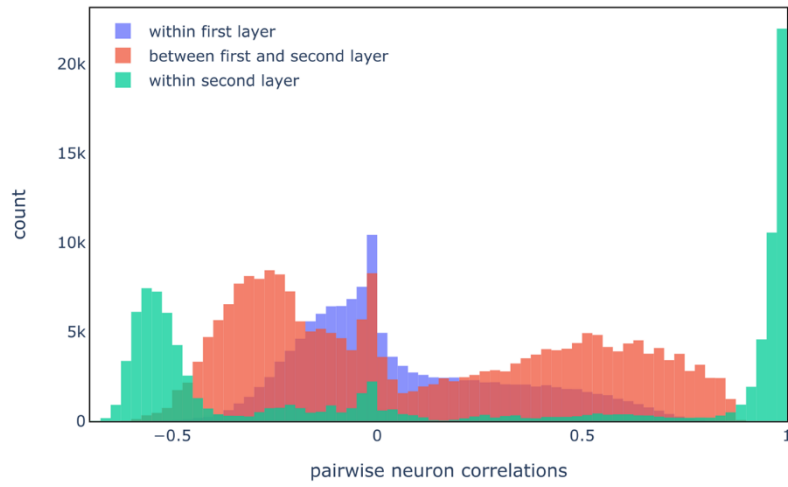


Figure 11-10 Analysis of second layer neurons: ATG_RXRb_TRANS_up. For details see caption of Figure 11-9.

A



B

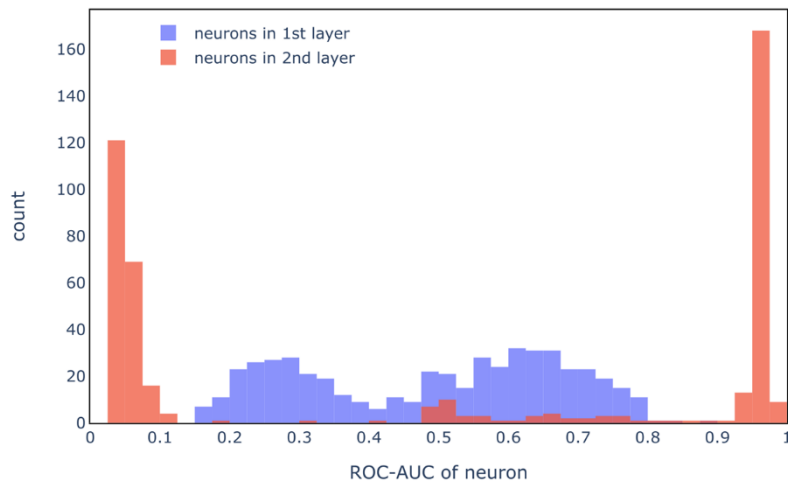


Figure 11-11 Analysis of second layer neurons: Adenosine A1 receptor. For details see caption of Figure 11-9.

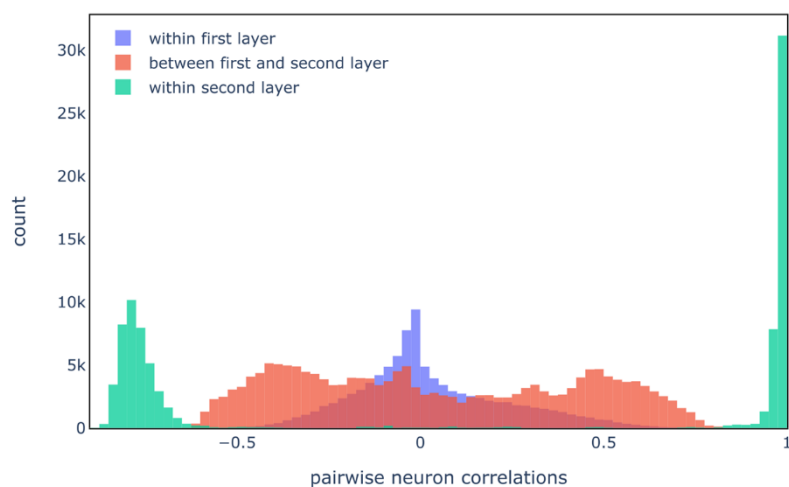
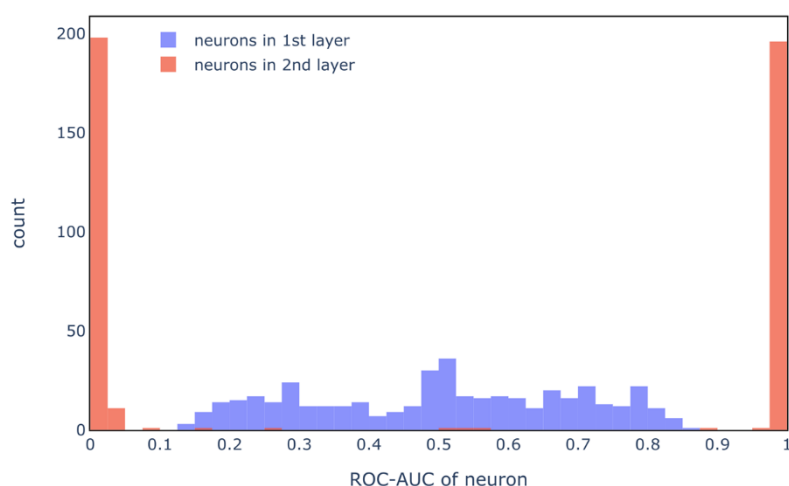
A**B**

Figure 11-12 Analysis of second layer neurons: Peroxisome proliferator activated receptor alpha. For details see caption of Figure 11-9.

The model for the ATG_RXRb_TRANS_up assay was further analysed in order to understand its behaviour. First, the predicted probabilities of the model on the training and validation data were inspected (see Figure 11-13A). It can be seen that, while the model achieved a relatively high AUC score, most predicted probabilities were close to 0.5. Notably, binary cross-entropy was used as the loss function to fit the model. Binary cross-entropy measures the distance of predicted probabilities of training data to the true labels (0 or 1) and hence the model did not fit the training data very well (in terms of binary cross-entropy). It was therefore suspected that the resulting model had been stopped early (the epoch was not recorded during the grid search). To further analyse this, the model

instance obtained in the grid search was trained for a further 10 epochs with identical hyperparameters. After each epoch of training, the AUC on training data, the AUC on validation data and the loss (average binary cross entropy across all batches of an epoch) were recorded (Figure 11-13B).

It can be seen that the average loss of the initial model was very high (data point at training epochs=0) and decreased steadily over the following epochs. Interestingly, the AUC score on both training and validation set dropped strongly after the first epoch of training, although the loss decreased. It seems that the model achieved a very good ranking of compounds (high AUC), while predicted probabilities were still very close to 0.5 (high binary cross entropy loss). In the following training epoch, AUC scores on both training and validation set increased. The AUC on training reached its highest value after 10 epochs when also the loss was lowest. The AUC on validation data never exceeded the initial AUC, while the model instance after seven epochs reached the highest value among those subjected to further training (~0.75). This analysis seems to confirm that the initial model instance resulted from a model stopped very early during training.

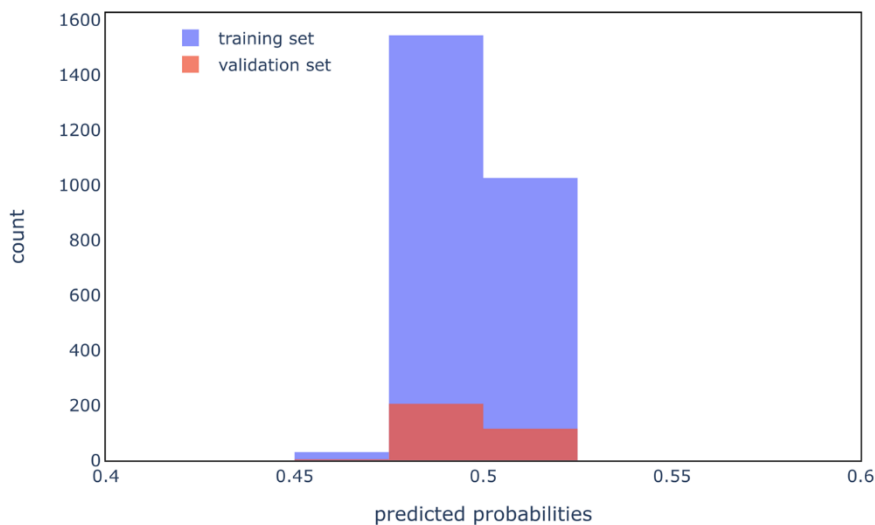
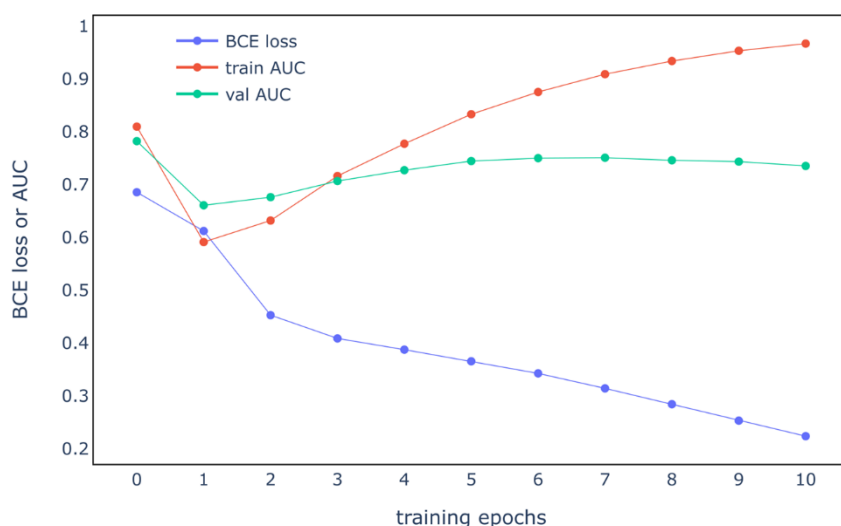
A**B**

Figure 11-13 Follow-up analyses on 2-layer model trained on ATG_RXRb_TRANS_up. **A:** Histogram showing predicted probabilities of model on training and validation set. **B:** The model instance saved during the grid search was trained for 10 further epochs. Shown in the line plot are AUC scores on the training set (of initial model and after each epoch), AUC scores on the validation set (of initial model and after each epoch), mean binary cross-entropy (BCE) loss of the initial model and in each epoch. The value at training epochs=0 is the average BCE loss of the initial on batches of training data (while no training is conducted). The value at training epochs=1 is the average BCE loss across all batches in the first epoch.

To better understand the effect of the further training epochs on the model, the model instance after 10 additional training epochs was analysed. Most predicted probabilities fall in the range 0-0.8 (see Figure D2, Appendix D), with a majority of predictions close to zero (non-toxic is the majority class). Pairwise neuron correlations and AUC scores of individual neurons are shown in Figure 11-14. It can be seen that the model shows similar characteristics as the models obtained for the other assays

(pairwise correlations between neurons in the second layer mostly close to +1 or below -0.5; neurons in second layer with AUC values relatively close to 0 or 1). The deviant behaviour of the model obtained in the grid search hence seems to be due to early stopping and not due to the assay.

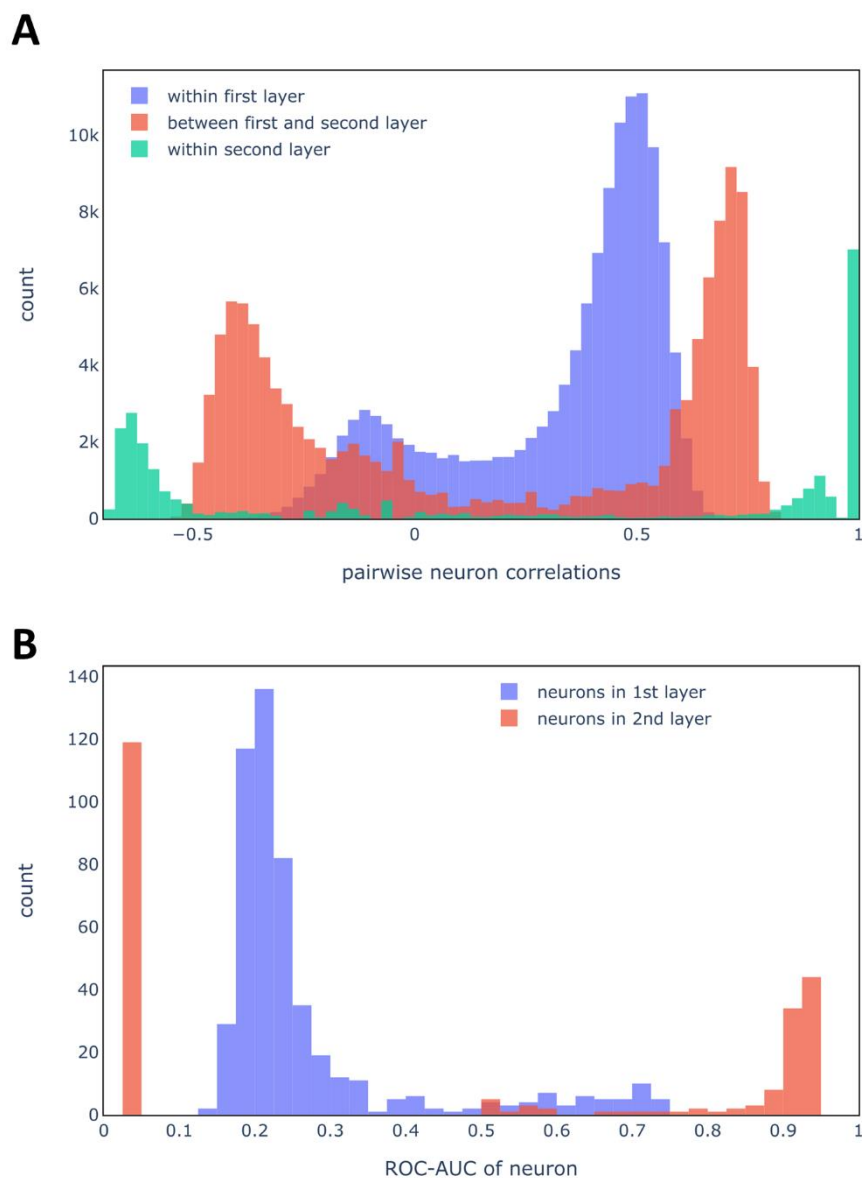


Figure 11-14 Analysis of second layer neurons after 10 training epochs on initial model on ATG_RXRb_TRANS_up. A: pairwise correlations between neurons (for details see caption of Figure 11-2). **B:** AUC scores of individual neurons on the training set (for details see caption of Figure 11-5).

In conclusion, all the analysed two-layer models showed characteristics similar to the *2layer_model* (Derek model), as long as they were not stopped very early during training. In all cases neurons in the first hidden layer seemed to detect specific chemical features (although it may be a number of different features), while neurons in the second hidden layer combined those features such that

nearly all those relevant for one the classes to be predicted were detected in individual neurons. The interpretation method developed in this thesis is based on the assumption that hidden neurons detect specific chemical features. This was not the case for the neurons in the second hidden layer of the analysed neural networks and hence the developed approach was deemed to be not applicable to those neurons.

11.4 Discussion

The aim of this chapter was to adapt the developed approach to interpret DNN models. It was shown that the approach can be applied to neurons in the first hidden layer of a DNN consisting of two layers and this resulted in overall good explanations of model predictions. Next, the neurons in the second hidden layer of DNNs were investigated. It was found that those neurons did not detect specific chemical features but instead they seemed to mostly group different chemical features associated with toxic and non-toxic predictions together. That is, the neurons could be divided into two ‘clusters’ of very similar neurons, A key assumption of the method for extracting substructures is that specific chemical features can be identified in individual neurons and hence the method was deemed inappropriate for application on the second layers of the DNNs under study.

It may still be the case that chemical features detected in the first hidden layer are refined to some extent which could make the second hidden layer useful to the network in terms of its predictive performance. For most of the datasets studies here the two layer DNNs provided a benefit in performances compared to networks consisting of a single layer, albeit the benefit was small in all cases. However, it seems that applying the IG_hidden method to neurons in the first hidden layer is sufficient to cover most chemical knowledge learned by the DNN.

In previous studies it was reported that hidden neurons of DNNs may detect pharmacophores or toxicophores and that neurons in deeper layers tend to detect larger chemical structures (Mayr et al., 2016; Preuer et al., 2019). However, the model instances they trained were not available for further examination. A major difference between the DNNs in the cited studies and the present study is that they used multi-task DNNs. In multi-task DNNs predictions for different tasks need to be made which means that the model needs to discover which chemical features are relevant for a certain task. Thus the last hidden layer of a multi-task DNN must detect chemical features specific to individual tasks or relevant to a set of modelled tasks (as some chemical features may be associated with different toxicity mechanisms). For this reason, it may be that the IG_hidden method can be better applied to hidden neurons of multi-task DNNs rather than single task DNNs, but this was beyond the scope of

the present study. It may also be the case that the choice of model architecture (number of layers, number of neurons, type of activation function, use of dropout etc.) and training parameters (loss function, optimization scheme etc.) impact what features individual neurons learn and hence how well suited the developed approach is to explain the model predictions.

The approach developed to extract substructures that activate a hidden neuron uses information about weights linking neurons to input features, which is only available for neurons in the first hidden layer. Therefore, the approach would need to be modified in order to be applicable to neurons in deep layers. Some suggestions on how that might be achieved are provided in Chapter 12.

11.5 Conclusion

The suitability of the IG_hidden method to explain predictions made by DNNs was investigated. The workflow developed to extract substructures was based on the assumption that individual neurons detect specific chemical features and that these can be found by considering the training compounds that most strongly activate the neuron. This was not the case for the neurons in the second hidden layer of the two layer DNNs and hence no attempts were made to extract substructures from the second layer. Nonetheless, it was shown that IG_hidden can be applied to neurons in the first hidden layer and, at least for the DNNs under study here, this seems to be sufficient to capture most of the chemical features learned by the model in order to explain the predictions.

Chapter 12 Conclusions and future work

This thesis investigated various aspects regarding the use of QSAR models for toxicity predictions of chemicals. In Chapters 5 and 6 the performance of multi-task and imputation models was compared to that of single task models. Moreover, attempts were made to rationalise observed differences in performance. In Chapters 8, 9, 10 and 11, a novel method to interpret neural network models was developed and analysed. In particular, the approach extracts substructures responsible for the activation of hidden neurons and uses this information to explain predictions made by the model.

In this chapter, first the results of the thesis are summarised. Then, limitations are described and suggestions for future work are made.

12.1 Summary

12.1.1 Multi-task and imputation modelling for toxicity prediction

In Chapter 5, various multi-task and imputation modelling approaches (multi-task DNN, FN, Macau) were tested on two in vitro toxicity datasets (Ames and Tox21). For traditional multi-task modelling (when no auxiliary assay data for test compounds is available), very little differences in performance between single task and multi-task models were found. In contrast, multi-task imputation models (when auxiliary assay data for test compounds is available) clearly outperformed single task imputation models. Attempts were then made to explain the observed differences between single task and multi-task imputation models. It was found that multi-task imputation models provide the largest benefits for compounds that are dissimilar to training compounds (which is when single task models tend to struggle) and for compounds with a large number of auxiliary assay labels available. Nonetheless, very little additional information (a single auxiliary label) may be sufficient to provide a benefit over the single task model. It was found that the MI-entropy ratio between target assay and auxiliary assay is a useful indicator to identify suitable auxiliary assays.

In Chapter 6, the suitability of multi-task imputation models on a larger toxicity dataset (ToxCast) was tested. As in Chapter 5, the multi-task imputation models clearly outperformed single task models. The recently published GHOST technique was tested as a strategy to aid with imbalanced datasets. For some of the assays, the MCC scores could be improved, however, GHOST was not found to be useful for all the assays. The impact of sparsity on model accuracy was then investigated by removing

randomly selected labels from the training set. This led to a drop in performance for both single task and multi-task models with the latter still being clearly superior. Moreover, the MI-entropy ratio was confirmed to be a useful metric to identify the most useful auxiliary assays within the large dataset.

Multi-task imputation techniques have been found to be very successful compared to single task techniques in this thesis. Naturally, these approaches are limited by the availability of experimental data about test compound and hence not applicable to virtual compounds. Yet it was found that even limited experimental data about a test compound can strongly improve predictions made for the compound. Unsurprisingly, the best auxiliary assays are those most closely related to the target assay. On the other hand, if two assays measure the same target effect they may contain very little independent information and the imputation model in this case might not provide any new insights, even though the performance metric for the target assay improved. Nonetheless, even assays not very closely related to the target may lead to meaningful improvements of model performance making imputation techniques suitable for a wide range of applications in toxicity prediction. Imputation models provide a means to combine any known experimental data about a test compound with its chemical structure, in order to improve predictions made for an unknown property of the compound.

12.1.2 Using extracted substructures to explain neural network models

In Chapter 8, a neural network with a single hidden layer trained to predict Ames mutagenicity was used to explore which chemical features are learned in individual neurons by examining both training compounds that strongly activating a neuron as well as learned weights connecting the neuron to input features. It was found that several different chemical features may be detected in a single hidden neuron. In turn, a particular chemical group may cause the activation of a number of different hidden neurons.

In Chapter 9, a method that automatically extracts substructures activating a hidden neuron was developed. The approach uses FCA to find meaningful chemical concepts by making use of (i) training compounds strongly activating the neuron and (ii) fingerprint bits with high weights that are contained in the selected compounds. A limitation of the method is that in its original form it can only be applied to the first hidden layer of a (feedforward) neural network. This is further discussed below.

Chapter 10 introduced a method to use the extracted substructures to explain predictions made by the model. This was done by using IG to determine the importance of neurons to the prediction (IG_hidden). By using a neural network trained on Derek alert labels (where “toxic” labels were

assigned according to clearly defined rules), the model explanations of IG_hidden were evaluated and compared to those found using the published IG_input method which is based on input features. The obtained model explanations were improved by changing parameters in the substructure extraction workflow. IG_hidden was slightly outperformed by IG_input when evaluating on individual compounds, yet both methods achieved comparable scores when considering average scores for individual alerts. Both methods performed better on different sets of compounds and hence may be used complementarily.

In Chapter 11, the applicability of IG_hidden to DNNs was tested. It was shown that the method (with the optimised parameters identified in Chapter 10) can be successfully applied to the first hidden layer of a DNN trained on Derek labels. Next, the role of neurons in the second hidden layer of DNNs trained on several different tasks (Derek labels, Tox21, ToxCast, ChEMBL) was analysed. It was found that the majority of those neurons do not detect specific chemical features and instead they detect all features linked to either toxicity or absence of toxicity. The method was therefore considered to be not applicable to neurons in the second hidden layer. However, it appeared that in the studied networks the most relevant chemical features were extracted in the first hidden layer and hence applying IG_hidden to the first hidden layer could provide good model explanations.

12.2 Limitations and future work

12.2.1 Multi-task and imputation modelling for toxicity prediction

For traditional multi-task models, at best very small improvements were found over the best single task model. For future work this raises the question of whether multi-task approaches are well suited for some datasets but not for others, or if improved multi-task modelling strategies could have led to better results for the datasets used in this thesis.

Ideally, characteristics of datasets where multi-task modelling will be successful can be defined. As described in Chapter 5, a previous study found the presence of structurally similar compounds (to the test set for the target assay) in the training set for an auxiliary assay as relevant for the success (Xu et al., 2017). If structurally similar compounds are present and the assays are correlated, the performance for the target assay may be increased, while the opposite was found if the assays are uncorrelated. In the studied dataset there, very little overlap existed between the assays and more general explanations for a wider range of datasets is required. Moreover, it would be helpful to

determine whether the same reasons determine the success of multi-task techniques other than multi-task DNNs (e.g. Feature Net).

FN models in this study were all trained following the same procedure. This was to include all assays and to use binary assay labels to fill gaps in the first step of the technique (see Methodology in Chapter 5). Feature Net models might be improved if poorly predicted auxiliary tasks are excluded, as these may add more noise than signal to the models in the second step. Moreover, using the raw predicted probabilities for the active class instead of binary labels might lead to better models, as this could provide a more nuanced signal. Further experiments may improve the performance of FN models.

In a very recent study, multi-task DNNs based on graph convolutions clearly outperformed various single task models for the task of *in vivo* brain penetration (auxiliary tasks were *in vitro* assays related to brain penetration) (Hamzic et al., 2022). It remains unclear, whether the modelling technique or the characteristics of the dataset were responsible for the success. In conclusion, more studies are required to disentangle the effects of datasets (especially the overlap and correlation between assays) and modelling techniques on the success of traditional multi-task models.

The datasets used in the experiments were chemically standardised, yet another source for bias was not addressed. Actives in high-throughput screening assays may be false positives due to unspecific effects such as compound aggregation or interfering with the readout instead of the specific biochemical interactions of interest. Such compounds are also referred to as PAINS (Pan-Assay Interference compounds) (Baell & Holloway, 2010; Klarner et al., 2022). In particular, a multi-task model may learn to detect PAINS rather than truly interesting compounds. For future studies, this issue should be addressed by either filtering out suspected PAINS or by conducting thorough experimental validations for actives.

In this work, *in vitro* toxicity tasks were used to test the success of imputation models. These models combine information about chemicals (chemical descriptors) with partially available experimental toxicity profiles of the compounds. In principle, the included modalities may be further extended. For toxicity prediction, *in vivo* effects are ultimately of interest. To predict *in vivo* toxicity, *in vitro* toxicity assay data (i.e. measuring cellular mechanisms related to the toxicity of interest) may be used as additional information, for instance in a FN model. *In vitro* toxicity data has been shown to be useful to predict *in vivo* toxicity in some previous studies (Liu et al., 2017; Thomas et al., 2012). Other recent studies used biological features such as gene expression profiles or morphological cell changes caused by compounds (Cell Painting Assay) to predict bioactivity and toxicity with some success (Moshkov et al., 2022; Seal et al., 2022; Trapotsi et al., 2021). Further studies are required to identify strategies on

how to best combine sparse heterogeneous data to predict toxicity. The multi-task techniques tested in this thesis might prove suitable also for such applications.

12.2.2 Using extracted substructures to explain neural network models

While the developed approach (IG_hidden) has shown some promising results, it possesses certain limitations. Firstly, the approach in its current form can only be applied to the first hidden layer of a neural network. The approach would need to be adapted in order to be applicable to deeper layers of a feedforward neural network or to neurons in a graph-convolutional neural network. Several suggestions are made in the following paragraphs.

The approach for substructure extraction in the first hidden layer combines information about training compounds that strongly activate the neuron with FP bits having high learned weights. Neurons in the second hidden layer are not directly connected to the input layer. Instead, they are connected to neurons in the first hidden layer. Once the current approach has been applied to the first hidden layer, those extracted substructures may be used instead of fingerprint bits to inform the substructure extraction for the second hidden layer. In particular, for a given neuron in the second hidden layer, the neurons in the preceding layer and the respective substructures may be used together with strongly activated training compounds to conduct the FCA and subsequent steps. One problem may be an increased computational cost, as a large number of substructures may have been extracted for a given neuron, while just one chemical environment is attached to an input bit (unless bit collisions occur). However, it may be that it is sufficient to consider a low number of neurons from the preceding layer to successfully adapt this approach.

Another idea is to determine how important an input feature is to a deep neuron using IG, although they are not directly connected with a weight in the network. In this work, IG has been used to determine the importance of an input feature to the model prediction (IG_input) and to determine the importance of a hidden neuron to the model prediction (IG_hidden). Similarly, the technique may be applied to determine the importance of an input feature to the activation of a deep neuron (all three problems are mathematically equivalent). Notably, IG would yield the most relevant input bits for the activation of an individual compound. In this way, substructures could be directly extracted from the training compounds (without conducting FCA). Or instead, for a given neuron the set of relevant input features could be determined across different training compounds (for instance, by averaging) to obtain a set of most relevant input features to be used in the FCA.

The two ideas proposed so far may be applicable to DNNs based on input features (such as Morgan FP), but not for other architectures like GCNs. GCNs perform graph convolutions on hidden representations of individual atoms (or bonds) to aggregate information from their vicinity, before obtaining a d-dimensional representation of a complete compound. In this representation (as shown in this work for hidden layers of a feedforward neural network), one dimension may encode the presence of substructure(s) relevant for the predicted task. The ideas described below are also applicable to such architectures.

IG has been applied to GCNs to identify atoms of a compound most relevant to the model prediction (Jiménez-Luna et al., 2021). Likewise, it may be used to determine which atoms are most relevant for the activation of neurons in the fully connected layers. This could be applied to training compounds that strongly activate the respective hidden neuron. Connected atoms in those training compounds could then be extracted as relevant substructures for the neuron. In a similar way, atoms or fragments could be removed from the input representation of a given training compound to check if they impact on the neuron activation of the compound. In this way relevant atoms and fragments may be identified. This is comparable to the approaches used to interpret QSAR models by perturbing the input representation of test compounds, summarised in Chapter 7. Each approach would yield chemical substructures to be used in IG_hidden.

Another possibility could be to use generative models in order to generate fragments that strongly activate a hidden neuron. Generative models are widely used for feature visualisation in the image domain (A. Nguyen et al., 2016) and are also applied to de novo design of compounds with desired properties (Bilodeau et al., 2022). For this application, a model would be required to generate substructures rather than complete compounds. The respective neural network QSAR model could be used to score generated fragments on how strongly they activate the relevant neuron of the neural network. This would mean that a separate model needs to be trained for each hidden neuron. For this to be efficient, a transfer learning approach might be possible where a general model would be trained to generate substructures which ideally can be quickly fine-tuned in order to obtain models for individual neurons. Substructures could then be sampled from the obtained generative models to be used in IG_hidden.

Another limitation of the approach is the poor performance on some of the infrequent alerts in the training set. In the current approach, substructures are extracted only from the training set. However, it is not necessary to know the toxicity label of a compound in order to test if it activates a given neuron. Therefore, any compound in principle may be used for the substructure extraction. For instance, compounds from ChEMBL may be used to extend the set of compounds used. Due to the

database size, it may be impractical to use all compounds in ChEMBL. A better strategy may be to cluster ChEMBL compounds according to chemical similarity and then sample from the clusters to ensure a diverse set of compounds is used. In this way the set of compounds used to extract substructures may be more balanced (in terms of different Derek alerts in this instance) than the training set which might improve the extraction of substructures associated with rare alerts.

The absence of substructures belonging to a certain alert may also be because compounds matching an alert do not activate any neuron very strongly. Instead, they might activate a number of neurons with moderate strength. A possible solution might be to extract substructures that activate a combinations of neurons rather than just a single neuron. Ideally, a projection of the original activation space could be found in which the dimensions correspond to the presence of specific alerts (more than this is the case for individual hidden neurons). For instance, Principal Component Analysis (PCA) could be used to obtain a projected space and then substructures corresponding to 'activation' of a given PC could be extracted. Also, this would potentially make the method more efficient as the substructure workflow could be run for a number of PCs which is much smaller than the number of neurons, provided that a large proportion of the original variance can be retained.

A limitation identified in Chapter 10 is that no substructure matches can be found for some of the neurons for a given test compound. Some of the changes suggested above may help to alleviate this problem. Another approach could be to encourage the method to find more generic substructures that match larger numbers of test compounds. For instance, a less strict threshold for neuron activation could be applied to smaller and hence more generic fragments. A trade-off has to be made however, as lowering the threshold too much could mean that unimportant substructures are increasingly extracted. Another strategy might be to describe substructures not as SMILES but rather with more flexible SMARTS. SMARTS patterns may describe specific substructures that strongly activate a neuron while allowing for different variations in the substructure (for instance, a match with any halogen atom rather than chlorine). Some attempts have been made to automatically construct meaningful SMARTS patterns that discriminate molecule classes in a previous study (Bietz et al., 2015).

The method proposed in this thesis was specifically developed to interpret neural networks and hence it is not model-agnostic. It is therefore not applicable to other ML algorithms. Attribution methods have mostly been used to explain predictions for other ML algorithms used in the field of QSAR modelling. As was seen by comparing IG_hidden to IG_input, IG_hidden may correctly explain predictions that are not well explained by input feature attribution methods. To confirm this, benchmarking of IG_hidden on a wider range of datasets as well as against more attribution techniques will be necessary. IG_hidden already performed comparably well as IG_input (when

evaluating on alerts rather than individual compounds). The proposed changes to the substructure extraction (see above) might make IG_hidden even more performant.

12.3 Final conclusions

Multi-task imputation has been shown to be promising technique for toxicity prediction in this work. When experimental toxicity data about test compounds of interest is available, multi-task imputation strategies may provide a clear benefit in performance compared to single task QSAR models. Such strategies could be adopted in various situations to improve the accuracy of toxicity prediction.

A novel method to interpret neural networks was developed. The method may complement the toolbox of existing interpretation techniques as it may provide insights not obtainable from methods focussing on input features. Several suggestions were made above to extend the applicability of the approach to different neural network architectures and how to improve its explanatory performance. Hence the method has the potential to make toxicity predictions more interpretable. This can both increase the confidence in predictions made by the models and provide actionable hypotheses to chemists, for instance when optimising drug candidates to reduce their toxicity.

Appendix

Appendix A

A sparse matrix X with m rows and n columns can be factorised to:

$$X_{m \times n} = U_{m \times d} \times V_{d \times n}$$

In probabilistic matrix factorisation (PMF), the entries of matrix X are considered to originate from a probability distribution (e.g. a Gaussian). The likelihood term of the matrix X given the model can be described as:

$$p(X|U, V, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [\mathcal{N}(X_{i,j}|U_i^T V_j, \alpha^{-1})]^{I_{i,j}}$$

Where $\mathcal{N}(x|\mu, \alpha^{-1})$ is a Gaussian distribution with mean μ and precision α (the inverse α^{-1} is the variance), and $I_{i,j}$ is 1 if the matrix cell $X_{i,j}$ was observed and zero otherwise. Furthermore, multivariate prior distributions for U and V are formulated defined by a mean of 0 and a joint precision for U (α_U) and V (α_V) across all dimensions. Their likelihood is given by:

$$p(U|\alpha_U) = \prod_{i=1}^N \mathcal{N}(U_i|0, \alpha_U^{-1}I)$$

$$p(V|\alpha_V) = \prod_{j=1}^M \mathcal{N}(V_j|0, \alpha_V^{-1}I)$$

It can be shown that fitting a model by maximising the likelihood term for X is equivalent to the previously shown objective which minimises the sum of squared residuals with squared regularisation terms for U and V (see section 3.4.7). The Macau algorithm (Simm et al., 2015) is an example of Bayesian probabilistic matrix factorisation (BPMF). (Salakhutdinov & Mnih, 2008) The use of fixed regularisation terms requires the search for ideal parameters. By using a Bayesian approach, model complexity can be automatically controlled using Bayesian inference.

Bayesian inference means that initial beliefs about a variable (here the model parameters) are expressed using a prior distribution which is updated to obtain a posterior distribution after the data has been observed. The method is based on Bayes' theorem which states:

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)}$$

where H stands for a hypothesis (here a certain value for model parameters) and E for the evidence (here the observed data in the matrix). $P(H)$ is the prior distribution describing the initial beliefs about hypotheses and $P(E|H)$ is the likelihood term which describes how likely it is to observe the data given a hypothesis. The prior distributions for U and V are formulated as in the equations above.

A special property of Macau is that it allows the incorporation of side information (e.g. chemical descriptors of a compound) into the model. Side information is used to modify the latent distribution of an instance in the form of a linear model. Specifically, the mean of the latent distribution is shifted by adding the product of the side information features and a learned weight matrix. The prior distribution for instance i of U (U_i) is obtained by:

$$p(U_i|x_i, \mu_U, \alpha_U) = \mathcal{N}(U_i|\mu_U + \beta_i^T \times x_i, \alpha_U^{-1})$$

Where x_i is a vector containing side information for instance i and β_i is a weight vector of the same dimensionality as x_i . If the training matrix contains no observations for a U_i , then the posterior distribution is fully determined by the side information.

Another feature of BPFM (and hence Macau) is that no assumptions are made for the parameters of the prior distributions of U and V . Instead, these are inferred using the data from so-called hyperprior distributions in an additional layer of Bayesian inference. Normal-Wishart distributions are used as the hyperprior distributions. Normal-Wishart distributions are the product of a Normal and a Wishart distribution (which is a multivariate generalisation of the gamma distribution). The hyperprior distribution is initialised with uninformative parameters (i.e. containing no beliefs about any of the matrix elements).

The predictive distribution for each entry of the matrix $X_{i,j}$ is given by the product of the posterior distributions U_i and V_j . Since it is difficult to find the joint distribution (U, V) analytically, Gibbs Sampling is used, which is a Markov chain Monte Carlo method (MCMC) (Neal, 1993). A Markov chain describes a stochastic sequence of states where the probability of obtaining a state depends only on the state of the previous chain element. Monte Carlo methods generally refer to algorithms where, instead of solving a task analytically, a good approximation is searched by employing stochastic sampling. Gibbs sampling can be used when a joint distribution cannot be easily found analytically (and hence directly sampled from), but sampling is possible from the conditional distributions.

As stated above, a prediction in the Macau model is obtained by taking the inner product between single samples from the latent distribution over an instance from U (U_i) and an instance from V (V_j)

(e.g. a user and a movie, respectively, in the Netflix task). By taking many consecutive samples from U_i and V_j , a distribution for the predicted value can be obtained. Each sampling operation is conditioned on the observed data, the respective counterpart of the previous iteration (U for V and vice versa) and the respective hyperparameters common for all U_i and V_j respectively. The hyperparameters of the prior distribution are inferred from hyperprior distributions. Hence, prior to the sampling from the posterior distributions for U and V , the hyperparameters of these have to be sampled conditioned on the hyperpriors and the posterior distribution of the previous iteration. The Gibbs sampling can be summarised as follows:

- Initialise the prior distributions U^1 and V^1
- For t in $[1, \dots, T]$ (the number of samples T to be drawn is defined by the user):
 - Sample the hyperparameters ($\Theta = \{mean, precision\}$) for U and V (from the hyperprior distribution) conditioned on the current state of U and V and the fixed parameters of the hyperprior distributions:

$$\Theta_U^t \sim p(\Theta_U | U^t, \Theta_0)$$

$$\Theta_V^t \sim p(\Theta_V | V^t, \Theta_0)$$

- For each i in $[1, \dots, N]$ (N instances in U):
 - Sample a vector representing instance i , conditioned on observed data X , the current state of V and the current hyperparameters for U :

$$U_i^{t+1} \sim p(U_i | X, V^t, \Theta_U^t)$$

- For each j in $[1, \dots, M]$ (M instances in V):
 - Sample a vector representing instance j , conditioned on observed data X , the current state of U and the current hyperparameters for V :

$$V_j^{t+1} \sim p(V_j | X, U^t, \Theta_V^t)$$

- A value for each $X_{i,j}$ can be computed by multiplying the sampled U_i and V_j . The obtained values form part of the predictive distribution for each cell of the matrix.
- From all samples in the predictive distribution, the first n (to be defined by the user) samples are discarded as so-called burn-in samples.

Appendix B

Table B1 Median ROC-AUC values and interquartile ranges for compound-based splits. Median scores and interquartile-ranges for each technique and dataset on the test set across the 20 random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model for each dataset in bold.

		Ames	Tox21
Single task	RF	0.866 (0.865-0.866)	0.818 (0.817-0.820)
	XGB	0.863 (0.862-0.864)	0.811 (0.811-0.813)
	ST-DNN	0.842 (0.840-0.844)	0.788 (0.784-0.790)
Multi-task	RF-FN	0.862 (0.860-0.863)	0.803 (0.802-0.804)
	XGB-FN	0.853 (0.852-0.855)	0.812 (0.811-0.813)
	ST-DNN-FN	0.836 (0.835-0.839)	0.774 (0.770-0.779)
	MT-DNN	0.871 (0.868-0.874)	0.825 (0.822-0.830)
	Macau	0.873 (0.872-0.874)	0.830 (0.830-0.831)

Table B2 Median ROC-AUC values and interquartile ranges for assay-based splits. Single task models are included as a benchmark. Median scores and interquartile ranges for each technique and dataset on the test set across 20 different random seeds. Before computing the median, the mean across the different assays for a single run was calculated. The best model for each dataset is in bold.

		Ames	Tox21
Single task	RF	0.848 (0.846-0.849)	0.817 (0.815-0.818)
	XGB	0.848 (0.846-0.848)	0.805 (0.804-0.807)
	ST-DNN	0.831 (0.829-0.833)	0.780 (0.778-0.783)
Multi-task	XGB-FN	0.923 (0.923-0.924)	0.866 (0.865-0.867)
	MT-DNN	0.935 (0.934-0.938)	0.867 (0.864-0.870)
	Macau	0.944 (0.943-0.945)	0.888 (0.887-0.888)

Comparison of traditional single task and multi-task models

Table B3 Median MCC scores on compound-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	mean
RF	0.632	0.571	0.376	0.436	0.536	0.614	0.57	0.59	0.247	0.442	0.647	0.64	0.525
XGB	0.616	0.593	0.332	0.47	0.588	0.639	0.72	0.609	0.317	0.375	0.646	0.653	0.547
DNN	0.577	0.563	0.358	0.375	0.524	0.609	0.65	0.599	0.244	0.436	0.655	0.655	0.520
RF_FN	0.603	0.592	0.342	0.227	0.554	0.554	0.616	0.55	0.305	0.496	0.636	0.633	0.509
XGB_FN	0.6	0.568	0.344	0.335	0.502	0.622	0.63	0.655	0.315	0.485	0.637	0.662	0.529
DNN_FN	0.611	0.579	0.38	0.296	0.516	0.606	0.64	0.585	0.305	0.501	0.628	0.641	0.524
MT-DNN	0.594	0.584	0.268	0.371	0.564	0.56	0.646	0.65	0.418	0.518	0.662	0.66	0.541
Macau	0.553	0.535	0.289	0.24	0.525	0.509	0.635	0.529	0.409	0.437	0.624	0.614	0.492
RF_FN, imput	0.756	0.748	0.562	0.607	0.675	0.79	0.712	0.765	0.669	0.825	0.759	0.787	0.721
XGB_FN, imput	0.764	0.759	0.47	0.576	0.677	0.807	0.735	0.78	0.683	0.747	0.724	0.803	0.710
DNN_FN, imput	0.737	0.748	0.522	0.55	0.702	0.734	0.704	0.743	0.573	0.647	0.711	0.775	0.679

Table B4 Median F1 scores on compound-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	Mean
RF	0.715	0.694	0.449	0.516	0.541	0.645	0.58	0.64	0.364	0.492	0.714	0.735	0.590
XGB	0.723	0.733	0.423	0.539	0.609	0.651	0.749	0.657	0.326	0.463	0.713	0.742	0.611
DNN	0.685	0.707	0.4	0.406	0.545	0.629	0.675	0.63	0.341	0.475	0.723	0.749	0.580
RF_FN	0.696	0.714	0.4	0.308	0.581	0.584	0.64	0.604	0.393	0.577	0.711	0.726	0.578
XGB_FN	0.71	0.717	0.448	0.419	0.545	0.649	0.667	0.699	0.385	0.559	0.714	0.756	0.606
DNN_FN	0.707	0.708	0.442	0.308	0.553	0.633	0.675	0.617	0.403	0.537	0.708	0.741	0.586
MT-DNN	0.701	0.718	0.337	0.4	0.598	0.594	0.671	0.684	0.486	0.582	0.735	0.757	0.605
Macau	0.651	0.666	0.356	0.254	0.535	0.522	0.653	0.558	0.449	0.5	0.699	0.718	0.547
RF_FN, imput	0.824	0.827	0.635	0.667	0.688	0.812	0.734	0.796	0.719	0.857	0.812	0.849	0.768
XGB_FN, imput	0.83	0.841	0.557	0.64	0.706	0.813	0.764	0.811	0.73	0.795	0.784	0.859	0.761
DNN_FN, imput	0.806	0.831	0.592	0.602	0.722	0.756	0.736	0.772	0.626	0.696	0.773	0.84	0.729

Table B5 Median ROC-AUC scores on compound-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	Mean
RF	0.891	0.872	0.773	0.85	0.902	0.878	0.942	0.912	0.736	0.849	0.893	0.893	0.866
XGB	0.897	0.879	0.789	0.847	0.878	0.874	0.905	0.886	0.806	0.812	0.879	0.907	0.863
DNN	0.871	0.867	0.759	0.775	0.896	0.867	0.891	0.89	0.694	0.813	0.876	0.908	0.842
RF_FN	0.865	0.883	0.745	0.846	0.909	0.837	0.933	0.928	0.775	0.838	0.88	0.894	0.861
XGB_FN	0.868	0.862	0.798	0.813	0.835	0.845	0.914	0.92	0.791	0.795	0.883	0.912	0.853
DNN_FN	0.862	0.858	0.769	0.778	0.875	0.839	0.882	0.907	0.684	0.807	0.881	0.904	0.837
MT-DNN	0.884	0.881	0.802	0.821	0.896	0.852	0.919	0.923	0.807	0.865	0.896	0.908	0.871
Macau	0.876	0.872	0.784	0.846	0.894	0.885	0.932	0.925	0.828	0.854	0.884	0.904	0.874
RF_FN, imput	0.952	0.951	0.874	0.948	0.972	0.936	0.98	0.979	0.946	0.974	0.948	0.956	0.951
XGB_FN, imput	0.955	0.947	0.876	0.947	0.963	0.944	0.981	0.975	0.96	0.968	0.951	0.965	0.953
DNN_FN, imput	0.94	0.935	0.871	0.903	0.948	0.939	0.954	0.953	0.832	0.921	0.933	0.959	0.924

Table B6 Median MCC scores on compound-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	mean
RF	0.428	0.526	0.724	0.454	0.417	0.564	0.207	0.403	0.167	0.195	0.446	0.29	0.402
XGB	0.498	0.587	0.694	0.395	0.4	0.528	0.205	0.437	0.277	0.203	0.516	0.376	0.426
DNN	0.439	0.57	0.636	0.393	0.43	0.562	0.239	0.414	0.311	0.218	0.486	0.315	0.418
RF_FN	0.4	0.526	0.707	0.409	0.405	0.584	0.187	0.331	0.179	0.092	0.443	0.243	0.376
XGB_FN	0.491	0.568	0.692	0.409	0.412	0.572	0.23	0.46	0.257	0.303	0.492	0.336	0.435
DNN_FN	0.433	0.57	0.657	0.349	0.402	0.546	0.291	0.393	0.294	0.211	0.456	0.287	0.407
MT-DNN	0.468	0.577	0.692	0.365	0.42	0.598	0.212	0.434	0.326	0.253	0.49	0.348	0.432
Macau	0.369	0.587	0.689	0.281	0.433	0.487	0	0.317	0.089	0	0.385	0.192	0.319
RF_FN, imput	0.495	0.526	0.737	0.418	0.632	0.651	0.376	0.472	0.388	0.248	0.557	0.404	0.492
XGB_FN, imput	0.544	0.615	0.784	0.41	0.592	0.697	0.409	0.53	0.443	0.367	0.617	0.506	0.543
DNN_FN, imput	0.488	0.568	0.7	0.406	0.565	0.641	0.403	0.491	0.452	0.317	0.54	0.442	0.501

Table B7 Median F1 scores on compound-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean
RF	0.462	0.533	0.725	0.385	0.458	0.556	0.196	0.441	0.144	0.161	0.488	0.231	0.398
XGB	0.556	0.578	0.696	0.4	0.43	0.549	0.225	0.53	0.301	0.124	0.59	0.411	0.449
DNN	0.48	0.565	0.645	0.376	0.442	0.539	0.213	0.483	0.319	0.219	0.55	0.313	0.428
RF_FN	0.447	0.533	0.706	0.354	0.42	0.582	0.167	0.407	0.147	0.104	0.479	0.237	0.382
XGB_FN	0.549	0.565	0.699	0.431	0.457	0.585	0.252	0.549	0.281	0.284	0.569	0.373	0.381
DNN_FN	0.479	0.565	0.657	0.352	0.415	0.544	0.253	0.45	0.286	0.203	0.512	0.29	0.466
MT-DNN	0.509	0.571	0.691	0.345	0.45	0.592	0.18	0.498	0.316	0.216	0.553	0.337	0.417
Macau	0.376	0.578	0.687	0.176	0.396	0.395	0	0.323	0.034	0	0.41	0.106	0.290
RF_FN, imput	0.539	0.533	0.735	0.342	0.661	0.629	0.367	0.546	0.347	0.267	0.598	0.43	0.500
XGB_FN, imput	0.596	0.607	0.783	0.429	0.617	0.701	0.425	0.607	0.465	0.343	0.676	0.537	0.566
DNN_FN, imput	0.527	0.565	0.706	0.41	0.575	0.644	0.403	0.556	0.471	0.329	0.599	0.456	0.52

Table B8 Median ROC-AUC scores on compound-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean
RF	0.896	0.78	0.895	0.793	0.783	0.824	0.775	0.815	0.795	0.785	0.844	0.842	0.819
XGB	0.888	0.787	0.878	0.779	0.745	0.824	0.742	0.814	0.814	0.743	0.87	0.853	0.811
DNN	0.883	0.746	0.834	0.742	0.737	0.812	0.755	0.793	0.791	0.714	0.843	0.801	0.788
RF_FN	0.871	0.778	0.891	0.813	0.769	0.832	0.777	0.799	0.75	0.709	0.821	0.827	0.803
XGB_FN	0.882	0.762	0.885	0.754	0.755	0.847	0.773	0.82	0.812	0.751	0.865	0.833	0.812
DNN_FN	0.869	0.73	0.828	0.724	0.717	0.814	0.751	0.773	0.759	0.702	0.836	0.804	0.776
MT-DNN	0.879	0.767	0.884	0.788	0.772	0.847	0.812	0.823	0.827	0.784	0.871	0.865	0.827
Macau	0.888	0.792	0.905	0.787	0.787	0.846	0.805	0.822	0.82	0.805	0.859	0.847	0.830
RF_FN, imput	0.915	0.842	0.983	0.883	0.853	0.929	0.905	0.857	0.885	0.836	0.906	0.907	0.892
XGB_FN, imput	0.917	0.828	0.984	0.822	0.847	0.943	0.894	0.868	0.911	0.853	0.908	0.917	0.891
DNN_FN, imput	0.905	0.786	0.915	0.805	0.817	0.887	0.844	0.838	0.861	0.804	0.884	0.88	0.852

Comparison of single task and multi-task imputation models

Table B9 Median MCC scores on assay-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	mean
RF	0.576	0.538	0.432	0.355	0.584	0.572	0.66	0.641	0.315	0.413	0.553	0.621	0.522
XGB	0.607	0.544	0.402	0.41	0.536	0.572	0.691	0.65	0.279	0.524	0.605	0.62	0.537
DNN	0.575	0.528	0.415	0.172	0.518	0.561	0.667	0.65	0.27	0.408	0.591	0.638	0.499
RF_FN	0.699	0.698	0.437	0.546	0.691	0.687	0.704	0.735	0.412	0.571	0.669	0.742	0.633
XGB_FN	0.726	0.695	0.48	0.566	0.732	0.802	0.726	0.711	0.559	0.728	0.697	0.705	0.677
DNN_FN	0.68	0.648	0.449	0.505	0.673	0.706	0.722	0.704	0.412	0.63	0.704	0.734	0.631
MT-DNN	0.711	0.672	0.508	0.585	0.716	0.71	0.752	0.789	0.511	0.683	0.728	0.746	0.676
Macau	0.716	0.708	0.426	0.566	0.73	0.743	0.722	0.743	0.562	0.704	0.735	0.796	0.679

Table B10 Median F1 scores on assay-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	Mean
RF	0.688	0.655	0.483	0.414	0.617	0.597	0.685	0.683	0.338	0.417	0.622	0.71	0.576
XGB	0.705	0.686	0.436	0.469	0.54	0.597	0.711	0.688	0.361	0.603	0.689	0.722	0.601
DNN	0.69	0.662	0.483	0.238	0.547	0.591	0.692	0.69	0.326	0.457	0.676	0.727	0.565
RF_FN	0.785	0.782	0.504	0.595	0.721	0.701	0.733	0.77	0.462	0.6	0.729	0.809	0.683
XGB_FN	0.799	0.785	0.583	0.593	0.761	0.81	0.755	0.742	0.606	0.769	0.765	0.783	0.729
DNN_FN	0.769	0.747	0.538	0.554	0.708	0.725	0.752	0.732	0.465	0.682	0.769	0.799	0.687
MT-DNN	0.789	0.766	0.6	0.616	0.741	0.723	0.775	0.816	0.562	0.723	0.791	0.811	0.726
Macau	0.788	0.788	0.508	0.593	0.756	0.747	0.747	0.766	0.6	0.742	0.793	0.849	0.723

Table B11 Median ROC-AUC scores on assay-based splits: Ames

	TA100	TA100_S9	TA102	TA102_S9	TA1535	TA1535_S9	TA1537	TA1537_S9	TA97	TA97_S9	TA98	TA98_S9	Mean
RF	0.871	0.864	0.776	0.699	0.909	0.864	0.886	0.928	0.781	0.825	0.894	0.883	0.848
XGB	0.879	0.856	0.804	0.744	0.907	0.85	0.894	0.918	0.702	0.823	0.884	0.902	0.847
DNN	0.862	0.848	0.735	0.682	0.894	0.827	0.899	0.917	0.723	0.806	0.883	0.907	0.832
RF_FN	0.902	0.922	0.857	0.863	0.969	0.947	0.947	0.951	0.868	0.957	0.94	0.934	0.921
XGB_FN	0.932	0.906	0.853	0.865	0.97	0.916	0.958	0.97	0.871	0.971	0.936	0.932	0.923
DNN_FN	0.917	0.893	0.805	0.804	0.964	0.901	0.955	0.964	0.819	0.908	0.936	0.944	0.901
MT-DNN	0.928	0.916	0.888	0.881	0.98	0.932	0.98	0.981	0.888	0.971	0.946	0.946	0.936
Macau	0.944	0.932	0.897	0.893	0.98	0.951	0.979	0.982	0.881	0.977	0.953	0.959	0.944

Table B12 Median MCC scores on assay-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	mean
RF	0.475	0.607	0.592	0.392	0.325	0.571	0.329	0.367	0.191	0.362	0.483	0.18	0.406
XGB	0.518	0.58	0.579	0.363	0.292	0.568	0.294	0.423	0.273	0.392	0.507	0.223	0.418
DNN	0.527	0.592	0.554	0.308	0.317	0.603	0.327	0.395	0.268	0.363	0.514	0.238	0.417
RF_FN	0.496	0.621	0.681	0.409	0.422	0.639	0.338	0.439	0.291	0.336	0.536	0.346	0.462
XGB_FN	0.543	0.642	0.677	0.443	0.524	0.703	0.381	0.483	0.411	0.462	0.611	0.372	0.521
DNN_FN	0.552	0.599	0.597	0.396	0.43	0.676	0.381	0.477	0.416	0.401	0.571	0.328	0.485
MT-DNN	0.555	0.626	0.567	0.428	0.435	0.704	0.371	0.503	0.422	0.383	0.607	0.438	0.503
Macau	0.5	0.621	0.514	0.272	0.368	0.63	0.111	0.513	0.132	0.17	0.529	0.267	0.385

Table B13 Median F1 scores on assay-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean
RF	0.518	0.587	0.59	0.328	0.295	0.551	0.276	0.422	0.157	0.281	0.519	0.124	0.387
XGB	0.573	0.568	0.568	0.331	0.268	0.559	0.308	0.51	0.306	0.396	0.58	0.259	0.436
DNN	0.564	0.575	0.544	0.258	0.308	0.599	0.238	0.473	0.263	0.356	0.567	0.256	0.417
RF_FN	0.536	0.603	0.686	0.333	0.373	0.629	0.295	0.514	0.25	0.345	0.594	0.288	0.453
XGB_FN	0.594	0.632	0.675	0.429	0.513	0.71	0.397	0.561	0.437	0.475	0.668	0.405	0.541
DNN_FN	0.588	0.587	0.603	0.361	0.409	0.687	0.33	0.545	0.42	0.414	0.625	0.342	0.492
MT-DNN	0.6	0.601	0.556	0.428	0.458	0.703	0.323	0.568	0.413	0.373	0.657	0.451	0.511
Macau	0.497	0.603	0.5	0.169	0.305	0.588	0.051	0.541	0.086	0.08	0.558	0.184	0.347

Table B14 Median ROC-AUC scores on assay-based splits: Tox21

	NR-AhR	NR-AR	NR-AR-LBD	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean
RF	0.91	0.81	0.809	0.778	0.718	0.896	0.788	0.827	0.812	0.776	0.882	0.794	0.817
XGB	0.887	0.756	0.857	0.814	0.685	0.886	0.735	0.819	0.809	0.755	0.889	0.775	0.806
DNN	0.872	0.727	0.833	0.755	0.685	0.859	0.665	0.796	0.81	0.757	0.875	0.734	0.781
RF_FN	0.927	0.857	0.958	0.857	0.763	0.949	0.874	0.847	0.874	0.833	0.907	0.887	0.878
XGB_FN	0.901	0.794	0.942	0.833	0.784	0.953	0.876	0.853	0.898	0.773	0.931	0.862	0.867
DNN_FN	0.888	0.77	0.927	0.804	0.749	0.935	0.771	0.827	0.875	0.79	0.899	0.811	0.837
MT-DNN	0.909	0.782	0.94	0.822	0.768	0.965	0.863	0.858	0.899	0.815	0.919	0.873	0.868
Macau	0.921	0.838	0.951	0.886	0.768	0.967	0.909	0.872	0.897	0.831	0.913	0.899	0.888

Appendix C

Table C1 Manually selected hyperparameters for XGB and XGB-FN

Hyperparameter	Selected value
<i>Num_round</i>	700
<i>Eta</i>	0.1
<i>Colsample_bytree</i>	0.5
<i>Alpha</i>	1
<i>lambda</i>	10
<i>Scale_pos_weight</i>	1

Table C2 Manually selected hyperparameters for multi-task DNN

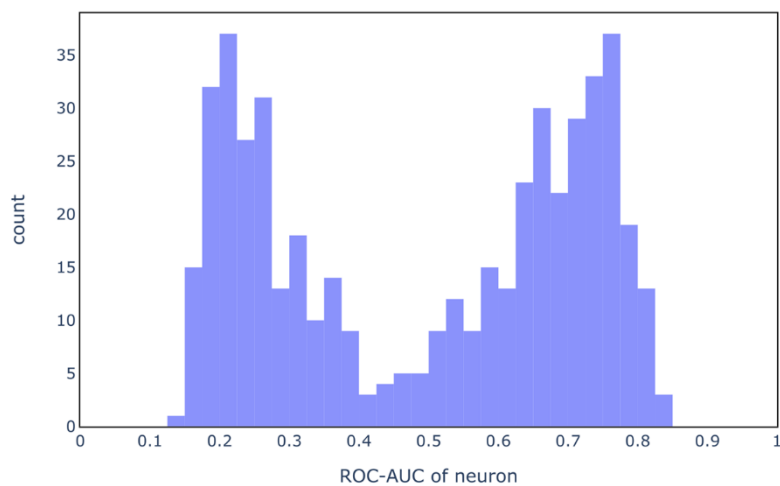
Hyperparameter	Values
Hidden layers	2
Nodes per hidden layer	2048
Learning rate	0.0003
Dropout	0.2
L2 regularisation	0.0001
Batch size	50
Number of epochs	10
Class weight	1

Table C3 Manually selected hyperparameters for Macau

Hyperparameter	Values
<i>Num_latent</i>	16
<i>nsamples</i>	3200
<i>burnin</i>	400

Appendix D

A: training set



B: validation set

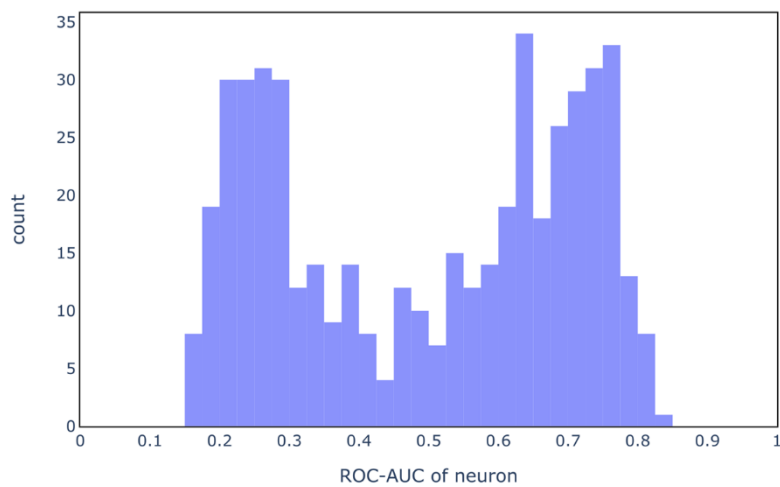


Figure D1 ROC_AUC scores for individual neurons: *dropout_model*. Neuron activations of compounds were considered as predictions and hence it was evaluated how well neurons rank toxic compounds higher than non-toxic ones. **A:** training set, **B:** validation set.

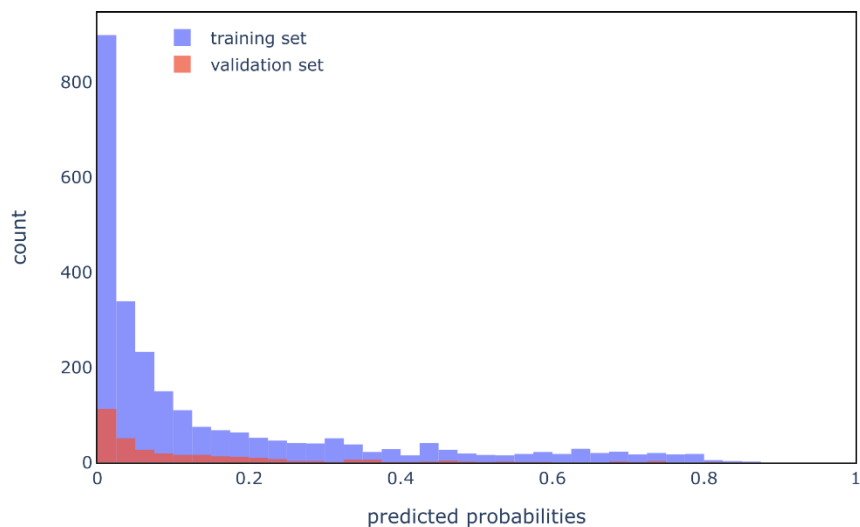


Figure D2 Predicted probabilities of model on ATG_RXRb_TRANS_up after 10 additional training epochs.

Table D1 Hyperparameters of studied DNNs from the ToxCast and ChEMBL dataset.

Hyperparameter	ATG_ERa_TRANS_up	ATG_RXRb_TRANS_up	ChEMBL226	ChEMBL239
Morgan FP radius	1	2	1	2
Neurons first layer	512	512	512	512
Neurons second layer	512	256	512	512
Batch size for optimisation	32	64	64	64
L2 regularisation of neuron weights	1×10^{-5}	1×10^{-5}	0	0.001
Dropout (both hidden layers)	0.2	0.2	0.2	0.2
Learning rate	0.001	0.0001	0.001	0.001
weights	1	1	balanced	1

Bibliography

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *ArXiv:1603.04467v2*.
<https://doi.org/10.48550/arXiv.1603.04467>
- Achten, C., & Hofmann, T. (2009). Native polycyclic aromatic hydrocarbons (PAH) in coals - A hardly recognized source of environmental contamination. *Science of the Total Environment*, *407*(8), 2461–2473. <https://doi.org/10.1016/j.scitotenv.2008.12.008>
- Agarap, A. F. M. (2019). Deep Learning using Rectified Linear Units (ReLU). *ArXiv:1803.08375v2*.
<https://doi.org/10.48550/arXiv.1803.08375>
- Aleksić, S., Seeliger, D., & Brown, J. B. (2022). ADMET Predictability at Boehringer Ingelheim: State-of-the-Art, and Do Bigger Datasets or Algorithms Make a Difference? *Molecular Informatics*, *41*(2), 1–16. <https://doi.org/10.1002/minf.202100113>
- Allen, T. E. H., Wedlake, A. J., Gelzynyte, E., Gong, C., Goodman, J. M., Gutsell, S., & Russell, P. J. (2020). Neural network activation similarity: a new measure to assist decision making in chemical toxicology. *Chemical Science*, *11*, 7335–7448. <https://doi.org/10.1039/d0sc01637c>
- Amberg, A., Beilke, L., Bercu, J., Bower, D., Brigo, A., Cross, K. P., Custer, L., Dobo, K., Dowdy, E., Ford, K. A., Glowienke, S., Van Gompel, J., Harvey, J., Hasselgren, C., Honma, M., Jolly, R., Kemper, R., Kenyon, M., Kruhlak, N., ... Myatt, G. J. (2016). Principles and procedures for implementation of ICH M7 recommended (Q)SAR analyses. *Regulatory Toxicology and Pharmacology*, *77*, 13–24. <https://doi.org/10.1016/j.yrtph.2016.02.004>
- Ames, B. N., Joyce, M., & Yamasaki, E. (1975). Methods for Detecting Carcinogens and Mutagens With the Salmonella/Mammalian-Microsome Mutagenicity Test. *Mutation Research*, *31*, 347–363. [https://doi.org/10.1016/0165-1161\(75\)90046-1](https://doi.org/10.1016/0165-1161(75)90046-1)
- Ancona, M., Ceolini, E., Öztireli, C., & Gross, M. (2018). Towards better understanding of gradient-based attribution methods for deep neural networks. *ArXiv:1711.06104v4*, 1–16. <https://doi.org/10.48550/arXiv.1711.06104>
- Ashby, J., & Tennant, R. W. (1988). Chemical structure, Salmonella mutagenicity and extent of carcinogenicity as indicators of genotoxic carcinogenesis among 222 chemicals tested in rodents by the U.S. NCI/NTP. *Mutation Research - Genetic Toxicology*, *204*(1), 17–115. [https://doi.org/10.1016/0165-1218\(88\)90114-0](https://doi.org/10.1016/0165-1218(88)90114-0)
- Bae, S. Y., Lee, J., Jeong, J., Lim, C., & Choi, J. (2021). Effective data-balancing methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints. *Computational Toxicology*, *20*, 100178. <https://doi.org/10.1016/j.comtox.2021.100178>
- Baell, J. B., & Holloway, G. A. (2010). New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *Journal of Medicinal Chemistry*, *53*(7), 2719–2740. <https://doi.org/10.1021/jm901137j>
- Baird, W. M., Hooven, L. A., & Mahadevan, B. (2005). Carcinogenic Polycyclic Aromatic Hydrocarbon-DNA Adducts and Mechanism of Action. *Environmental and Molecular Mutagenesis*, *45*(2–3), 106–114. <https://doi.org/10.1002/em.20095>

- Balchin, D., Hayer-Hartl, M., & Hartl, F. U. (2016). In vivo aspects of protein folding and quality control. *Science*, 353(6294). <https://doi.org/10.1126/science.aac4354>
- Basak, S. C., Grunwald, G. D., Gute, B. D., Balasubramanian, K., & Opitz, D. (2000). Use of Statistical and Neural Net Approaches in Predicting Toxicity of Chemicals. *Journal of Chemical Information and Computer Sciences*, 40(4), 885–890. <https://doi.org/10.1021/ci9901136>
- Bassan, A., Alves, V. M., Amberg, A., Anger, L. T., Auerbach, S., Beilke, L., Bender, A., Cronin, M. T. D., Cross, K. P., Hsieh, J.-H., Greene, N., Kemper, R., Kim, M. T., Mumtaz, M., Noeske, T., Pavan, M., Pletz, J., Russo, D. P., Sabnis, Y., ... Myatt, G. J. (2021). In silico approaches in organ toxicity hazard assessment: Current status and future needs in predicting liver toxicity. *Computational Toxicology*, 20, 100187. <https://doi.org/10.1016/j.comtox.2021.100187>
- Batista, G. E. A. P. A., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5–6), 519–533. <https://doi.org/10.1080/713827181>
- Benedict, W. F., Baker, M. S., Haroun, L., Choi, E., & Ames, B. N. (1977). Mutagenicity of Cancer Chemotherapeutic Agents in the Salmonella/Microsome Test. *Cancer Research*, 37, 2209–2213. <https://pubmed.ncbi.nlm.nih.gov/193638/>
- Benigni, R., Battistelli, C. L., Bossa, C., Tcheremenskaia, O., & Crettaz, P. (2013). New perspectives in toxicological information management, and the role of ISSTOX databases in assessing chemical mutagenicity and carcinogenicity. *Mutagenesis*, 28(4), 401–409. <https://doi.org/10.1093/mutage/get016>
- Benigni, R., & Bossa, C. (2011). Mechanisms of chemical carcinogenicity and mutagenicity: A review with implications for predictive toxicology. *Chemical Reviews*, 111(4), 2507–2536. <https://doi.org/10.1021/cr100222q>
- Benigni, R., Bossa, C., Richard, A. M., & Yang, C. (2008). A novel approach: Chemical relational databases, and the role of the ISSCAN database on assessing chemical carcinogenicity. *Annali Dell'Istituto Superiore Di Sanità*, 44(1), 48–56. <https://pubmed.ncbi.nlm.nih.gov/18469376/>
- Benigni, R., Giuliani, A., Franke, R., & Gruska, A. (2000). Quantitative structure-activity relationships of mutagenic and carcinogenic aromatic amines. *Chemical Reviews*, 100(10), 3697–3714. <https://doi.org/10.1021/cr9901079>
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems 24*. <https://papers.nips.cc/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html>
- Bietz, S., Schomburg, K. T., Hilbig, M., & Rarey, M. (2015). Discriminative Chemical Patterns: Automatic and Interactive Design. *Journal of Chemical Information and Modeling*, 55(8), 1535–1546. <https://doi.org/10.1021/acs.jcim.5b00323>
- Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., & Jensen, K. F. (2022). Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5), e1608. <https://doi.org/10.1002/wcms.1608>
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End Learning for Self-Driving Cars. *ArXiv:1604.07316*, 1–9. <https://doi.org/10.48550/arXiv.1604.07316>
- Bosc, N., Atkinson, F., Felix, E., Gaulton, A., Hersey, A., & Leach, A. R. (2019). Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. *Journal of Cheminformatics*, 11, 4. <https://doi.org/10.1186/s13321-018-0325-4>

- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152. <https://doi.org/10.1145/130385.130401>
- Botchkarev, A. (2019). Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *Interdisciplinary Journal of Information, Knowledge and Management*, 14, 45–79. <https://doi.org/10.28945/4184>
- Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLoS ONE*, 12(6), e0177678. <https://doi.org/10.1371/journal.pone.0177678>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. https://doi.org/10.1007/978-3-662-56776-0_10
- Brown, B. R., Firth, W. J., & Yielding, L. W. (1980). Acridine structure correlated with mutagenic activity in Salmonella. *Mutation Research - Fundamental and Molecular Mechanisms of Mutagenesis*, 72(3), 373–388. [https://doi.org/10.1016/0027-5107\(80\)90112-8](https://doi.org/10.1016/0027-5107(80)90112-8)
- Brownlee, J. (2019). *Loss and Loss functions for Training Deep Learning Neural Networks*. Machine Learning Mastery. <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>
- Brownlee, J. (2020). *Ordinal and One-Hot Encodings for Categorical Data*. Machine Learning Mastery. <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>
- Bugeac, C. A., Ancuceanu, R., & Dinu, M. (2021). QSAR models for active substances against *Pseudomonas aeruginosa* using disk-diffusion test data. *Molecules*, 26(6), 1734. <https://doi.org/10.3390/molecules26061734>
- Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28, 41–75. <https://doi.org/10.1111/j.1468-0319.1995.tb00042.x>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., Consonni, V., Kuz'min, V. E., Cramer, R., Benigni, R., Yang, C., Rathman, J., Terfloth, L., Gasteiger, J., Richard, A., & Tropsha, A. (2014). QSAR modeling: Where have you been? Where are you going to? *Journal of Medicinal Chemistry*, 57(12), 4977–5010. <https://doi.org/10.1021/jm4004285>
- Chollet, F., & others. (2015). *Keras*. <https://keras.io>
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20, 273–297. <https://doi.org/10.1109/64.163674>
- Corvi, R., & Madia, F. (2017). In vitro genotoxicity testing—Can the performance be enhanced? *Food and Chemical Toxicology*, 106(Part B), 600–608. <https://doi.org/10.1016/j.fct.2016.08.024>
- Cronin, M. T. D. (2013). An introduction to chemical grouping, categories and read-across to predict toxicity. In *Chemical Toxicity Prediction* (pp. 1–29). Royal Society of Chemistry. <https://doi.org/10.1039/9781849734400-00001>
- Cross, K. P., & Ponting, D. J. (2021). Developing structure-activity relationships for N-nitrosamine activity. *Computational Toxicology*, 20, 100186. <https://doi.org/10.1016/j.comtox.2021.100186>

- Dalby, A., Nourse, J. G., Hounshell, W. D., Gushurst, A. K. I., Grier, D. L., Leland, B. A., & Laufer, J. (1992). Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences*, 32(3), 244–255. <https://doi.org/10.1021/ci00007a012>
- Davis, I. L., & Stentz, A. (1995). Sensor fusion for autonomous outdoor navigation using neural networks. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 3, 338–343. <https://doi.org/10.1109/IROS.1995.525906>
- de la Vega de León, A., Chen, B., & Gillet, V. J. (2018). Effect of missing data on multitask prediction methods. *Journal of Cheminformatics*, 10, 26. <https://doi.org/10.1186/s13321-018-0281-z>
- Dey, A. (2016). Machine Learning Algorithms: A Review. *International Journal of Computer Science and Information Technologies*, 7(3), 1174–1179. www.ijcsit.com
- Dipple, A. (1995). DNA adducts of chemical carcinogens. *Carcinogenesis*, 16(3), 437–441. <https://doi.org/10.1093/carcin/16.3.437>
- Dronkert, M. L. G., & Kanaar, R. (2001). Repair of DNA interstrand cross-links. *Mutation Research - DNA Repair*, 486(4), 217–247. [https://doi.org/10.1016/S0921-8777\(01\)00092-1](https://doi.org/10.1016/S0921-8777(01)00092-1)
- Drucker, H., Surges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems* 9. <https://papers.nips.cc/paper/1996/hash/d38901788c533e8286cb6400b40b386d-Abstract.html>
- Durant, J. L., Leland, B. A., Henry, D. R., & Nourse, J. G. (2002). Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6), 1273–1280. <https://doi.org/10.1021/ci010132r>
- ECHA. (2017). Non-animal approaches. Current status of regulatory applicability under the REACH, CLP and Biocidal Products regulation. In *Publications Office of the EU*. <https://doi.org/10.2823/000784>
- Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 973–978.
- Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). Visualizing higher-layer features of a deep network. In *University of Montreal Technical Report* (Issue 1341). <https://www.semanticscholar.org/paper/Visualizing-Higher-Layer-Features-of-a-Deep-Network-Erhan-Bengio/65d994fb778a8d9e0f632659fb33a082949a50d3>
- Ertl, P., Rohde, B., & Selzer, P. (2000). Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *Journal of Medicinal Chemistry*, 43(20), 3714–3717. <https://doi.org/10.1021/jm000942e>
- Esposito, C., Landrum, G. A., Schneider, N., Stiefl, N., & Riniker, S. (2021). GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning. *Journal of Chemical Information and Modeling*, 61(6), 2623–2640. <https://doi.org/10.1021/acs.jcim.1c00160>
- Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence*, 20(1), 18–36. <https://doi.org/10.1111/j.0824-7935.2004.t01-1-00228.x>
- EU. (2006). *Regulation (EC) No 1907/2006 of the European Parliament and of the Council*. <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:136:0003:0280:en:PDF>
- EU. (2009). *Regulation (EC) No 1223/2009 of the European Parliament and of the Council*.

- <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32009R1223&from=EN>
- European Medicines Agency. (2018). *ICH guideline M7(R1) on assessment and control of DNA Reactive (Mutagenic) Impurities in Pharmaceuticals to Limit Potential Carcinogenic Risk*. <https://www.ema.europa.eu/en/ich-m7-assessment-control-dna-reactive-mutagenic-impurities-pharmaceuticals-limit-potential>
- Feldmann, C., Philipps, M., & Bajorath, J. (2021). Explainable machine learning predictions of dual - target compounds reveal characteristic structural features. *Scientific Reports*, *11*(21594). <https://doi.org/10.1038/s41598-021-01099-4>
- Ferguson, L. R., & Denny, W. A. (2007). Genotoxicity of non-covalent interactions: DNA intercalators. *Mutation Research - Fundamental and Molecular Mechanisms of Mutagenesis*, *623*(1–2), 14–23. <https://doi.org/10.1016/j.mrfmmm.2007.03.014>
- Fernandez, M., Ban, F., Woo, G., Hsing, M., Yamazaki, T., Leblanc, E., Rennie, P. S., Welch, W. J., & Cherkasov, A. (2018). Toxic Colors: The Use of Deep Learning for Predicting Toxicity of Compounds Merely from Their Graphic Images [Research-article]. *Journal of Chemical Information and Modeling*, *58*(8), 1533–1543. <https://doi.org/10.1021/acs.jcim.8b00338>
- Fishbein, L. (1976). Industrial Mutagens and Potential Mutagens I. Halogenated aliphatic derivatives. *Mutation Research*, *32*(3–4), 267–307. [https://doi.org/10.1016/0165-1110\(76\)90003-8](https://doi.org/10.1016/0165-1110(76)90003-8)
- Flamant, F., Baxter, J. D., Forrest, D., Refetoff, S., Samuels, H., Scanlan, T. S., Vennström, B., & Samarut, J. (2006). The pharmacology and classification of the nuclear receptor superfamily: Thyroid hormone receptors. *Pharmacological Reviews*, *58*(4), 705–711. <https://doi.org/10.1124/pr.58.4.3>
- Forman, B. M., Goode, E., Chen, J., Oro, A. E., Bradley, D. J., Perlmann, T., Noonan, D. J., Burka, L. T., McMorris, T., Lamph, W. W., Evans, R. M., & Weinberger, C. (1995). Identification of a nuclear receptor that is activated by farnesol metabolites. *Cell*, *81*(5), 687–693. [https://doi.org/10.1016/0092-8674\(95\)90530-8](https://doi.org/10.1016/0092-8674(95)90530-8)
- Fortmann-Roe, S. (2012). *Understanding the Bias-Variance Tradeoff*. <http://scott.fortmann-roe.com/docs/BiasVariance.html>
- Fourches, D., Muratov, E., & Tropsha, A. (2010). Trust, But Verify: On the Importance of Chemical Structure Curation in Cheminformatics and QSAR Modeling Research. *Journal of Chemical Information and Modeling*, *50*(7), 1189–1204. <https://doi.org/10.1021/ci100176x>
- Freese, E. (1959). The difference between spontaneous and base-analogue induced mutations of phage T4. *Proceedings of the National Academy of Sciences*, *45*(4), 622–633. <https://doi.org/10.1073/pnas.45.4.622>
- Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, *1*(1), 55–77. <https://doi.org/10.1023/A:1009778005914>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. <https://doi.org/10.2307/2699986>
- Furukawa, A., Ono, S., Yamada, K., Torimoto, N., Asayama, M., & Muto, S. (2022). A local QSAR model based on the stability of nitrenium ions to support the ICH M7 expert review on the mutagenicity of primary aromatic amines. *Genes and Environment*, *44*(10). <https://doi.org/10.1186/s41021-022-00238-1>
- Gardiner, E. J., & Gillet, V. J. (2015). Perspectives on Knowledge Discovery Algorithms Recently Introduced in Chemoinformatics: Rough Set Theory, Association Rule Mining, Emerging

- Patterns, and Formal Concept Analysis. *Journal of Chemical Information and Modeling*, 55(9), 1781–1803. <https://doi.org/10.1021/acs.jcim.5b00198>
- Gardner, J. W. (2002). Death by Water Intoxication. *Military Medicine*, 167(5), 432–434. <https://doi.org/10.1093/milmed/167.5.432>
- Gasteiger, J. (2006). The central role of chemoinformatics. *Chemometrics and Intelligent Laboratory Systems*, 82(1–2), 200–209. <https://doi.org/10.1016/j.chemolab.2005.06.022>
- Gasteiger, J., & Zupan, J. (1993). Neural Networks in Chemistry. *Angewandte Chemie International Edition in English*, 32(4), 503–527. <https://doi.org/10.1002/anie.199305031>
- Gaulton, A., Hersey, A., Nowotka, M., Patrícia Bento, A., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrían-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magariños, M. P., Overington, J. P., Papadatos, G., Smit, I., & Leach, A. R. (2017). The ChEMBL database in 2017. *Nucleic Acids Research*, 45(D1), D945–D954. <https://doi.org/10.1093/nar/gkw1074>
- Gellatly, N., & Sewell, F. (2019). Regulatory acceptance of in silico approaches for the safety assessment of cosmetic-related substances. *Computational Toxicology*, 11, 82–89. <https://doi.org/10.1016/j.comtox.2019.03.003>
- Géron, A. (2019). The Machine Learning Landscape. In *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems* (2nd Editio, pp. 3–32). O'Reilly Media. <https://doi.org/10.1201/9780367229184-1>
- Gini, G., Zanolli, F., Gamba, A., Raitano, G., & Benfenati, E. (2019). Could deep learning in neural networks improve the QSAR models? *SAR and QSAR in Environmental Research*, 30(9), 617–642. <https://doi.org/10.1080/1062936X.2019.1650827>
- Glave, W. R., & Hansch, C. (1972). Relationship between lipophilic character and anesthetic activity. *Journal of Pharmaceutical Sciences*, 61(4), 589–591. <https://doi.org/10.1002/jps.2600610420>
- Gluck, D. J. (1965). A Chemical Structure Storage and Search System Developed at Du Pont. *Journal of Chemical Documentation*, 5(1), 43–51. <https://doi.org/10.1021/c160016a008>
- Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O., & Baker, N. (2017). Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. *ArXiv:1706.06689*. <https://doi.org/10.48550/arXiv.1706.06689>
- Golbraikh, A., Muratov, E., Fourches, D., & Tropsha, A. (2014). Data set modelability by QSAR. *Journal of Chemical Information and Modeling*, 54(1), 1–4. <https://doi.org/10.1021/ci400572x>
- Göller, A. H., Kuhnke, L., Montanari, F., Bonin, A., Schneckener, S., ter Laak, A., Wichard, J., Lobell, M., & Hillisch, A. (2020). Bayer's in silico ADMET platform: a journey of machine learning over the past two decades. *Drug Discovery Today*, 25(9), 1702–1709. <https://doi.org/10.1016/j.drudis.2020.07.001>
- Gomez-Urbe, C. A., & Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4), Article 13. <https://doi.org/10.1145/2843948>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Feedforward Networks. In *Deep Learning* (pp. 164–223). MIT Press. <https://www.deeplearningbook.org/>
- Greene, N., & Naven, R. (2009). Early toxicity screening strategies. *Current Opinion in Drug Discovery and Development*, 12(1), 90–97.
- Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random

- forests. *Statistics and Computing*, 27(3), 659–678. <https://doi.org/10.1007/s11222-016-9646-1>
- Groom, C. R., Bruno, I. J., Lightfoot, M. P., & Ward, S. C. (2016). The Cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72(2), 171–179. <https://doi.org/10.1107/S2052520616003954>
- Guha, R., & Jurs, P. C. (2005). Determining the validity of a QSAR model - A classification approach. *Journal of Chemical Information and Modeling*, 45(1), 65–73. <https://doi.org/10.1021/ci0497511>
- Gujarati, D. (2019). The linear regression model. In *Linear Regression: A Mathematical Introduction* (pp. 1–21). SAGE Publications Inc. <https://doi.org/10.1017/cbo9780511802447.012>
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157–1182. <https://doi.org/10.1016/j.aca.2011.07.027>
- Hamel, A., Roy, M., & Proudlock, R. (2016). The Bacterial Reverse Mutation Test. In *Genetic Toxicology Testing* (pp. 79–138). Elsevier Inc. <https://doi.org/10.1016/b978-0-12-800764-8.00004-5>
- Hammett, L. P. (1937). The Effect of Structure upon the Reactions of Organic Compounds. Benzene Derivatives. *Journal of the American Chemical Society*, 59(1), 96–103. <https://doi.org/10.1021/ja01280a022>
- Hamzic, S., Lewis, R., Desrayaud, S., Soylu, C., Fortunato, M., Gerebtzoff, G., & Rodríguez-Pérez, R. (2022). Predicting In Vivo Compound Brain Penetration Using Multi-task Graph Neural Networks. *Journal of Chemical Information and Modeling*, 62(13), 3180–3190. <https://doi.org/10.1021/acs.jcim.2c00412>
- Hansch, C., Muir, R. M., Fujita, T., Maloney, P. P., Geiger, F., & Streich, M. (1963). The Correlation of Biological Activity of Plant Growth Regulators and Chloromycetin Derivatives with Hammett Constants and Partition Coefficients. *Journal of the American Chemical Society*, 85(18), 2817–2824. <https://doi.org/10.1021/ja00901a033>
- Hansen, K., Mika, S., Schroeter, T., Sutter, A., ter Laak, A., Steger-Hartmann, T., Heinrich, N., & Müller, K.-R. (2009). Benchmark data set for in silico prediction of Ames mutagenicity. *Journal of Chemical Information and Modeling*, 49(9), 2077–2081. <https://doi.org/10.1021/ci900161g>
- Hanser, T., Barber, C., Marchaland, J. F., & Werner, S. (2016). Applicability domain: towards a more formal definition. *SAR and QSAR in Environmental Research*, 27(11), 865–881. <https://doi.org/10.1080/1062936X.2016.1250229>
- Hanser, T., Barber, C., Rosser, E., Vessey, J. D., Webb, S. J., & Werner, S. (2014). Self organising hypothesis networks: A new approach for representing and structuring SAR knowledge. *Journal of Cheminformatics*, 6(21). <https://doi.org/10.1186/1758-2946-6-21>
- Hardy, A., Benford, D., Halldorsson, T., Jeger, M. J., Knutsen, H. K., More, S., Naegeli, H., Noteborn, H., Ockleford, C., Ricci, A., Rychen, G., Schlatter, J. R., Silano, V., Solecki, R., Turck, D., Benfenati, E., Chaudhry, Q. M., Craig, P., Frampton, G., ... Younes, M. (2017). Guidance on the use of the weight of evidence approach in scientific assessments. *EFSA Journal*, 15(8), e04971. <https://doi.org/10.2903/j.efsa.2017.4971>
- Harren, T., Matter, H., Hessler, G., Rarey, M., & Grebner, C. (2021). Interpretation of Structure – Activity Relationships in Real-World Drug Design Data Sets Using Explainable Artificial Intelligence. *Journal of Chemical Information and Modeling*, 62(3), 447–462. <https://doi.org/10.1021/acs.jcim.1c01263>

- Harvey, R. G., & Geacintov, N. E. (1988). Intercalation and Binding of Carcinogenic Hydrocarbon Metabolites to Nucleic Acids. *Accounts of Chemical Research*, 21(2), 66–73. <https://doi.org/10.1021/ar00146a004>
- Hasselgren, C., Ahlberg, E., Akahori, Y., Amberg, A., Anger, L. T., Atienzar, F., Auerbach, S., Beilke, L., Bellion, P., Benigni, R., Bercu, J., Booth, E. D., Bower, D., Brigo, A., Cammerer, Z., Cronin, M. T. D., Crooks, I., Cross, K. P., Custer, L., ... Myatt, G. J. (2019). Genetic toxicology in silico protocol. *Regulatory Toxicology and Pharmacology*, 107, 104403. <https://doi.org/10.1016/j.yrtph.2019.104403>
- Heller, S. R., McNaught, A., Pletnev, I., Stein, S., & Tchekhovskoi, D. (2015). InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, 7(23). <https://doi.org/10.1186/s13321-015-0068-4>
- Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1), 55–67. <https://doi.org/10.2307/1271436>
- Horton, N. J., & Kleinmann, K. P. (2007). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. *American Statistician*, 61(1), 79–90. <https://doi.org/10.5840/owl202153137>
- Hossin, M., & Sulaiman, N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1–11. <https://doi.org/10.5121/ijdkp.2015.5201>
- Huang, R., Xia, M., Sakamuru, S., Zhao, J., Shahane, S. A., Attene-Ramos, M., Zhao, T., Austin, C. P., & Simeonov, A. (2016). Modelling the Tox21 10 K chemical profiles for in vivo toxicity prediction and mechanism characterization. *Nature Communications*, 7, 10425. <https://doi.org/10.1038/ncomms10425>
- Humbeck, L., Morawietz, T., Sturm, N., Zalewski, A., Harnqvist, S., Heyndrickx, W., Holmes, M., & Beck, B. (2021). Don't overweight weights: Evaluation of weighting strategies for multi-task bioactivity classification models. *Molecules*, 26(22), 6959. <https://doi.org/10.3390/molecules26226959>
- Irwin, B. W. J., Levell, J. R., Whitehead, T. M., Segall, M. D., & Conduit, G. J. (2020). Practical Applications of Deep Learning To Impute Heterogeneous Drug Discovery Data. *Journal of Chemical Information and Modeling*, 60(6), 2848–2857. <https://doi.org/10.1021/acs.jcim.0c00443>
- Irwin, B. W. J., Mahmoud, S., Whitehead, T. M., Conduit, G. J., & Segall, M. D. (2020). Imputation versus prediction: applications in machine learning for drug discovery. *Future Drug Discovery*, 2(2). <https://doi.org/10.4155/fdd-2020-0008>
- ISSTOX Chemical Toxicity Databases. (2021). <https://www.iss.it/isstox>
- Jackson, S. P. (2002). Sensing and repairing DNA double-strand breaks. *Carcinogenesis*, 23(5), 687–696. <https://doi.org/10.1093/carcin/23.5.687>
- Janani, C., & Kumari, B. D. R. (2015). PPAR gamma gene – A review. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 9(1), 46–50. <https://doi.org/10.1016/j.dsx.2014.09.015>
- Jiménez-Luna, J., Grisoni, F., & Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2, 573–584. <https://doi.org/10.1038/s42256-020-00236-4>
- Jiménez-Luna, J., Skalic, M., & Weskamp, N. (2022). Benchmarking Molecular Feature Attribution

- Methods with Activity Cliffs. *Journal of Chemical Information and Modeling*, 62(2), 274–283. <https://doi.org/10.1021/acs.jcim.1c01163>
- Jiménez-Luna, J., Skalic, M., Weskamp, N., & Schneider, G. (2021). Coloring Molecules with Explainable Artificial Intelligence for Preclinical Relevance Assessment. *Journal of Chemical Information and Modeling*, 61(3), 1083–1094. <https://doi.org/10.1021/acs.jcim.0c01344>
- Jochum, C., Gasteiger, J., & Ugi, I. (1980). The Principle of Minimum Chemical Distance (PMCD). *Angewandte Chemie International Edition in English*, 19(7), 495–505. <https://doi.org/10.1002/anie.198004953>
- Johnson, C., Ahlberg, E., Anger, L. T., Beilke, L., Benigni, R., Bercu, J., Bobst, S., Bower, D., Brigo, A., Campbell, S., Cronin, M. T. D., Crooks, I., Cross, K. P., Doktorova, T., Exner, T., Faulkner, D., Fearon, I. M., Fehr, M., Gad, S. C., ... Myatt, G. J. (2020). Skin sensitization in silico protocol. *Regulatory Toxicology and Pharmacology*, 116, 104688. <https://doi.org/10.1016/j.yrtph.2020.104688>
- Johnson, M., Basak, S., & Maggiora, G. (1988). A characterization of molecular similarity methods for property prediction. *Mathematical and Computer Modelling*, 11, 630–634. [https://doi.org/10.1016/0895-7177\(88\)90569-9](https://doi.org/10.1016/0895-7177(88)90569-9)
- JRC. (2010). Applicability of QSAR analysis to the evaluation of the toxicological relevance of metabolites and degradates of pesticide active substances for dietary risk assessment. *EFSA Supporting Publications*, 7(5), 50E. <https://doi.org/10.2903/sp.efsa.2010.en-50>
- Ju, K.-S., & Parales, R. E. (2010). Nitroaromatic Compounds, from Synthesis to Biodegradation. *Microbiology and Molecular Biology Reviews*, 74(2), 250–272. <https://doi.org/10.1128/mubr.00006-10>
- Kawajiri, K., & Fujii-Kuriyama, Y. (2017). The aryl hydrocarbon receptor: a multifunctional chemical sensor for host defense and homeostatic maintenance. *Experimental Animals*, 66(2), 75–89. <https://doi.org/10.1538/expanim.16-0092>
- Kazius, J., McGuire, R., & Bursi, R. (2005). Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48(1), 312–320. <https://doi.org/10.1021/jm040835a>
- Kier, L. B., & Hall, L. H. (1981). Derivation and significance of valence molecular connectivity. *Journal of Pharmaceutical Sciences*, 70(6), 583–589. <https://doi.org/10.1002/jps.2600700602>
- Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *ArXiv:1412.6980*. <https://doi.org/10.48550/arXiv.1412.6980>
- Kirkland, D., Aardema, M., Henderson, L., & Müller, L. (2005). Evaluation of the ability of a battery of three in vitro genotoxicity tests to discriminate rodent carcinogens and non-carcinogens: I. Sensitivity, specificity and relative predictivity. *Mutation Research - Genetic Toxicology and Environmental Mutagenesis*, 584(1–2), 1–256. <https://doi.org/10.1016/j.mrgentox.2005.02.004>
- Klaassen, C. (2008). *Casaraett and Doull's Toxicology: The Basic Science of Poisons* (7. Edition). McGraw-Hill.
- Klarner, L., Reutlinger, M., Schindler, T., Deane, C., & Morris, G. (2022). Bias in the Benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms. *ICML 2022, 2nd AI for Science Workshop*. <https://openreview.net/forum?id=Gc5oq8sr6A3>
- Kleinstreuer, N. C., Karmaus, A., Mansouri, K., Allen, D. G., Fitzpatrick, J. M., & Patlewicz, G. (2018).

- Predictive Models for Acute Oral Systemic Toxicity: A Workshop to Bridge the Gap from Research to Regulation. *Computational Toxicology*, 8(11), 21–24. <https://doi.org/10.1016/j.comtox.2018.08.002>
- Kok, J. N., Boers, E. J. W., Kusters, W. A., van der Putten, P. Der, & Poel, M. (2002). Artificial Intelligence: Definition, Trends, Techniques, and Cases. In *Knowledge for sustainable development: an insight into the Encyclopedia of life support systems* (pp. 1095–1107). UNESCO Publishing.
- Kolmar, S. S., & Grulke, C. M. (2021). The effect of noise on the predictive limit of QSAR models. *Journal of Cheminformatics*, 13(1), 92. <https://doi.org/10.1186/s13321-021-00571-7>
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>
- Lalko, J. F., Kimber, I., Frank Gerberick, G., Foertsch, L. M., Api, A. M., & Dearman, R. J. (2012). The direct peptide reactivity assay: Selectivity of chemical respiratory allergens. *Toxicological Sciences*, 129(2), 421–431. <https://doi.org/10.1093/toxsci/kfs205>
- Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10, 1096. <https://doi.org/10.1038/s41467-019-08987-4>
- Le Roux, N., & Bengio, Y. (2007). Continuous Neural Networks. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2, 404–411. <https://doi.org/10.1109/ETFA.1992.683281>
- Lea, I. A., Gong, H., Paleja, A., Rashid, A., & Fostel, J. (2017). CEBS: A comprehensive annotated database of toxicological data. *Nucleic Acids Research*, 45(D1), D964–D971. <https://doi.org/10.1093/nar/gkw1077>
- Leach, A. R., & Gillet, V. J. (2007). Molecular Descriptors. In *An Introduction to Chemoinformatics* (pp. 53–74). Springer.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2011). Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10), 95–103. <https://doi.org/10.1145/2001269.2001295>
- Liu, J., Patlewicz, G., Williams, A. J., Thomas, R. S., & Shah, I. (2017). Predicting Organ Toxicity Using in Vitro Bioactivity Data and Chemical Structure. *Chemical Research in Toxicology*, 30(11), 2046–2059. <https://doi.org/10.1021/acs.chemrestox.7b00084>
- Loh, W. (2008). Encyclopedia of statistics in quality and reliability. *Choice Reviews Online*, 45(12), 45-6515-45–6515. <https://doi.org/10.5860/CHOICE.45-6515>
- Loyola-Gonzalez, O. (2019). Black-box vs. White-Box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access*, 7, 154096–154113. <https://doi.org/10.1109/ACCESS.2019.2949286>
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30. <https://doi.org/10.48550/arXiv.1705.07874>
- Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., & Svetnik, V. (2015). Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 55(2), 263–274. <https://doi.org/10.1021/ci500747n>

- Maggiora, G. M. (2006). On Outliers and Activity Cliffs - Why QSAR Often Disappoints. *Journal of Chemical Information and Modeling*, 46(4), 1535. <https://doi.org/10.1021/ci060117s>
- Maggiora, G. M., Vogt, M., Stumpfe, D., & Bajorath, J. (2014). Molecular Similarity in Medicinal Chemistry. *Journal of Medicinal Chemistry*, 57(8), 3186–3204. <https://doi.org/10.1021/jm401411z>
- Mannhold, R., Poda, G. I., Ostermann, C., & Tetko, I. V. (2009). Calculation of Molecular Lipophilicity: State-of-the-Art and Comparison of LogP Methods on more than 96,000 Compounds. *Journal of Pharmaceutical Sciences*, 98(3), 861–893. <https://doi.org/10.1002/jps.21494>
- Manson, M. M. (1980). Epoxides--is there a human health problem? *Occupational and Environmental Medicine*, 37(4), 317–336. <https://doi.org/10.1136/oem.37.4.317>
- Marchant, C. A., Briggs, K. A., & Long, A. (2008). In Silico Tools for Sharing Data and Knowledge on Toxicity and Metabolism: Derek for Windows, Meteor, and Vitic. *Toxicology Mechanisms and Methods*, 18(2–3), 177–187. <https://doi.org/10.1080/15376510701857320>
- Martin, E. J., Polyakov, V. R., Zhu, X.-W., Tian, L., Mukherjee, P., & Liu, X. (2019). All-Assay-Max2 pQSAR: Activity Predictions as Accurate as Four-Concentration IC50 s for 8558 Novartis Assays. *Journal of Chemical Information and Modeling*, 59(10), 4450–4459. <https://doi.org/10.1021/acs.jcim.9b00375>
- Mathea, M., Klingspohn, W., & Baumann, K. (2016). Chemoinformatic Classification Methods and their Applicability Domain. *Molecular Informatics*, 35(5), 160–180. <https://doi.org/10.1002/minf.201501019>
- Matsumoto, T., Sakari, M., Okada, M., Yokoyama, A., Takahashi, S., Kouzmenko, A., & Kato, S. (2013). The Androgen Receptor in Health and Disease. *Annual Review of Physiology*, 75, 201–224. <https://doi.org/10.1146/annurev-physiol-030212-183656>
- Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2), 442–451. [https://doi.org/10.1016/0005-2795\(75\)90109-9](https://doi.org/10.1016/0005-2795(75)90109-9)
- Matveieva, M., & Polishchuk, P. (2021). Benchmarks for interpretation of QSAR models. *Journal of Cheminformatics*, 13, 41. <https://doi.org/10.1186/s13321-021-00519-x>
- May, P., & May, E. (1999). Twenty years of p53 research: structural and functional aspects of the p53 protein. *Oncogene*, 18(53), 7621–7636. <https://doi.org/10.1038/sj.onc.1203285>
- Mayr, A., Klambauer, G., Unterthiner, T., & Hochreiter, S. (2016). DeepTox: Toxicity Prediction using Deep Learning. *Frontiers in Environmental Science*, 3, 80. <https://doi.org/10.3389/fenvs.2015.00080>
- McCloskey, K., Taly, A., Monti, F., Brenner, M. P., & Colwell, L. J. (2019). Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences*, 116(24), 11624–11629. <https://doi.org/10.1073/pnas.1820657116>
- Meir, R., & Rätsch, G. (2003). An Introduction to Boosting and Leveraging. In *Advanced Lectures on Machine Learning* (pp. 118–183). Springer. https://doi.org/10.1007/3-540-36434-X_4
- Menke, J., & Koch, O. (2021). Using Domain-Specific Fingerprints Generated Through Neural Networks to Enhance Ligand-Based Virtual Screening. *Journal of Chemical Information and Modeling*, 61(2), 664–675. <https://doi.org/10.1021/acs.jcim.0c01208>
- Moshkov, N., Becker, T., Yang, K., Horvath, P., Caicedo, J. C., Wagner, B. K., Dancik, V., Clemons, P., Singh, S., & Carpenter, A. E. (2022). Predicting compound activity from phenotypic profiles and

- chemical structures. *BioRxiv Preprint*. <https://doi.org/10.1101/2020.12.15.422887>
- Müller, L., Kikuchi, Y., Probst, G., Schechtman, L., Shimada, H., Sofuni, T., & Tweats, D. (1999). ICH-Harmonised guidances on genotoxicity testing of pharmaceuticals: evolution, reasoning and impact. *Mutation Research/Reviews in Mutation Research*, *436*(3), 195–225. [https://doi.org/10.1016/S1383-5742\(99\)00004-6](https://doi.org/10.1016/S1383-5742(99)00004-6)
- Muramatsu, M., & Inoue, S. (2000). Estrogen Receptors: How Do They Control Reproductive and Nonreproductive Functions? *Biochemical and Biophysical Research Communications*, *270*(1), 1–10. <https://doi.org/10.1006/bbrc.2000.2214>
- Myatt, G. J., Ahlberg, E., Akahori, Y., Allen, D., Amberg, A., Anger, L. T., Aptula, A., Auerbach, S., Beilke, L., Bellion, P., Benigni, R., Bercu, J., Booth, E. D., Bower, D., Brigo, A., Burden, N., Cammerer, Z., Cronin, M. T. D., Cross, K. P., ... Hasselgren, C. (2018). In silico toxicology protocols. *Regulatory Toxicology and Pharmacology*, *96*, 1–17. <https://doi.org/10.1016/j.yrtph.2018.04.014>
- National Toxicology Program. (2004). *A National Toxicology Program for the 21 st Century, a roadmap for the future*. https://ntp.niehs.nih.gov/ntp/about_ntp/ntpvision/ntproadmap_508.pdf
- Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. In *Technical Report (CRG-TR-93-1)*, Department of Computer Science, University of Toronto. <https://www.cs.princeton.edu/courses/archive/fall07/cos597C/readings/Neal1993.pdf>
- Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in Neural Information Processing Systems* *29*. <https://doi.org/10.48550/arXiv.1605.09304>
- Nguyen, A., Yosinski, J., & Clune, J. (2019). Understanding Neural Networks via Feature Visualization: A Survey. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning: Vol. 11700 LNCS* (pp. 55–76). Springer. https://doi.org/10.1007/978-3-030-28954-6_4
- Nguyen, T., Nioi, P., & Pickett, C. B. (2009). The Nrf2-Antioxidant Response Element Signaling Pathway and Its Activation by Oxidative Stress. *The Journal of Biological Chemistry*, *284*(20), 13291–13295. <https://doi.org/10.1074/jbc.R900010200>
- Norinder, U., Spjuth, O., & Svensson, F. (2020). Using Predicted Bioactivity Profiles to Improve Predictive Modeling. *Journal of Chemical Information and Modeling*, *60*(6), 2830–2837. <https://doi.org/10.1021/acs.jcim.0c00250>
- NRC. (1983). *Report on Reports: Risk Assessment in the Federal Government: Managing the Process*. The National Academies Press. <https://doi.org/10.1080/00139157.1983.9931232>
- NRC. (1994). Science and Judgment in Risk Assessment. In *Science and Judgment in Risk Assessment*. The National Academies Press. <https://doi.org/10.17226/2125>
- NRC. (2007). *Toxicity testing in the 21st century: A vision and a strategy*. The National Academies Press. <https://doi.org/10.17226/11970>
- OECD. (1997). Test No. 471: Bacterial Reverse Mutation Test. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2001). Test No. 416: Two-Generation Reproduction Toxicity Study. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2004). OECD Principles for the Validation, for Regulatory Purposes, of (Quantitative) Structure-Activity Relationship Models. *OECD Series on Testing and Assessment*.

- <https://www.oecd.org/env/guidance-document-on-the-validation-of-quantitative-structure-activity-relationship-q-sar-models-9789264085442-en.htm>
- OECD. (2008). Test No. 425: Acute Oral Toxicity: Up-and-Down Procedure (UDP). *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2011). Test No. 451: Carcinogenicity Studies. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2017). Test No. 402: Acute Dermal toxicity: Fixed dose procedure. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2018a). Test No. 407: Repeated Dose 28-Day Oral Toxicity Study in Rodents. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2018b). Test No. 408: Repeated Dose 90-Day Oral Toxicity Study in Rodents. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2018c). Test No. 433: Acute Inhalation Toxicity: Fixed Concentration Procedure. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- OECD. (2018d). Test No. 452: Chronic Toxicity Studies. *OECD Publishing*. <https://doi.org/10.1787/20745788>
- Olah, C. (2015). *Understanding LSTM Networks*. Colah's Blog. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature Visualization. *Distill*. <https://doi.org/10.23915/distill.00007>
- Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., & Mordvintsev, A. (2018). The Building Blocks of Interpretability. *Distill*. <https://doi.org/10.23915/distill.00010>
- Omenn, G. S. (1995). Assessing the risk assessment paradigm. *Toxicology*, *102*(1–2), 23–28. [https://doi.org/10.1016/0300-483X\(95\)03034-D](https://doi.org/10.1016/0300-483X(95)03034-D)
- Pak, M., & Kim, S. (2017). A review of deep learning in image recognition. *4th International Conference on Computer Applications and Information Processing Technology*, 1–3. <https://doi.org/10.1109/CAIPT.2017.8320684>
- Paracelsus. (1965). Die dritte Defension wegen des Scribens der neuer Recepte. In *Das Buch Paragranum / Septem Defensiones* (pp. 508–513). Holzinger. <http://www.zeno.org/nid/20009261362>
- Park, S. H., Kang, N., Song, E., Wie, M., Lee, E. A., Hwang, S., Lee, D., Ra, J. S., Park, I. B., Park, J., Kang, S., Park, J. H., Hohng, S., Lee, K., & Myung, K. (2019). ATAD5 promotes replication restart by regulating RAD51 and PCNA in response to replication stress. *Nature Communications*, *10*, 5718. <https://doi.org/10.1038/s41467-019-13667-4>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8024–8035. <https://doi.org/10.48550/arXiv.1912.01703>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhoffer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: machine learning in python. *Journal of*

- Machine Learning Research*, 12, 2825–2830. <https://doi.org/10.48550/arXiv.1201.0490>
- Polishchuk, P. (2017). Interpretation of Quantitative Structure-Activity Relationship Models: Past, Present, and Future. *Journal of Chemical Information and Modeling*, 57(11), 2618–2639. <https://doi.org/10.1021/acs.jcim.7b00274>
- Polishchuk, P. G., Kuz'min, V. E., Artemenko, A. G., & Muratov, E. N. (2013). Universal approach for structural interpretation of QSAR/QSPR models. *Molecular Informatics*, 32(9–10), 843–853. <https://doi.org/10.1002/minf.201300029>
- Poulsen, H. E., Prieme, H., & Loft, S. (1998). Role of oxidative DNA damage in cancer initiation and promotion. *European Journal of Cancer Prevention*, 7(1), 9–16.
- Preuer, K., Klambauer, G., Rippmann, F., Hochreiter, S., & Unterthiner, T. (2019). Interpretable Deep Learning in Drug Discovery. In W. Samek, G. Montavon, A. Vedaldi, L. Hansen, & K. Müller (Eds.), *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (pp. 331–345). Springer. https://doi.org/10.1007/978-3-030-28954-6_18
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106. <https://doi.org/10.1007/bf00116251>
- RDKit: Open-source cheminformatics*. (n.d.). Retrieved May 25, 2021, from <http://www.rdkit.org>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-Agnostic Interpretability of Machine Learning. *ArXiv:160605386v1*. <https://doi.org/10.48550/arXiv.1606.05386>
- Richard, A. M., Judson, R. S., Houck, K. A., Grulke, C. M., Volarath, P., Thillainadarajah, I., Yang, C., Rathman, J., Martin, M. T., Wambaugh, J. F., Knudsen, T. B., Kancherla, J., Mansouri, K., Patlewicz, G., Williams, A. J., Little, S. B., Crofton, K. M., & Thomas, R. S. (2016). ToxCast Chemical Landscape: Paving the Road to 21st Century Toxicology. *Chemical Research in Toxicology*, 29(8), 1225–1251. <https://doi.org/10.1021/acs.chemrestox.6b00135>
- Riniker, S., & Landrum, G. A. (2013). Similarity maps - A visualization strategy for molecular fingerprints and machine-learning methods. *Journal of Cheminformatics*, 5, 43. <https://doi.org/10.1186/1758-2946-5-43>
- Ripley, L. S. (1990). Frameshift Mutation: Determinants of Specificity. *Annual Review of Genetics*, 24, 189–213. https://doi.org/10.1007/978-1-4419-6247-8_13791
- Rittig, J. G., Gao, Q., Dahmen, M., Mitsos, A., & Schweidtmann, A. M. (2022). Graph neural networks for the prediction of molecular structure-property relationships. *ArXiv:2208.04852v1*. <https://doi.org/10.1039/d1dd00037c>
- Rodríguez-Pérez, R., & Bajorath, J. (2020a). Interpretation of compound activity predictions from complex machine learning models using local approximations and Shapley values. *Journal of Medicinal Chemistry*, 63(16), 8761–8777. <https://doi.org/10.1021/acs.jmedchem.9b01101>
- Rodríguez-Pérez, R., & Bajorath, J. (2020b). Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. *Journal of Computer-Aided Molecular Design*, 34, 1013–1026. <https://doi.org/10.1007/s10822-020-00314-0>
- Rogers, D., & Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5), 742–754. <https://doi.org/10.1021/ci100050t>
- Roncaglioni, A., Toropov, A. A., Toropova, A. P., & Benfenati, E. (2013). In silico methods to predict drug toxicity. *Current Opinion in Pharmacology*, 13(5), 802–806. <https://doi.org/10.1016/j.coph.2013.06.001>

- Roy, P. P., Leonard, J. T., & Roy, K. (2008). Exploring the impact of size of training sets for the development of predictive QSAR models. *Chemometrics and Intelligent Laboratory Systems*, *90*(1), 31–42. <https://doi.org/10.1016/j.chemolab.2007.07.004>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. <https://doi.org/10.1038/323533a0>
- Sakamuru, S., Li, X., Attene-Ramos, M. S., Huang, R., Lu, J., Shou, L., Shen, M., Tice, R. R., Austin, C. P., & Xia, M. (2012). Application of a homogenous membrane potential assay to assess mitochondrial function. *Physiol Ogical Genomics*, *44*(9), 495–503. <https://doi.org/10.1152/physiolgenomics.00161.2011>
- Salakhutdinov, R., & Mnih, A. (2008). Bayesian Probabilistic Matrix Factorisation using Markov Chain Monte Carlo. *25th International Conference on Machine Learning*, 880–887. <https://doi.org/https://doi.org/10.1145/1390156.1390267>
- Salakhutdinov, R., & Mnih, A. (2007). Probabilistic Matrix Factorization. *Advances in Neural Information Processing Systems 20*, 1257–1264. <https://papers.nips.cc/paper/2007/hash/d7322ed717dedf1eb4e6e52a37ea7bcd-Abstract.html>
- Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II-Recent progress. *IBM Journal of Research and Development*, *11*(6), 601–617. <https://doi.org/10.1147/rd.116.0601>
- Sanz, F., Pognan, F., Steger-Hartmann, T., Díaz, C., & ETOX. (2017). Legacy data sharing to improve drug safety assessment: The eTOX project. *Nature Reviews Drug Discovery*, *16*(12), 811–812. <https://doi.org/10.1038/nrd.2017.177>
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems 28*, 2503–2511. <https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html>
- Seal, S., Carreras-Puigvert, J., Trapotsi, M.-A., Yang, H., Spjuth, O., & Bender, A. (2022). Integrating cell morphology with gene expression and chemical structure to aid mitochondrial toxicity detection. *Communications Biology*, *5*, 858. <https://doi.org/10.1038/s42003-022-03763-5>
- Shalabi, L. Al, Shaaban, Z., & Kasasbeh, B. (2006). Data Mining: A Preprocessing Engine. *Journal of Computer Science*, *2*(9), 735–739. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.2072&rep=rep1&type=pdf>
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- Shapley, L. S. (1953). A Value for n-Person Games. In *Contributions to the Theory of Games (AM-28)* (Volume II, pp. 307–318). Princeton University Press. <https://doi.org/10.1515/9781400881970-018>
- Sheng, V. S., & Ling, C. X. (2006). Thresholding for making classifiers cost-sensitive. *Proceedings of the 21st National Conference on Artificial Intelligence*, *1*, 476–481. <https://www.aaai.org/Papers/AAAI/2006/AAAI06-076.pdf>
- Sherhod, R., Judson, P. N., Hanser, T., Vessey, J. D., Webb, S. J., & Gillet, V. J. (2014). Emerging pattern mining to aid toxicological knowledge discovery. *Journal of Chemical Information and Modeling*, *54*(7), 1864–1879. <https://doi.org/10.1021/ci5001828>

- Sheridan, R. P. (2019). Interpretation of QSAR Models by Coloring Atoms According to Changes in Predicted Activity: How Robust Is It? *Journal of Chemical Information and Modeling*, 59(4), 1324–1337. <https://doi.org/10.1021/acs.jcim.8b00825>
- Sheridan, R. P. (2022). Stability of Prediction in Production ADMET Models as a Function of Version: Why and When Predictions Change. *Journal of Chemical Information and Modeling*, 62(15), 3477–3485. <https://doi.org/10.1021/acs.jcim.2c00803>
- Sheridan, R. P., Feuston, B. P., Maiorov, V. N., & Kearsley, S. K. (2004). Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. *Journal of Chemical Information and Computer Sciences*, 44(6), 1912–1928. <https://doi.org/10.1021/ci049782w>
- Sheridan, R. P., Karnachi, P., Tudor, M., Xu, Y., Liaw, A., Shah, F., Cheng, A. C., Joshi, E., Glick, M., & Alvarez, J. (2020). Experimental Error, Kurtosis, Activity Cliffs, and Methodology: What Limits the Predictivity of Quantitative Structure–Activity Relationship Models? *Journal of Chemical Information and Modeling*, 60(4), 1969–1982. <https://doi.org/10.1021/acs.jcim.9b01067>
- Sheridan, R. P., Wang, W. M., Liaw, A., Ma, J., & Gifford, E. M. (2016). Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling*, 56(12), 2353–2360. <https://doi.org/10.1021/acs.jcim.6b00591>
- Simm, J., Arany, A., Zakeri, P., Haber, T., Wegner, J. K., Chupakhin, V., Ceulemans, H., & Moreau, Y. (2015). Macau: Scalable Bayesian Multi-relational Factorization with Side Information using MCMC. *ArXiv:1509.04610v2*. <https://doi.org/10.48550/arXiv.1509.04610>
- Simpson, E. R., Clyne, C., Rubin, G., Boon, W. C., Robertson, K., Britt, K., Speed, C., & Jones, M. (2002). Aromatase — a Brief Overview. *Annual Review of Physiology*, 64, 93–127. <https://doi.org/10.1146/annurev.physiol.64.081601.142703>
- Sistare, F. D., Mattes, W. B., & LeCluyse, E. L. (2016). The promise of new technologies to reduce, refine, or replace animal use while reducing risks of drug induced liver injury in pharmaceutical development. *ILAR Journal*, 57(2), 186–211. <https://doi.org/10.1093/ilar/ilw025>
- Snyder, R. D. (2009). An Update on the Genotoxicity and Carcinogenicity of Marketed Pharmaceuticals with Reference to In Silico Predictivity. *Environmental and Molecular Mutagenesis*, 50(6), 435–450. <https://doi.org/10.1002/em.20485>
- Song, B., Zhang, G., Zhu, W., & Liang, Z. (2014). ROC operating point selection for classification of imbalanced data with application to computer-aided polyp detection in CT colonography. *International Journal of Computer Assisted Radiology and Surgery*, 9(1), 79–89. <https://doi.org/10.1007/s11548-013-0913-8>
- Sosnin, S., Karlov, D., Tetko, I. V., & Fedorov, M. V. (2019). Comparative Study of Multitask Toxicity Modeling on a Broad Chemical Space. *Journal of Chemical Information and Modeling*, 59(3), 1062–1072. <https://doi.org/10.1021/acs.jcim.8b00685>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. [https://doi.org/10.1016/0370-2693\(93\)90272-J](https://doi.org/10.1016/0370-2693(93)90272-J)
- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *Proceedings of the 34th International Conference on Machine Learning*, 70, 3319–3328. <https://doi.org/10.48550/arXiv.1703.01365>
- Sushko, I., Salmina, E., Potemkin, V. A., Poda, G., & Tetko, I. V. (2012). ToxAlerts: A web server of structural alerts for toxic chemicals and compounds with potential adverse reactions. *Journal of Chemical Information and Modeling*, 52(8), 2310–2316. <https://doi.org/10.1021/ci300245q>

- Svetnik, V., Liaw, A., Tong, C., Christopher Culberson, J., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Computer Sciences*, 43(6), 1947–1958. <https://doi.org/10.1021/ci034160g>
- Swain, M. (2015). *CIRpy*. <https://github.com/mcs07/CIRpy>
- Swain, M. (2016). *MolVS*. <https://github.com/mcs07/MolVS>
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *ArXiv:1312.6199v3*. <https://doi.org/10.48550/arXiv.1312.6199>
- Tannenbaum, J., & Bennett, B. T. (2015). Russell and Burch's 3Rs then and now: The need for clarity in definition and purpose. *Journal of the American Association for Laboratory Animal Science*, 54(2), 120–132. <https://www.ingentaconnect.com/content/aalas/jaalas/2015/00000054/00000002/art00002>
- Thomas, R. S., Black, M. B., Li, L., Healy, E., Chu, T. M., Bao, W., Andersen, M. E., & Wolfinger, R. D. (2012). A comprehensive statistical analysis of predicting in vivo hazard using high-throughput in vitro screening. *Toxicological Sciences*, 128(2), 398–417. <https://doi.org/10.1093/toxsci/kfs159>
- Thomas, R. S., Paules, R. S., Simeonov, A., Fitzpatrick, S. C., Crofton, K. M., Casey, W. M., & Mendrick, D. L. (2018). The US Federal Tox21 Program: A strategic and operational plan for continued leadership. *Altex*, 35(2), 163–168. <https://doi.org/10.14573/altex.1803011>
- Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B (Methodological)*, 58(1), 267–288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- Tox21 Challenge dataset*. (2014). <https://tripod.nih.gov/tox21/challenge/data.jsp>
- Trapotsi, M.-A., Mervin, L. H., Afzal, A. M., Sturm, N., Engkvist, O., Barrett, I. P., & Bender, A. (2021). Comparison of Chemical Structure and Cell Morphology Information for Multitask Bioactivity Predictions. *Journal of Chemical Information and Modeling*, 61(3), 1444–1456. <https://doi.org/10.1021/acs.jcim.0c00864>
- Tuesuwan, B., & Vongsutilers, V. (2021). Nitrosamine Contamination in Pharmaceuticals: Threat, Impact, and Control. *Journal of Pharmaceutical Sciences*, 110(9), 3118–3128. <https://doi.org/10.1016/j.xphs.2021.04.021>
- Varnek, A., Gaudin, C., Marcou, G., Baskin, I., Pandey, A. K., & Tetko, I. V. (2009). Inductive transfer of knowledge: Application of multi-task learning and Feature Net approaches to model tissue-air partition coefficients. *Journal of Chemical Information and Modeling*, 49(1), 133–144. <https://doi.org/10.1021/ci8002914>
- Wade, D. R., Airy, S. C., & Sinsheimer, J. E. (1978). Mutagenicity of aliphatic epoxides. *Mutation Research/Genetic Toxicology*, 58(2–3), 217–223. [https://doi.org/10.1016/0165-1218\(78\)90012-5](https://doi.org/10.1016/0165-1218(78)90012-5)
- Walker, S. H., & Duncan, D. B. (1967). Estimation of the Probability of an Event as a Function of Several Independent Variables. *Biometrika*, 54(1–2), 167–179. <https://doi.org/10.1093/biomet/54.1-2.167>
- Wang, Y., Bryant, S. H., Cheng, T., Wang, J., Gindulyte, A., Shoemaker, B. A., Thiessen, P. A., He, S., & Zhang, J. (2017). PubChem BioAssay: 2017 update. *Nucleic Acids Research*, 45(D1), D955–D963.

<https://doi.org/10.1093/nar/gkw1118>

- Warr, W. A. (2011). Representation of chemical structures. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(4), 557–579. <https://doi.org/10.1002/wcms.36>
- Watford, S., Pham, L. L., Wignall, J., Shin, R., Martin, M. T., & Friedman, K. P. (2019). ToxRefDB version 2.0: Improved utility for predictive and retrospective toxicology analyses. *Reproductive Toxicology*, 89, 145–158. <https://doi.org/10.1016/j.reprotox.2019.07.012>
- Weinberg, S. (2003). *Good Laboratory Practice Regulations*. Marcel Dekker, Inc.
- Weininger, D. (1988). SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences*, 28(1), 31–36. <https://doi.org/10.1021/ci00057a005>
- Wenzel, J., Matter, H., & Schmidt, F. (2019). Predictive Multitask Deep Neural Network Models for ADME-Tox Properties: Learning from Large Data Sets. *Journal of Chemical Information and Modeling*, 59(3), 1253–1268. <https://doi.org/10.1021/acs.jcim.8b00785>
- Whitehead, T. M., Irwin, B. W. J., Hunt, P., Segall, M. D., & Conduit, G. J. (2019). Imputation of Assay Bioactivity Data Using Deep Learning. *Journal of Chemical Information and Modeling*, 59(3), 1197–1204. <https://doi.org/10.1021/acs.jcim.8b00768>
- Wiener, H. (1947). Structural Determination of Paraffin Boiling Points. *Journal of the American Chemical Society*, 69(1), 17–20. <https://doi.org/10.1021/ja01193a005>
- Wille, R. (1982). Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In *Ordered Sets* (pp. 445–470). Springer. https://doi.org/10.1007/978-94-009-7798-3_15
- Willett, P. (2006). Similarity-based virtual screening using 2D fingerprints. *Drug Discovery Today*, 11(23–24), 1046–1053. <https://doi.org/10.1016/j.drudis.2006.10.005>
- Williams, R. V., Amberg, A., Brigo, A., Coquin, L., Giddings, A., Glowienke, S., Greene, N., Jolly, R., Kemper, R., O’Leary-Steele, C., Parenty, A., Spirkl, H. P., Stalford, S. A., Weiner, S. K., & Wichard, J. (2016). It’s difficult, but important, to make negative predictions. *Regulatory Toxicology and Pharmacology*, 76, 79–86. <https://doi.org/10.1016/j.yrtph.2016.01.008>
- Williams, R. V., DeMarini, D. M., Stankowski, L. F., Escobar, P. A., Zeiger, E., Howe, J., Elespuru, R., & Cross, K. P. (2019). Are all bacterial strains required by OECD mutagenicity test guideline TG471 needed? *Mutation Research - Genetic Toxicology and Environmental Mutagenesis*, 848(December), 503081. <https://doi.org/10.1016/j.mrgentox.2019.503081>
- Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., Assempour, N., Iynkkaran, I., Liu, Y., Maclejewski, A., Gale, N., Wilson, A., Chin, L., Cummings, R., Le, D., ... Wilson, M. (2018). DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Research*, 46(D1), D1074–D1082. <https://doi.org/10.1093/nar/gkx1037>
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., ... Dean, J. (2016). Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv:1609.08144*. <https://doi.org/10.48550/arXiv.1609.08144>
- Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., & Pande, V. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2), 513–530. <https://doi.org/10.1039/c7sc02664a>

- Xu, Y., Ma, J., Liaw, A., Sheridan, R. P., & Svetnik, V. (2017). Demystifying Multitask Deep Neural Networks for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling*, *57*(10), 2490–2504. <https://doi.org/10.1021/acs.jcim.7b00087>
- Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T., Jensen, K., & Barzilay, R. (2019). Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, *59*(8), 3370–3388. <https://doi.org/10.1021/acs.jcim.9b00237>
- Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, *26*, 289–315. <https://doi.org/10.1007/s00365-006-0663-2>
- Young, D., Martin, T., Venkatapathy, R., & Harten, P. (2008). Are the chemical structures in your QSAR correct? *QSAR and Combinatorial Science*, *27*(11–12), 1337–1345. <https://doi.org/10.1002/qsar.200810084>
- Zakarya, D., Boulaamail, A., Larfaoui, E. M., & Lakhlifi, T. (1997). QSARs for toxicity of DDT-type analogs using neural network. *SAR and QSAR in Environmental Research*, *6*(3–4), 183–203. <https://doi.org/10.1080/10629369708033251>
- Zakharov, A. V., Peach, M. L., Sitzmann, M., & Nicklaus, M. C. (2014). QSAR modeling of imbalanced high-throughput screening data in PubChem. *Journal of Chemical Information and Modeling*, *54*(3), 705–712. <https://doi.org/10.1021/ci400737s>
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *67*(2), 301–320. <https://doi.org/10.1111/j.1467-9868.2005.00503.x>