INTERPRETATION, IDENTIFICATION AND REUSE OF MODELS

THEORY AND ALGORITHMS WITH APPLICATIONS IN PREDICTIVE TOXICOLOGY

Anna Maria PALCZEWSKA

A thesis submitted for the degree of

Doctor of Philosophy

School of Electrical Engineering and Computer Science

University of Bradford

2014

Anna M. Palczewska
Interpretation, Identification and Reuse of Models

Keywords: Model interpretation, Model identification, Model governance, Reuse of models, Predictive toxicology, Random forest model, Feature contributions, Pareto optimality

Abstract

This thesis is concerned with developing methodologies that enable existing models to be effectively reused. Results of this thesis are presented in the framework of Quantitative Structural-Activity Relationship (QSAR) models, but their application is much more general. QSAR models relate chemical structures with their biological, chemical or environmental activity. There are many applications that offer an environment to build and store predictive models. Unfortunately, they do not provide advanced functionalities that allow for efficient model selection and for interpretation of model predictions for new data. This thesis aims to address these issues and proposes methodologies for dealing with three research problems: model governance (management), model identification (selection), and interpretation of model predictions. The combination of these methodologies can be employed to build more efficient systems for model reuse in QSAR modelling and other areas.

The first part of this study investigates toxicity data and model formats and reviews some of the existing toxicity systems in the context of model development and reuse. Based on the findings of this review and the principles of data governance, a novel concept of model governance is defined. Model governance comprises model representation and model governance processes. These processes are designed and presented in the context of model management. As an application, minimum information requirements and an XML representation for QSAR models are proposed.

Once a collection of validated, accepted and well annotated models is available within a model governance framework, they can be applied for new data. It may happen that there is more than one model available for the same endpoint. Which one to chose? The second part of this thesis proposes a theoretical framework and algorithms that enable automated identification of the most reliable model for new data from the collection of existing models. The main idea is based on partitioning of the search space into groups and assigning a single model to each group. The construction of this partitioning is difficult because it is a bi-criteria problem. The main contribution in this part is the application of Pareto points for the search space partition. The proposed methodology is applied to three endpoints in chemoinformatics and predictive toxicology.

After having identified a model for the new data, we would like to know how the model obtained its prediction and how trustworthy it is. An interpretation of model predictions is straightforward for linear models thanks to the availability of model parameters and their statistical significance. For non linear models this information can be hidden inside the model structure. This thesis proposes an approach for interpretation of a random forest classification model. This approach allows for the determination of the influence (called feature contribution) of each variable on the model prediction for an individual data. In this part, there are three methods proposed that allow analysis of feature contributions. Such analysis might lead to the discovery of new patterns that represent a standard behaviour of the model and allow additional assessment of the model reliability for new data. The application of these methods to two standard benchmark datasets from the UCI machine learning repository shows a great potential of this methodology. The algorithm for calculating feature contributions has been implemented and is available as an R package called `rfFC`.

This work has been funded by BBSRC and Syngenta (International Research Centre at Jealott's Hill, Bracknell, UK).

Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Anna Palczewska

Acknowledgements

I would like to thank my husband who has been supportive and understanding over the years of my PhD studies. I would like to express my gratitude for his advice in research work.

Special thanks go to my supervisors: Prof. Daniel Neagu, Mick Ridley, and Kim Travis for all their support in my professional development, advice and guidance during the entire period of my PhD studies. I would like to thank them for proofreading my thesis.

Dr. Richard Marchese Robinson is thanked for the professional collaboration, support and advice, and many enjoyable discussions inside and outside of work. I also would like to thank him for testing and debugging the rfFC package.

I would like to thank Dr. Xin Fu for the professional collaboration on data and model governance framework, and implementation of MADFARM prototype.

John Delaney and Dr. Nathan Kidley from the Computational Chemistry Department, Syngenta, are thanked for the professional collaboration, consultation and support in chemistry related problems and, finally, for giving me access to the in-house logP dataset.

I would like to thank Prof. Mark Cronin and Dr. Steve Enoch for providing access to the IGC50 for *Tetraphymenra poriformfism* dataset.

Many thanks go to my colleagues Dr. Longzhi Yang and Dr. Paul Trandle with whom I had a pleasure to collaborate during my project.

Finally, I would like to acknowledge the support and funding of Syngenta and BBSRC as part of the Industrial CASE scheme.

Publications and Presentations

Books and Journals

- Anna Palczewska, Jan Palczewski, Richard Marchese Robinson, Daniel Neagu. Interpreting random forest classification models using a feature contribution method. *in Integration of Reusable Systems*, ser. Advances in Intelligent and Soft Computing, T. Bouabana-Tebibel and S. H. Rubin, Eds. Springer International Publishing, 2014, vol. 263, pp. 193–218

- Anna Palczewska, Daniel Neagu, Mick Ridley. Using Pareto points for model identification in predictive toxicology. *Journal of Cheminformatics*, vol. 5, no. 1, p. 16, 2013

- Anna Palczewska, Xin Fu, Paul Trundle, Longzhi Yang, Daniel Neagu, Mick Ridley, Kim Travis. Towards model governance in predictive toxicology. *International Journal of Information Management*, vol. 33, no. 3, pp. 567–582, 2013

- Xin Fu, Anna Wojak (Palczewska), Daniel Neagu, Mick Ridley, Kim Travis. Data governance in predictive toxicology: A review. *Journal of Cheminformatics*, vol. 3, no. 1, p. 24, 2011

Conference proceedings

- Anna Palczewska, Jan Palczewski, Richard M. Robinson, Daniel Neagu. Interpreting random forest models using a feature contribution method. *In Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference*, pp.112–119, 2013.

- Anna Wojak (Palczewska), Daniel Neagu, Mick Ridley. Double min-score (DMS) algorithm for automated model selection in predictive toxicology. *In United Kingdom Workshop in Computational Intelligence (UKCI 2011)*, pp.150–156, 2011

Presentation and posters

- Presentation at Joint Biological Sciences and Product Safety Research Collaborations Review Event, *Using Pareto Points for QSAR Model Identification in Predictive Toxicology*, 2–3 Sep. 2013, Syngenta, Bracknell, UK
- Presentation at IEEE 14th International Conference on Information Reuse and Integration, *Interpreting random forest models using a feature contribution method*, 14–16 Aug. 2013, San Francisco, USA
- Presentation at Seminar of Artificial Intelligence Research Group, *Using Pareto Points for QSAR Model Identification in Predictive Toxicology*, 11 Jun. 2013, University of Bradford, UK
- Presentation at Seminar of Artificial Intelligence Research Group, *Latex for Scientific Writing – A Hands-on Introduction to Latex*, 16 Nov. 2012, University of Bradford, UK
- Poster at SEURAT Summer School, *A Computer Tool Assisting the Quality Control of Toxicity Data in Chemical Presentation and Physicochemical Descriptors*, 4–8 Jun. 2012, Lisbon, Portugal
- Presentation at Fika Seminar, *Searching large graph databases*, 3 Feb. 2012, University of Bradford, UK
- Presentation at Fika Seminar, *Single-source shortest path problems: Dijkstra and A* algorithms*, 27 Oct. 2011 University of Bradford, UK
- Poster at The Syngenta Biological Sciences and Product Safety Research Days, *Towards Data and Model Governance in Predictive Toxicology*, 19–20 Sep. 2011, Syngenta, Bracknell, UK

- Presentation at XI Annual Workshop on Computational Intelligence UKCI, *Double Min-Score (DMS) Algorithm for Automated Model Selection in Predictive Toxicology*, 7–9 Sep. 2011, Manchester, UK
- Presentation at Seminar of Artificial Intelligence Research Group, *Double Min-Score (DMS) Algorithm for Automated Model Selection in Predictive Toxicology*, 6 Sep. 2011, University of Bradford, UK

Software

- `rfFC` - random forest Feature Contributions R package

Contents

List of Figures

List of Tables

Glossary

Greek Symbols

Γ	Pareto Set
$\amalg\Gamma$	Initial Pareto Set

Other Symbols

q^2	Predictive Squared Correlation Coefficient
r^2	Squared Correlation Coefficient

Acronyms

AD	Applicability Domain
APMI	Average Pareto Model Identification
BCW	Breast Cancer Wisconsin Dataset
CAS	Chemical Abstract Service
CPMI	Centroid Pareto Model Identification
CV	Cross-Validation
DMS	Double Min Score Algorithm
IGC50	50% Growth Inhibition Concentration

IMS	Information Management System
InChI	IUPAC International Chemical Identifier
IT	Information Technology
LMO	Leave-Many-Out
LOO	Leave-One-Out
MAE	Mean Absolute Error
MIAQMR	Minimum Information about a QSAR Model Representation
NPN	Non-Polar Narcosis QSAR Model
OCHEM	Online Chemical Modeling Environment
OECD	Organisation for Economic Co-operation and Development
PMML	Predictive Model Markup Language
PN	Polar Narcosis QSAR Model
QMRF	QSAR Model Reporting Format
QSAR	Quantitative StructureActivity Relationship
REACH	Registration, Evaluation, Authorisation & Restriction of CHemicals
RF	Random Forest
RMSE	Root Mean Square Error
SMILE	Simplified Molecular Input Line Entry Specification
TPT	*Tetrahymena pyriformis* Toxicity
UCI	University of California Irvine

“...essentially, all models are wrong,
but some are useful...”

– George E. P. Box

Chapter 1

Introduction

This thesis is concerned with methodological problems arising in model identification, interpretation and reuse. Proposed solutions are applied in QSAR modelling framework in predictive toxicology.

1.1 Background and Motivation

A phenomenon of fast growing data has been observed in the last decade. Data representation, integration and storage have turned out to be big challenges and attracted great interest in order to reuse existing information. Garzotto et al. [31] defined the term "reuse" as usage of existing data objects in different contexts and for different purposes. According to the authors, reuse is also a technology that proposes new methods for optimizing data representation, develops strategies and algorithms for applying integration approaches in novel domains, and develops models that can be used for decision-making processes in various application domains.

This thesis focuses on the third aspect of information reuse, that is models. Together with the rapidly increasing amount of data, the number and variety of models has increased dramatically thanks to user-friendly machine learning and data mining tools. This has happened especially in domains such as: medicine, life sciences, agriculture, etc., where models are built based on existing experimental data and are used to make predictions for new data. There is a need to build a framework for efficient model management. This thesis is concerned with developing methodologies that en-

able existing models to be effectively reused. Results of this thesis are presented in the framework of Quantitative Structural-Activity Relationship (QSAR) models, but their application is much more general.

Predictive toxicology is concerned with the development of models that are able to predict the toxicity of chemicals [41]. A large number of publicly available databases, development of computational chemistry and biology, and rapidly increasing number of *in-vitro* assays have contributed to the development of more accurate predictive models. These models are important for many governmental, academic and business organisations because they enable:

- fast evaluation of chemical toxicity,
- earlier rejection of chemicals that may fail at the chemical development phase,
- reduction in the number of animal tests,
- reduction in the cost of development of new chemical compounds.

The increasing interest in model reuse have also been driven by current requirements of Registration, Evaluation, Authorisation & Restriction of CHemicals (REACH) [94] legislation. This regulatory body accepts chemicals that were tested using inter alia *in silico* modelling (predictive models or virtual screening techniques) when models were properly validated and documented. Models must also be statistically significant and robust and have their application boundary defined.

One of the most known and accepted *in-silico* methods, which are used in this thesis, are Quantitative Structure–Activity Relationship (QSAR) models. They are mathematical models which relate a biological activity of chemicals to their structural, and physiochemical properties. According to REACH Regulation Annex XI [93], results of QSAR modelling may be used instead of animal testing when: QSAR model was scientifically validated, substance falls within its applicability domain, results are adequate for the purpose of classification and labelling and/or risk assessment, and finally, a documentation of the applied method is provided.

The fact that models have become considered as an alternative to animal testing caused a rapid development of *in silico* methods for screening chemical compounds and a development of model validation techniques in order to prove model reliability

and predictivity. Existing solutions focus on the toxicity data integration and the development of platforms/systems that provide methods/tools to build high quality models. Gramatica [34] proposed methods for QSAR model validation which have become fundamentals for the current Organisation for Economic Co-operation and Development (OECD) QSAR validation principles [77]. Tropsha [113] described a workflow for QSAR model development that includes these principles. For example, following the good practice of model development and model validation principles, Hardy et al. [39] introduced interoperable, standard-based framework (OpenTox) for the support of predictive toxicology data management, algorithms, modelling, validation and reporting. Sushko at al. [105] proposed the Online Chemical Modeling Environment (OCHEM) – a web-based platform that aims to automate and simplify typical steps required for QSAR modeling. Both platforms (OpenTox and OCHEM) consist of two major subsystems: the database of experimental measurements and the modeling frameworks.

The above examples show an interest in QSAR model development, aggregation and utilisation. Whilst the above mentioned platforms provide excellent modelling frameworks and are hosts of models that were generated within such frameworks, model reuse and results validation is left to users. For a collection of models for the same endpoint, a user is required to analyse and compare models with respect to their input variables, applicability domain and accuracy, to *identify* the most suitable one for new data. This is a manual process and requires a lot of effort and knowledge. Model selection can be aided by studying distances between models [49, 73, 111] and their applicability domains [53]. The decision if new data fall inside the applicability domain is based on the average distance of query data to the data from the applicability domain. For models, where new data fall into their applicability domains, the decision which one should be used is again difficult. For example, there is also no information about model predictivity in various areas of the applicability domain. This is why there is a need to develop methods that combine: model predictivity and applicability domain in order to provide a framework for automated model identification.

The second element of model reuse is an *interpretation* of model predictions for a given data record and each variable used by the model. This interpretation can help to understand how the model makes its prediction and how reliable the prediction is for new data. It might increase the trust in the model. Such interpretation is straightforward for models where there is access to the model variables and parameters. For non-

linear or ”black box“ models such information is hidden within the model structure. The extraction of this information is a challenging problem that has recently begun attracting attention of researchers. For example, see Carlsson et al. [12] who develop a method for local interpretation of Support Vector Machine (SVM) and Random Forest models, and Kiz'min et al. [67] who show how to extract feature contributions in random forest regression models. Interpretation of model prediction is considered particularly valuable in such domains as chemoinformatics, bioinformatics or predictive toxicology [95]. The knowledge of a chemical fragment or chemical properties that contribute to the adverse effect of that chemical compound can support drug design processes.

1.2 Problem Description

The previous section provides examples of toxicity systems that support development of good quality models. Current studies focus on providing user-friendly environments for model building, model validation and reporting. Models are collected in databases for further reuse. Due to an increasing amount of experimental data, we may find more than one model for the same endpoint. In this case a decision which one should be used is not straightforward. The lack of automated methods that allow analysis of models and their selection may discourage potential users. They may prefer to generate a new model, which they will trust, instead of using an existing one. This situation is mostly, but not only, limited to local models that can not become global tools because they were developed for a particular group of chemicals. Such models, even if they can contribute to the fast evaluation of new chemicals, may be forgotten or lost. To address this problem, our main research question is:

How can existing models be efficiently reused?

This thesis aims to answer the above question. The answer comprises three research directions.

The first one, which we call *model governance*, covers the area of model object representation and model management. Developed models must be properly annotated and validated prior to their further usage. There should be defined processes that

allow continuous model evaluation (validation and reporting regarding to the organizational and authorities requirements). These processes should ensure model quality and security in their future reuse. Model representation should be as much as possible transparent which may allow model exchange initiatives across various organizations.

The second research direction, which we call *model identification*, covers an area of problems related to model selection from a collection of existing models. In cases when there is a number of models for the same endpoint, with overlapping or disjoint applicability domains, such selection is not trivial. Models must be compared and standard techniques for model selection can not be applied (especially for models with different applicability domains). Incorporating applicability domains in the model comparison can also be difficult because some parameters may not be available.

The last research direction, *model interpretation*, covers the analysis of model predictions. This includes a discovery of mechanisms that lead a model to make a particular decision. This is straightforward for linear models, where there is an easy access to model parameters and their statistical significance. For non-linear and so called "black-box" models, this information is hidden inside the model structure and, hence, it is not directly available. Special methods must be designed to enable model interpretation.

1.2.1 Thesis Aims

This thesis addresses the above discussed issues and proposes a theoretical framework and algorithms for each of above presented research directions. This includes definition of the framework for data and model storage, methodology for automated model identification and methods for model interpretation. In respect to each of research directions, this thesis:

1. investigates toxicity data and model formats, and reviews some of the existing toxicity systems in the context of model reuse,

2. defines a new concept of model governance, model governance processes and proposes a theoretical framework for data and model management. This also includes a proposition of a model object representation format,

3. proposes a theory and original algorithms for the model identification problem

and applies them to real toxicity data and models for various endpoints to demonstrate their advantage and potential in predictive toxicology,

4. introduces an algorithm for interpreting random forest models which is an extension of feature contributions method of Kuz'min et al. [67] and implements it into an R package,

5. proposes original methods for the analysis of feature contributions and tests them using classification benchmark datasets.

Although this research mostly focuses on the application of novel approaches in predictive toxicology, the methodological aspect of this work provides theory and algorithms that can be implemented in any domain that accepts data-driven modelling.

1.2.2 Methodology and Data

This thesis is concerned with methodological problems arising in model reuse. To achieve the research aims, the following existing methodologies were used:

- principles of data governance [20] to inform the design of model governance processes,

- Pareto optimality approach [23] to solve the bi-criteria problem of model identification. The decision on which model can be the most suitable for new data is a trade-off between the similarity of this data to a group of elements in the search space and the accuracy of the model for these elements,

- the random forest method proposed by Breiman [6] as a basis to develop model interpretation algorithm,

- the clustering algorithm `k-means` [40] that is used in analysis of feature contributions.

This research was mainly motivated by the need of model reuse in predictive toxicology and the solutions were presented in the area for QSAR modelling. In this thesis the following endpoints and models were used:

- IGC50 for Tetrahymena poriformism (TETRATOX data [99]) downloaded from [45], with two QSAR mode of action models (polar narcosis and non polar narcosis) reported in the JRC QSAR Model Database [54],
- Measured LogP Syngenta's in-house dataset, with Syngenta's in-house model for CLogP. Two existing tools were also used to calculate LogP: KOWWIN from EPI Suite [28] and MLogP from Dragon software [109],
- Chemical persistence in soil, which was prepared in collaboration with Syngenta, and a number of models that were obtained during the competition published and run by Syngenta at IDEACONNECTION [42],
- Two datasets: Breast Cancer Wisconsin and Iris downloaded from UCI Machine Learning Repository [115, 116] and models developed using random forest method.

1.3 Thesis Structure

This thesis is arranged into six chapters and one appendix:

- Chapter 2 – presents a literature review. It includes a review of toxicity data formats, practices in QSAR model development process, a review of current validation techniques and a critical review of some toxicity platforms. Elements of the review presented in this chapter can be found in:
  - Anna Palczewska, Xin Fu, Paul Trundle, Longzhi Yang, Daniel Neagu, Mick Ridley, Kim Travis. Towards model governance in predictive toxicology. *International Journal of Information Management*, vol. 33. no. 3, pp. 567–582, 2013
  - Xin Fu, Anna Wojak (Palczewska), Daniel Neagu, Mick Ridley, Kim Travis. Data governance in predictive toxicology: A review. *Journal of Cheminformatics*, vol. 3, no.1, p. 24, 2011
- Chapter 3 – introduces a novel concept of model governance. In this chapter, the term of model governance is formulated and three model governance processes

are defined: model evaluation, model validation, and model control. A conceptual framework for data and model governance is established and introduced. To represent a model as an object, six rules were introduced to define minimum information about QSAR model representation that are required for model governance. An XML schema based on the proposed rules was defined. The Model and Data Farm (MADFARM) platform was developed in collaboration with Syngenta as a proof of concept of the proposed theoretical framework of model governance. The model governance framework presented in this chapter was published in

  - Anna Palczewska, Xin Fu, Paul Trundle, Longzhi Yang, Daniel Neagu, Mick Ridley, Kim Travis. Towards model governance in predictive toxicology. *International Journal of Information Management*, vol. 33. no. 3, pp. 567–582, 2013

- Chapter 4 – proposes the framework for automated model identification for new data. This is a theoretical framework that defines a search space (chemical space) and its partitioning model. This model divides a search space into disjoint groups and assigns the most predictive model to each group. To construct such a partition, three approaches were proposed here. One based on the nearest neighbourhood called the Double Min Score algorithm (DMS) and two based on Pareto optimality which was used to define the Pareto Neighbourhood: Average Pareto Model Identification (APMI) and Centroid Pareto Model Identification (CPMI) algorithms. This is a new approach in model management and mining. This theoretical framework together with proposed algorithms were published in:

  - Anna Wojak (Palczewska), Daniel Neagu, Mick Ridley. Double min-score (DMS) algorithm for automated model selection in predictive toxicology. *In United Kingdom Workshop in Computational Intelligence (UKCI 2011)*, pp.150–156, 2011
  - Anna Palczewska, Daniel Neagu, Mick Ridley. Using Pareto points for model identification in predictive toxicology. *Journal of Cheminformatics*, vol. 5, no.1, p.16, 2013

  Three endpoints are used to validate the proposed methodology: IGC50 for

Tetrahymena pyriformis, LogP for Syngenta dataset, and chemical persistence in soil. Models for prediction of the last endpoint were collected from a competition organised by Syngenta.

- Chapter 5 – extends to classification problems the feature contribution method, originally proposed for the interpretation of random forest regression models [67]. Feature contributions explain how a model makes decisions for a given instance. This approach uses a probabilistic interpretation of the random forest prediction. In this chapter, three novel methods for analysing feature contributions: median, clustering and log-likelihood were also introduced. These methods have been tested using general classification benchmark datasets. The results were published in:

  - Anna Palczewska, Jan Palczewski, Richard M. Robinson, Daniel Neagu. Interpreting random forest models using a feature contribution method. *In Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference*, pp.112–119, 2013.
  - Anna Palczewska, Jan Palczewski, Richard Marchese Robinson, Daniel Neagu. Interpreting random forest classification models using a feature contribution method. *in Integration of Reusable Systems*, ser. Advances in Intelligent and Soft Computing, T. Bouabana-Tebibel and S. H. Rubin, Eds. Springer International Publishing, 2014, vol. 263, pp. 193–218

- Chapter 6 – presents conclusions drawn from the work presented in preceding chapters and offers suggestions for future research.

- Appendix A – includes documentation for the random forest Feature Contributions (`rfFC`) package developed in the project. It implements the method presented in Chapter 5.

Chapter 2

Data and Models in Predictive Toxicology

Reuse of toxicity information facilitates the reduction in the number of animal testing in domains such as: pharmacology, cosmetics or agriculture. This is why data curation and integration have become recent challenges and enjoy a lot of scientific interest. High quality toxicity information is required to build accurate predictive models for toxicity values. This chapter presents various toxicity data representations; data curation and data integration techniques; a role and process of predicting modelling; and a brief review of existing integrated toxicity platforms. Parts of this review were published in [30] and [83].

2.1 Predictive Toxicology

Toxicology is defined as the study of adverse effects of chemicals on biological systems such as a cell, tissue, organ or an entire organism [41]. It is the study of symptoms, mechanisms, treatments and detection of chemical toxicity. A large number of *in-vivo – in-vitro* tests is required in order to explain these toxic effects. The analysis of changes in molecular expression, toxicological parameters, and integrating response data are used to describing functioning organisms [119]. This knowledge is applied in safety evaluation and risk assessment to protect human and environmental health.

Predictive toxicology provides various computational methods to predict the po-

tential impact of a chemical compound on human or environmental health. Various chemical, biological and toxicological data is combined into sets and used to build predictive models. These models identify parameters that are relevant for a particular toxic effect [41]. When biochemical mechanisms are known, the set of parameters can be pre-defined, reducing the complexity of the model development process. Unfortunately, biochemical mechanisms are often unknown. In such case, methods for parameter selection should be used to limit the number of suitable parameters that explain a given toxic effect.

Expert systems (often known as rule-driven) and data-driven methods are two main strategies in building predictive models. The expert systems are computer systems that mimic the decision-making ability of a human expert [50]. The most popular rule-based system for the prediction of toxicity (genotoxicity, carcinogenicity or skin sensitization) is DEREK [70]. This system has been developed by LHASA [69] - a not-for-profit company based in Leeds. The rules are being developed by the expert toxicologists who work for Lhasa or various other experts that use this system. Data-driven methods based on the development of predictive models from experimental toxicity data. There are various methods available (statistical and machine learning methods) and the decision as to which method should be used depends on the application and complexity of a problem. These methods offer the potential for a fast, rigorous and reliable evaluation of untested chemical toxicity. They are also used in the prioritization of chemical compounds [41] for physical toxicity assays.

A main challenge in predictive toxicology is information reuse. This includes: data integration and model aggregation to provide an interoperable, flexible and transparent framework for automated modelling and testing. The development of good predictive models depends on quality of experimental data [59]. Not complete or not relevant data lead to the generation of inaccurate models. Due to the presence of various toxicity data representations distributed across many organisations, data integration is a difficult problem. It involves data curation and a data quality assessment to ensure the accuracy of collected information. In literature, one can find ongoing projects aiming to provide integrated platforms for toxicity data exchange across several institutions such as research, business or governmental laboratories. It is an important step to propose standards in toxicity data representation. This will increase the reuse of collected information and facilitate collaboration between various institutions. Model

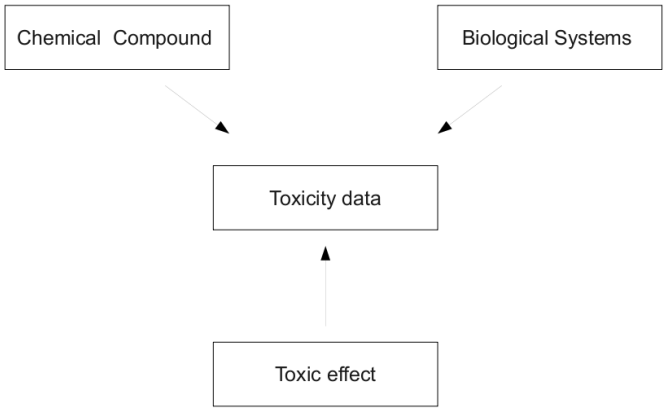


Figure 2.1: Toxicity data classification.

aggregation has become a valuable technique in toxicity estimation since models have begun to be considered as an alternative for animal testing. The standardisation of the model representation and management policies will lead to an increase in the number of model repositories.

2.2 Toxicity Data Representation

Toxicity data is a combination of chemical, biological and toxicological information (see Figure 2.1). It is used as an input to data-driven approaches to build predictive methods, and to validate these methods by comparing results of predictions with the real measurements. The knowledge of data type and representation leads to a choice of adequate modelling techniques which results in development of fast and efficient predictive methods. In this section, various standards used to represent a chemical compound and biological response are discussed.

2.2.1 Chemical Information

A two-dimensional structural representation of a chemical compound and its semantic nomenclature had been established before the end of the nineteenth century [32]. In the first half of the twentieth century fragment-coding systems were developed to identify sets of sub-structural fragments presented in a molecule. The development of computer

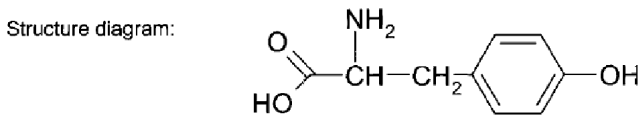
systems and computational chemistry required the presence of more sophisticated and machine-readable representations of a chemical compound.

There are many ways to represent a chemical compound including: names, formula, line-symbol notation, molecular representation, physical and chemical properties and fingerprint. Names and indexes like the CAS number are used to identify a query chemical compound and enable fast information (chemical properties) retrieval from large databases. The CAS number is assigned by Chemical Abstract Service [14] to all publicly available chemicals. It does not relate any chemical properties to structures. Its numerical value is assigned in sequential, increasing order when a substance is added into the CAS REGISTRY database. It is a unique numerical identifier with the following format: XXXXXXX-XX-X. The first group may contain up to seven digits, the second group contains only two digits and the last consists of one digit called checksum. This number allow for a quick check if a query chemical compound identifier is correct.

The second group of chemical representations uses their molecular structure. Currently, 1-D, 2-D and 3-D molecular representations are known and there is still a strong interest in deriving new molecular representations [41].

- 1-D representation is a linear string notation of a chemical compound formula (see Figure 2.2). The most popular formats are:
  - SMILES language (Simplified Molecular Input Line Entry Specification),
  - WLN (Wiswesser Line Notation),
  - InChI (IUPAC International Chemical Identifier),
  - ROSDAL (Representation Of Structure Diagram Arranged Linearly).

  Over last few years SMILES and InChI have become the most used line notations. SMILES are string notations decoding the molecular structure. They are obtained by printing the symbol nodes encountered in a depth-first tree traversal of a chemical graph [32]. Often, SMILES are not unique. A chemical compound can have a few SMILES notation caused by using different starting points in the traversal procedure. InChI keys describe chemical substances using information layers including: atoms and their bond connectivity, tautomeric information, isotope information, stereochemistry, and electronic charge information [74]. In



Trivial name:	tyrosine
Systematic names:	β-(*p*-hydroxyphenyl)alanine
	α-amino-*p*-hydroxyhydrocinnamic acid
WLN	QVYZ1R DQ
SMILES	OC(=O)C(N)CC1=CC=C(O)C=C1
ROSDAL	1O-2=3O,2-4-5N,4-6-7=-12-7,10-13O
SLN	OHC(=O)CH(NH2)CH2C[1]=CHCH=C(OH)CH=CH@1

Figure 2.2: Names and line notations for a tyrosine structure diagram [32].

contrast to widely used CAS registry numbers, SMILES and InChI are computed from the structural information and they are readable by experts. They are also are well suited for chemical compound searching and retrieval from large chemical databases.

- 2-D representation includes *connection tables* (see Figure 2.3). It is a graph representation $G = (V, E)$ where molecular atoms define a set of graph nodes V and bonds represent a set of edges E. The connection table consists of three parts. The first line in the table, called the header block, contains: molecule name and file origin counts of atoms and bonds. The second part, called the atoms block, includes: one line per atom and specifies 2D coordinates, atom symbol, isotope, charge and stereo code. And the last part, called the bonds block, contains: one line per bond (each bond shown once) specifies row numbers for atoms, and codes for bond type, bond stereochemistry. The molecular graph representation is used for queries in similarity searching and especially in sub-structure searching.

- 3-D representation contains the graph representation extended by 3D coordinates, molecular surface or conformations information. This representation is used in searching pharmacophoric patterns [41].

```
24 24  0     1  0  0  0  0  0999 V2000
   4.5981    2.8450    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   2.8660   -3.1550    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   2.8660    2.8450    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   4.5981    0.8450    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   2.8660    0.8450    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   3.7320    1.3450    0.0000 C   0  0  3  0  0  0  0  0  0  0  0  0
   2.8660   -0.1550    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   2.0000   -0.6550    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   3.7320   -0.6550    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   3.7320    2.3450    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  …

 1 10  1  0  0  0  0
 1 23  1  0  0  0  0
 2 13  1  0  0  0  0
 2 24  1  0  0  0  0
 3 10  2  0  0  0  0
 4  6  1  0  0  0  0
 4 19  1  0  0  0  0
…..
```

Figure 2.3: A fragment of tyrosine - connection table representation [88].

Another representation format of a chemical compound is a fragment-based code (index) of its molecule structure. Presence or absence of a certain structural fragment is encoded in a binary vector called a *fingerprint* [32]. This representation is widely used in substructure searching. There are many similarity metrics such as Hamming distance, Dice coefficient, Euclidean distance to compare two binary vectors to test their similarity. Various measures have been studied by the Sheffield research group in the context of chemical similarity, and the results are presented in [121]. The most common similarity measure between two molecules A and B is Tanimoto coefficient defined as follows:

$$T_{AB} = \frac{c}{a + b - c} \tag{2.1}$$

where a and b are numbers of bits set on in the molecules A and B, respectively, and c is the number of bits set on in both molecules. Comparing with atom-by-atom searching (for a molecular representation), the advantage of using fingerprints is the faster search time for large databases. Unfortunately, the fragment code is not unique. Several structures can have the same fingerprint representation. This is why, the *circular fingerprint* has become very popular. It can be used to generate patterns of various diameter for a molecule. The diameter represents the size of the fragment used to be encoded. By increasing the diameter, one can enrich the information about the molecules. However,

this will also increase the overhead of balancing the fingerprint size and reducing the bit clashes. Nevertheless, fingerprint is a very useful tool to filter a large dataset to find frequently repeated structural patterns.

The last group of chemical representation is called *descriptors*. There are various physical and chemical properties of a chemical compound calculated from its molecular representation. There are four types of descriptors: topological, geometrical, electronic and hybrid. Topological descriptors are derived from connection tables and include information about a number of atoms, bonds and substructures. They includes also topological indices, such as connectivity or kappa indices. From 3D molecular representation, the geometrical descriptors are calculated. They include information such as principal moments of inertia, molecular volume or cross-sectional areas. Electronic descriptors include LUMO and HOMO energies, bond orders or partial atomic charges. Various combinations of the above described descriptor types are called hybrid descriptors and they are mostly used in the modelling of quantitative structure-activity relationships. The most comprehensive collection of molecular descriptors with detailed review is presented by Todeschini et al. [112]. All descriptors are listed with their definition, symbols and labels, formulas, some numerical examples, data and molecular graphs, while numerous figures and tables aid comprehension of the definitions

2.2.2 Biological Information

Biological information is derived from *in-vitro* and *in-vivo* assays. *In-vivo* data refers to information collected from experiments or studies done on living organisms. This involves animal testing and clinical trials. The development of molecular biology contributed to an increasing number of *in-vitro* tests. It is focused on organs, tissues, cells, cellular components, proteins, and biomolecules. *In-vitro* research is more suitable for the deduction of biological changes in the organism (mechanisms of action) and due to its relatively low cost, it is competitive to *in-vivo* study [61]. Unfortunately, direct extrapolation from *in-vitro* to *in-vivo* systems can give misleading results. It is related to the various conditions which are presented during experiments. Successful *in-vitro* tests should usually be followed by *in-vivo* studies.

Biological data is derived from transcriptomics, proteomics and metabonomics

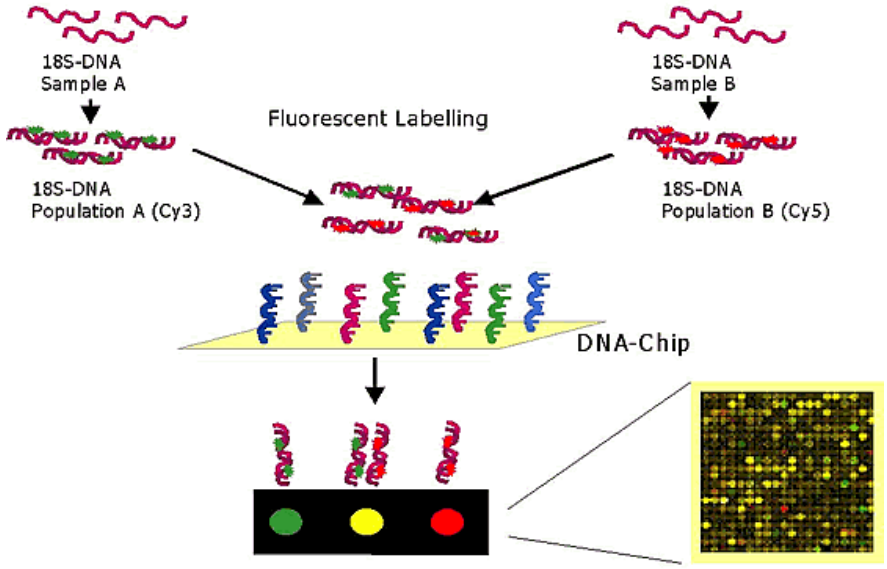


Figure 2.4: Microarray example of biological data representation [127].

studies. They are often used to study mechanisms of genotoxicity and carcinogenicity. Various technologies such as cDNA, mRNA microarrays, protein chips and NMR are used in discovery of toxicant pathways and mode of action that may cause a toxic effect [119]. The example of the microarray technique for gene profiling is presented in Figure 2.4. DNA chips are used to hybridize two DNA strands. The results are scanned and stored as images. Further, these images are normalised and analysed in order to explain molecular changes.

To provide standards in experiment reporting, the Minimum Information About Microarray Experiment (MIAME) format, was proposed in [5, 25]. It provides a set of rules that contribute to a standardisation of biological experiments. An efficient description of experiments allows for its sufficient interpretation, replication and comparison with other similar experiments. This process involves: automated data mining techniques and data analysis, standards in experiment descriptions and development of query structures (see Figure 2.5).

According to MIAME 2.0 [5], the six following elements must be provided to support microarray publications:

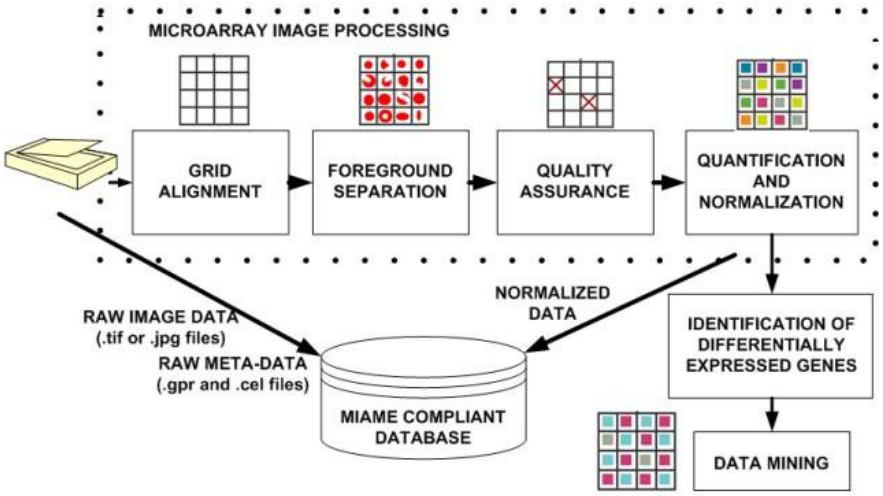


Figure 2.5: Microarray data processing workflow [47].

- experimental design - includes: author, contact information, URL, citations, experiment title, type of experiment, experimental variable, quality indicators, relationships between the array and sample entities,
- array design - includes: information about array manufacturing given by provider (platform type, provider, surface type), description of elements or spots used on a surface (e.g. DNA clones), description of specific properties of each element (e.g. DNA sequence),
- samples - include labelled nucleic acids that represent a transcript in a sample for which the gene expression profile was established (source of the original sample with any biological *in-vivo* or *in-vitro* treatments applied, technical extraction of nucleic acids or their subsequent labelling),
- hybridization - includes: laboratory conditions under which the hybridization were carried out (procedures and parameters),
- measurements - includes: raw data (scans of array), quantification matrices based on image analysis; specification (gene expression matrix),
- normalization - includes: analysis of multiple samples to identify relative changes in expression level, different express genes, discovery of gene classes or samples having similar patterns.

2.2.3 Toxic Effect

A chemical substance that causes an adverse effect (toxicant) is recorded together with a dose and its exposure time on a living organism. There are two types of doses: internal and delivered. The first type, often called absorbed dose, describes the total volume of substance that is absorbed and distributed throughout the organism, often expressed in terms of the concentration in plasma/blood or in an organ. The delivered dose is the total dose given to the organism irrespective of what fraction of this is absorbed. This is often expressed in units of mg/kg (amount of chemical/body weight). In short and long term experiments we consider the following types of the exposure times: less then 24 hours, one day up to one month, up to three months and longer than three months. These exposure times are called: acute, subacute, subchronic and chronic exposure, respectively. According to the type of exposure and dose, there are various defined responses of a live organism [52]:

- acute toxicity - an adverse or undesirable effect occurred in a short period of time (24 hours) that results either from a single or multiple exposures,
- chronic toxicity - an adverse or undesirable effect after long-term exposure (months, years), usually repeated and lower level exposures,
- local toxicity - an adverse or undesirable effect occurred during contact with toxicant (e.g. skin burns),
- reversible toxicity - an adverse or undesirable effect than can be reversed after the exposure stopped,
- systematic toxicity - an adverse or undesirable effect can be seen in some part of a live organism resulting from distribution of the chemical around the body,
- delayed/latent toxicity - an adverse or undesirable effect that occurred long after exposure,
- allergic reaction - reaction to a toxicant caused by an altered state of the normal immune response (never seen for the first exposure but seen with subsequent exposures).

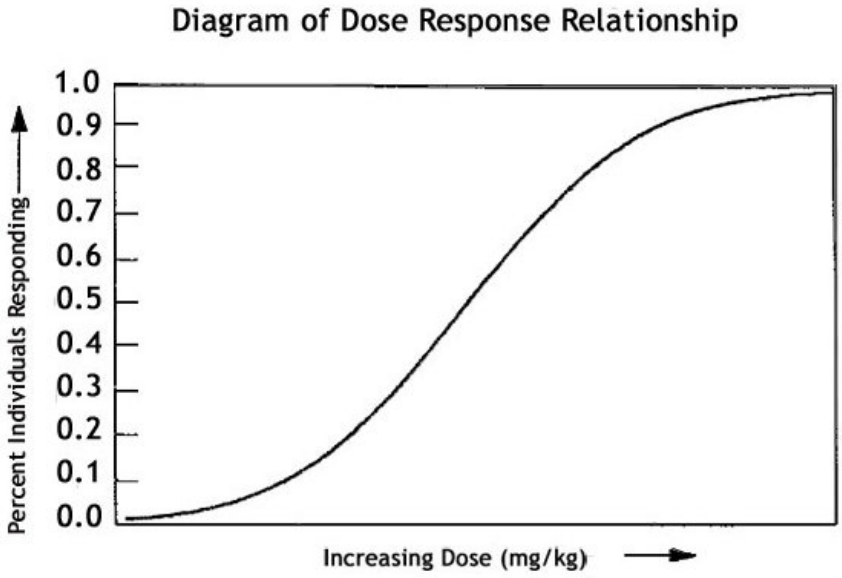


Figure 2.6: Dose-response relationship.

Toxic effects on an organism are related to the amount of exposure often called dose. The dose-response relationships describe the change in the effect caused by different levels of doses after a certain exposure time. Toxicological studies are focused on measurements of dose-responses parameters to define safe and hazardous levels and dosages for various chemicals. The dose-response curve illustrate this relation the X-axis correspond to the concentration of the chemical usually given in milligrams, micrograms, or grams per kilogram of body-weight for oral exposures or milligrams per cubic meter for inhalation exposures. The Y-axis corresponds to the biological response (see Figure 2.6). The are many various concentration measures. All of them depend to the exposure time and exposure route (e.g inhalation, dietary). The most common concentration measures are:

- LD_{50} – lethal dose required to kill half the members of a tested population after a specified test duration,
- EC_{50} – half maximal effective concentration of a chemicals which induces a response halfway between the baseline and maximum after a specified exposure time. It is used to measure drug potency,
- IC_{50} – half maximal inhibitory concentration is a measure that describes how

much of a particular chemical (inhibitor) is needed to inhibit a given biological process by half,

- TD – toxic dose that will produce signs of toxicity in a certain percentage of organisms,
- NOEL – no-observable-effect-level dose is the highest dose or exposure level of a chemical that produces no noticeable toxic effect on the organism.

2.3 Data Integration

In-vivo and *in-vitro* data are distributed across various resources such as scientific articles, company internal reports, governmental organisation documents and many institutional services. Together with chemical information *in-vitro* data are used to predict *in-vivo* toxicity and to prioritise animal testing. Integrating this information in publicly available datasets by sufficient extraction, curation and pre-procession is both challenging and extremely valuable.

Data integration is concerned with providing tools for unified access to data from different sources [68]. The data format is defined by the global schema to represent all information which can be query by a user (see Figure 2.7). It is a significant approach for both the enterprise and scientific information integration. Especially, it is also important for the rapid developing life sciences where information exchange is a one of the main challenges to multi-organisation collaboration.

The additional aspect of data integration is to ensure the high quality of combined information that is provided to the user. There are many data quality dimensions such as:

- specification - measures data standards, data models, meta data, and reference data in terms of existence, completeness, quality, and documentation,
- completeness - measures data attributes according to existence, validity, structure or content,
- accuracy - measures correctness of database content,
- consistency (synchronisation) - measures data equivalence across enterprise,

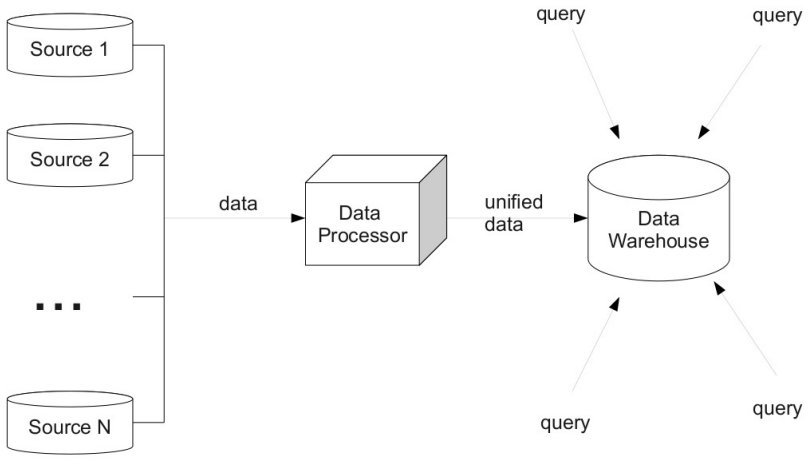


Figure 2.7: Simple schema for data integration.

- timeliness (availability) - measures data validity and availability in a given framework of time,
- security - measures the level of the information security.

Systems which deal with data integration, provide internal rules to ensure data quality and its correctness. Unfortunately, recent studies show lack of consistency in structural representation and systematic chemical identifiers within and between databases [1]. This has an impact for data merging, especially when systematic identifiers are used as a key index for structure integration or cross-querying several databases. This enhances a need for a definition of chemistry standardisation rules and their implementation in order to increase information consistency.

In predictive toxicology the quality of individual data must be assessed according to the correctness of the chemical/biological information and toxicity values. The current standard is to use the Klimisch schema proposed in [63]. This schema provides a set of criteria required to assess data quality:

- reliability – data must be reliable, they must accurately represent the toxic endpoint,
- consistent – experimental results must be repeatable with small statistical error,

- reproducible – experiment procedures should be independent from the environment and repeated tests in various laboratories should give similar results

This schema considers only four categories of data reliability: reliable without restriction, reliable with restriction, not reliable, not assignable. Unfortunately, the Klimisch schema is very general and it is difficult to distinguish between these categories. To address this gap the reliability assessment tool (ToxRTool) [98] was developed. The transparency of the categories is increased by an extended list of evaluation criteria for scoring toxicity information. Both methods are designed as a set of questions and require a human expert to provide an answer. Thus, the assessment of the data quality is biased by an expert's experiences and preferences. To reduce this bias and to provide more automated way of quality assessment a fuzzy expert system has been proposed in [126]. This system uses rules from ToxRTool, and is able to evaluate reliabilities of toxicological data based on the currently available metadata.

The process of data integration combined with the information quality assessment is called *data curation*. This process does not stop, the data should be constantly validated, integrated and maintained when there are new experimental results available. It is also an important step in modelling while high quality data is required to build accurate predictive models. Poor quality toxicity data with errors and a lack of information contributes to poor predictive performance and low statistical fit. The following sections introduced the processes of model development and their validation in predictive toxicology, as well as discuss the importance of the usage of *in-silico* methods in order to reduce the number of animal tests.

2.4 Models in Predictive Toxicology

Integration of *in-vivo* and *in-vitro* data, development of statistical, cheminformatics and bioinformatics algorithms, and data mining tools have led to an enormous increase in the number of models for predictive toxicology. Assessment and application of computational methods (often called *in-silico*) can be used to reduce animal testing. There are two main contemporary approaches: Quantitative Structure-Activity Relationships (QSAR) which seek to predict the toxicological effects of compounds solely from their molecular structure, and Physiologically Based Pharmacokinetic (PBPK) modelling

which can be used to extrapolate between *in-vitro* and *in-vivo* exposure conditions. In subsequent chapters of this thesis all presented work is exclusively concerned with QSAR models.

2.4.1 Quantitative Structure-Activity Relationships

Quantitative structure-activity relationships (QSAR) correlate a chemical structure and properties with biological, chemical or environmental activity [75] whereas SAR associate the molecular features with its activity [3, 41]. Recently, many QSAR modelling tools have been developed using the following techniques: Partial Least Squares Regression (PLS) [123], Decision Tree [96], K-Nearest Neighbours (KNN) [60], Support Vector Machine SVM [58], Artificial Neural Network (ANN) [3, 41] and Random Forest [6, 106]. There is also a lot of interest aiming to support automated QSAR modelling [39, 46, 113] and to build consensus models [128].

QSAR models play the crucial role in virtual screening and *in-silico* modelling. They are considered to be an alternative to expensive animal testing due to their relatively low cost. In the European Union the Registration, Evaluation and Authorisation of Chemicals (REACH) [94] legislation allows for a registration of chemicals which were tested using, inter alia, virtual screening tools. The usage of such modelling tools requires a proof of their reliability and predictivity by a well documented validation process. To make a reliable prediction, a model should be statistically significant and robust, have its application boundaries defined and be validated by an external dataset [33, 114]. Based on this assumption, the first validation principles were assessed in 2002 and then extended in 2004. Currently, they are known as the OECD Principles for QSAR Validation [77]. According to these principles, QSAR models should be associated with the following information:

- Defined Endpoint (Principle 1): The intent of this principle is to ensure clarity in the endpoint being predicted by a given model, since a given endpoint could be determined by different experimental protocols and under different experimental conditions.

- Unambiguous Algorithm (Principle 2): The intent of this principle is to ensure transparency in the model algorithm that generates predictions of an endpoint

from information on chemical structure and/or physiochemical properties. It is recognized that, in the case of commercially-developed models, this information is not always made publicly available. However, without this information, the performance of a model cannot be independently established, which is likely to represent a barrier for regulatory acceptance.

- Defined Domain of Applicability (Principle 3): The need to define an applicability domain expresses the fact that QSARs are reductionist models which are inevitably associated with limitations in terms of the types of chemical structures, physiochemical properties and mechanisms of action for which the models can generate reliable predictions.

- Appropriate Measures of Goodness-of-Fit, Robustness and Predictivity (Principle 4): The wording of the principle is intended to simplify the overall set of principles, but not to lose the distinction between the internal performance of a model (as represented by goodness-of-fit and robustness) and the predictivity of a model (as determined by external validation).

- Mechanistic Interpretation (Principle 5): It is recognised that it is not always possible, from a scientific viewpoint, to provide a mechanistic interpretation of a given QSAR, or there even be multiple mechanistic interpretations of a given model. The absence of a mechanistic interpretation for a model does not mean that a model is not potentially useful in the regulatory context. The intention of this principle is not to reject models that have no apparent mechanistic basis, but to ensure that some consideration is given to the possibility of a mechanistic association between the descriptors used in a model and the endpoint being predicted, and also to ensure that this association is documented.

In general the QSAR model development process is divided into three steps: data preparation, model generation and validation. These steps together with the above principles require more detailed analysis in the model development process (see Figure 2.8). In the following sections, the description of good practices to provide accurate models is presented.

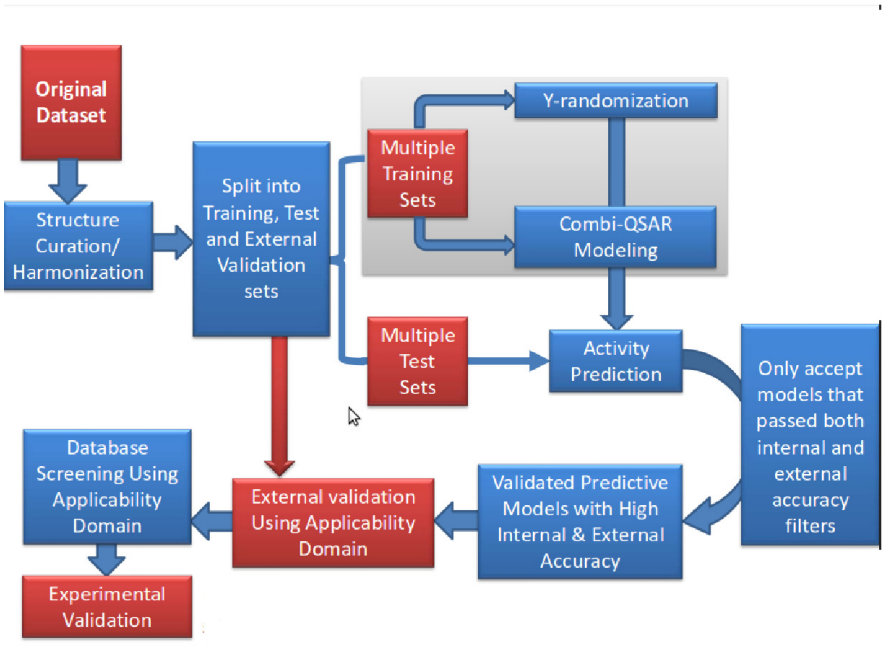


Figure 2.8: QSAR development workflow [113].

2.4.2 Data Preparation

Data quality inherently affects the quality of models [29]. Thus, data curation is a crucial step in the data preprocessing phase. This is also a first step in model development process. The processed quality data is further split into two sub-subsets: modelling and external validation sets. There is a big discussion of the best method for the partition percentage. This operation relies on the size of the entire dataset and it is subjectively based on the modeller decision. The best partition should guarantee that these two subsets are spread far apart over a wide area of the chemical space and are well balanced [113]. The validation dataset should be distributed across this space to ensure diversity of selected chemicals. Random splitting is the most naive method and unfortunately it does not satisfy the above assumption. There are more efficient partitioning techniques which are based on similarity analysis [34]. The modelling dataset is further split into training and testing datasets. Both these sets are used in model generation and more detailed description is presented in the next section.

Another issue of data preparation is the size of a dataset. The number of chemical compounds included in a dataset should not be either too small or too large. Large

datasets can produce an inefficiency of generated models whereas small datasets may result in inaccurate models. Such inefficiency is related to a selected approach or method processing the large dataset. Each method is limited by an available space and computation time needed to build a model. The dataset should be properly balanced [113, 114]. This means that the number of elements (chemical compounds) from different classes or categories (based on their activity) should be equalized. Unbalanced datasets cause higher errors of a correct prediction for the smaller number of elements within a class.

The last step in data preparation is outlier and activity cliff detection [72]. The main hypothesis in QSAR modelling is that similar chemicals have similar properties [51]. Based on this definition, activity cliffs are defined by areas in the chemical descriptors space where the similarity hypothesis does not hold. There are two types of outliers: leverage and activity. They represent either the real values or errors in the structure representation as well as in their activities. There are many methods for outlier detection: similarity distance measures, Hotelling's test, or Cook's distance [11, 75]. In QSAR modelling, the common practice is to remove outliers before model generation. Their presence in the training dataset will lead to model instability. Nerveless, taking outliers into account to develop models and to provide the analysis of their mechanistic interpretation can open a new perspective in building QSAR models.

2.4.3 Model Development

For model generation, the preprocessed modelling dataset is used. This dataset is again partitioned many times using the well known cross-validation (CV) technique. Cross validation involves round estimations of a model. One round of the cross-validation method splits a dataset into training and testing sets. The training dataset is used to generate a model whereas the testing dataset is used to assess its predictivity. Many rounds of CV are applied and validation statistics are collected. A model with the best predictive ability is selected for further external validation tests (see Figure 2.8). The most common CV methods for predictive toxicology are presented in [101, 113, 114, 123, 128] and they involve:

- LMO CV (leave-many-out cross-validation) - is generalisation of LOO CV (leave-one-out). In the literature it is also known as k-fold cross-validation. The mod-

elling dataset is divided onto k separate subsets of equal size. $k-1$ subsets are used in model generation and the one remaining subset is used for model validation.

- Bootstrapping - re-sampling dataset on k groups, each of size N. Elements from the modelling dataset are selected randomly, thus, some elements may be placed several times or never selected into the training dataset. Models generated on these N elements are validated using other elements from the modelling dataset. To generate the best model, this procedure is repeated k times.

- Y-randomisation test - the modelling dataset is partitioned into training and testing datasets. The dependent variable vector of the training dataset is shuffled randomly before a model is generated. The testing dataset is used for model validation.

The number of available molecular descriptors which can be calculated based on the chemical molecular structure is in the thousands. This makes QSAR modelling a difficult task according to the dimensionality of the data. To reduce descriptor space feature selection techniques are required. There are three common methods: filters, wrappers and embedded. Wrappers select the possible sets of descriptors and run the model on each set. They are very time consuming and not efficient according to the size of input data. The filter method selects a subset of the best descriptors according to some filter criteria to find the most relevant variables. The embedded methods are embedded in the model generation process. Features are added or removed while building the model, depending on the changes in model accuracy [37, 66].

2.4.3.1 Internal Validation

The best model is selected based on internal validation. Before diagnostic statistics for the internal validation are presented, the following information has to be recalled. Consider the input dataset (T) described as follow:

$$T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{ik} & y_1 \\ x_{21} & x_{22} & \cdots & x_{2k} & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} & y_k \end{bmatrix}$$

where chemicals are defined by tuples $(x_i, y_i) \in X \times Y$ for $i = 1, \ldots, n$. The set $X = \{X_1, \cdots, X_k\}$ defines molecular, chemical or physical descriptors whereas Y is a set of observed activity values. The QSAR model M maps the descriptor space into activity domain:

$$M : X \to Y. \tag{2.2}$$

The model M can use regression methods for continuous data or classification techniques for discrete values. For regression models we consider two correlation coefficients: squared correlation coefficient r^2 and predictive squared correlation coefficient q^2. A square correlation coefficient r^2 of fitting model has become a very popular measure of the model goodness-of-fit and it is defined as follow:

$$r^2 = 1 - \frac{\sum_{i=1}^{TR} (y_i - {y_i}^{fit})^2}{\sum_{i=1}^{TR} (y_i - \bar{y})^2} = 1 - \frac{RSS}{SS} \tag{2.3}$$

where y_i is the observed chemical activity, y_i^{fit} is its fitted value by a model M.

The average of the observed value $\bar{y}$ over the training dataset (TR) is given by formula:

$$\bar{y} = \frac{1}{n} \sum_{i=0}^{n} y_i \quad n = |TR|. \tag{2.4}$$

In formula (2.3), RSS is called the residual sum of squares and SS the sum of squares. The r^2 statistic describes the ability of a model to reproduce data within the training dataset but it is not enough to describe the robustness and predictivity of this model [11, 34, 101]. These characteristics are defined by the predictive squared correlation coefficient q^2. Internal validation is performed to calculate this coefficient. A model M is applied for the testing dataset (TS) to predict the activity value $\hat{y}_i$ for all chemicals in this set. The predictive squared correlation coefficient q^2 is defined as follows:

$$q^2 = 1 - \frac{\sum_{i=1}^{TS} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{TS} (y_i - \bar{y})^2} = 1 - \frac{PRESS}{SS} \tag{2.5}$$

where $\bar{y}$ is defined by formula (2.4) and PRESS is known as the predictive residual sum of squares. The average of the observed activities for the training data set in equation (2.5) is used because it leads to the reduction of noise caused by variation of the testing dataset mean.

		observed	
		Positive	Negative
predicted	Positive	True Positive (TP)	False Positive (FP) (error I)
	Negative	False Negative (FN) (error II)	True Negative (TN)
		Sensitivity	Specificity

Figure 2.9: Diagnostic test: sensitivity and specificity.

For cross-validation and automated QSAR modelling, models with higher q^2 values are used for further external validation. Many authors consider their models highly predictive in the case when $q^2 > 0.5$ [33]. The low value of q^2 indicates a low predictive power of the model whereas high q^2 does not prove its high predictivity. Thus, external validation is required.

For QSAR classification models, misclassification statistics are used including: specificity, sensitivity, accuracy and precision (see Figure 2.9). The confusion matrix is used to display predictions made by the model. It contains a number of correct classifications and two types of errors. The error of a type I called *false positive* rejects the hypothesis when it is true. The error of type II (*false negative*) does not reject the hypothesis at the moment when it should be rejected. The correct classification are called true positive and true negative. Together with the errors, they are used in the diagnostic test to define model predictivity [35, 37].

The accuracy is a true value of the model predictivity. The precision is a measure of the accuracy that defines the number of elements a specific class has predicted. It is defined by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where TP is the number of true positive and FP is the number of false positive predictions for the considered class. The sensitivity measure (called recall) is the ability of a predictive model to select instances of a certain class from a dataset. It often corresponds to the true positive rate and it is defined by the formula:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where true positive (TP) and false negative (FN) predictions are related to the considered class. The specificity corresponds to the true-negative rate and is defined as follows:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

where TN is the number of true negative and FP is the number of false positive predictions. Sensitivity and specificity define model robustness. There are various methods for prediction error calculation. Mean absolute error, root mean squared error, relative absolute error and root relative squared error are all used in QSAR modelling.

2.4.3.2 Applicability Domain

The last step of the model development process is its applicability domain estimation. The applicability domain (AD) (see Figure 2.10) is defined as *"the response and chemical structure space in which the model makes predictions with a given reliability"* [75]. The chemical space is a multidimensional space, where each dimension represents: structural, physical, chemical or biological property of a chemical compound. Applicability domain determines the boundary of chemical sub-space where models are reliable and it also supports the controlled extrapolation of these models into entire chemical space. This fact ensures that the QSAR model can be used for chemicals which fall into its applicability domain and at the same time it does not guarantee a high model predictivity. Applying these models for chemicals from outside of their applicability domains increases the likelihood of inaccurate prediction.

The process of AD estimation is model-dependent and based on a training set domain, moreover, there is a relation between the AD estimation and variable selection techniques [128]. Thus, there is no universal method for AD estimation. As shown in [53] the different approaches produce different applicability domains. The choice of the particular AD estimation methods depends on a requirement for the data distribution in a training set and the dimensionality of the model. Currently, there are four main techniques and there is still ongoing research to find efficient methods.

The most common technique for the applicability domain estimation uses range-based methods [75]. Chemical descriptors are defined as ranges of their values and generate a hyper-rectangle. The applicability domain is defined by this hyper-rectangle. Unfortunately, this method does not detect the intersection of the hyper-planes and

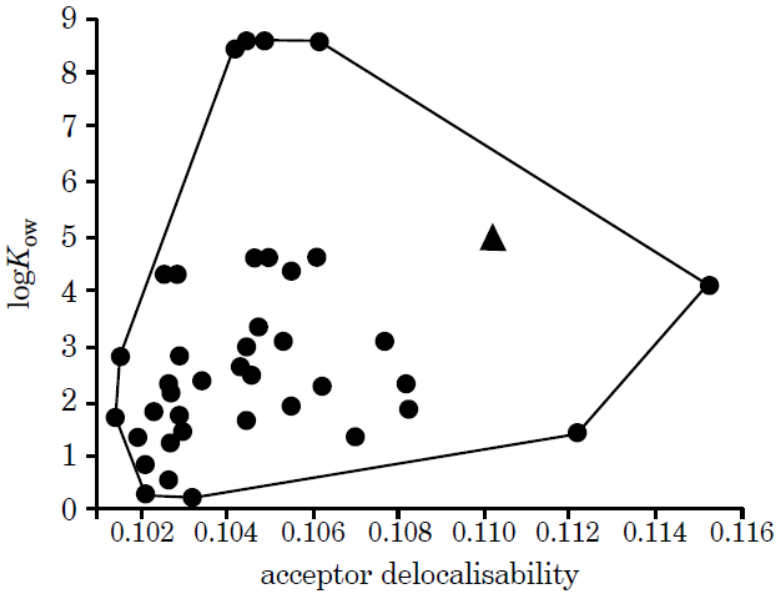


Figure 2.10: Example of applicability domain estimation for model predicting $\log K_{ow}$ using the acceptor delocalisability descriptor [75].

does not take into account the correlation between descriptors. An another common technique to assess the model applicability domain is Principal Component Analysis (PCA) [57]. It is based on the rotation of dependent variables X (descriptors) to correct the correlation between them.

The convex hull calculation is another example of applicability domain estimation. This method estimates the coverage of the n-dimensional set of variables. In two-dimensional space, it is represented as a polygonal figure whose interior defines the model applicability space. This approach is well known in computational geometry [17]. There are a few efficient algorithms for two and three dimensional problems. Unfortunately, with the increase of a number of descriptors used to generate model, the complexity of the convex hull calculation also increases. Additionally, this method does not detect empty regions in the descriptor space. As it is shown in Figure 2.10 the data covers evenly a space for $\log K_{ow} < 5$ and acceptor < 0.110. The other regions in the convex hull does not contains many data (triangle point) or are empty.

One of the most efficient approaches to estimate the applicability domain are distance based techniques. These methods calculate the distance from a searching point

to the training dataset. There are many approaches: distance to the mean of the dataset, average of all distances between query point and the dataset or maximum distance. The Euclidean distance is the most frequently used technique, however, the Mahalanobis and city-block are used as well [53, 75]. Together, they are the most common methods for finding similarity of chemical compounds. For a given model the applicability domain threshold is defined. For all chemicals where the average distance to the training dataset is greater then this threshold, the model can give an inaccurate prediction.

The last method to estimate the applicability domain is based on density estimation. This method involves the determination of the high density region. There are two approaches: parametric and non parametric. Parametric methods ensure the the data distribution is close to standard normal distribution (Gaussian Process). Non parametric methods do not make any assumption about data distribution (kernel density estimation). The calculation of the highest density regions is a complex process according to the dimensionality of the chemical space, thus, there is a challenge to provide a fast and efficient algorithm. Recent studies [9] show that the random forest classifier is comparable with the well know Gaussian Process regression [91] for applicability domain estimation. The authors provided a generic machine schema for class probability estimators.

The applicability domain can be in two types: global and local. Global applicability domain defines a broad chemical space using all pre-calculated chemical compound descriptors, whereas local applicability domain is defined by selected descriptors in the model generation process. The breadth of the applicability domain has influence on the model predictivity. The narrower the applicability domain, the higher the predictivity of the model. The applicability domain is often used to validate a predictive model. The elements that are within the boundary of applicability domain as well as elements from the outside the applicability domain are used for quantitative assessment of the model robustness and its predictive power [114]. The next section discusses the external validation methods.

2.4.4 External Model Validation

In Section 2.4.3.1, the internal validation for the model development process was discussed. Two measurements of model reliability r^2 and predictivity q^2 for regression

model were introduced. In [33, 114] authors showed that the cross-validation correlation coefficient q^2 is an insufficient factor to ensure the predictive power of QSAR models. According to the workflow presented in Figure 2.8 (see Section 2.4.3), a model should be tested on an external dataset using model applicability domain [113]. This step is called the external model validation process.

The feature selection and model generation techniques impact on model predictivity power. Applying one model on many subsets of selected features or applying various algorithms using the same features set leads to different results. Despite this fact, in many QSAR studies, an automated modelling technique is recommended. Multiple models are built at the same time and internal validation is used to select the most accurate model. Such model is further validated using an external validation dataset (TE). For classification models the same techniques for assessing model accuracy are used as in internal validation. For regression models the predictive squared coefficient correlation q^2_{ext} is defined as follows:

$$q^2_{ext} = 1 - \frac{\sum_{i=1}^{TE}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{TE}(y_i - \bar{y})^2}, \tag{2.6}$$

where $\bar{y}$ is defined by formula (2.4). The coefficients q^2 (see formula (2.5)) and q^2_{ext} are equivalent. The first is calculated for internal model validation over the testing dataset and the second for the external dataset. Both refer to the same mean of the observed values for the training dataset. Additionally, the coefficient of determination r^2_{ext} is calculated as follows:

$$r^2_{ext} = 1 - \frac{\sum_{i=1}^{TE}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{TE}(y_i - \bar{y}_{TE})^2}, \tag{2.7}$$

where the $\bar{y}_{TE}$ is a mean of observed values in the validation dataset.

The analysis of the squared correlation coefficients defined by formulas (2.3) and (2.7) shows that $r^2 \in [0, 1]$. For the most naive model where the expected prediction is close to the mean of observed values ($\hat{y} \approx \bar{y}$), then r^2 is equal to zero. In this case, parameters of a regression model are independent from observations and do not explain them. Such a model does not explain any variations of the activity and we can assume it does not predict better than the mean of the dataset. We consider a predictor to be better than using a mean when $RSS - SS \leq 0$ (see formula (2.3)). In a case, when

$RSS \to O$ then $r^2 \longrightarrow 1$. This means that a model has an ability to reproduce data from the training dataset. When $r^2 \equiv 1$ the model can be consider as ideal or be over-fitted. Over-fitting occurs when a model is very complex, such as having too many parameters relative to the number of observations. The over-fitted model generally has poor predictive performance, as it can exaggerate minor fluctuations in the data.

To detect over-fitting the construction of the *ideal model* is used. To construct the ideal model we map the observed values versus predicted. The regression line can be defined as $y = a\hat{y} + b$ where the slope a is a correlation coefficient:

$$a = \frac{\sum_i^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sum_i^n (\hat{y}_i - \bar{\hat{y}})^2} \tag{2.8}$$

and the intercept is defined:

$$b = \bar{y} - a\bar{\hat{y}}, \tag{2.9}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the average values of the observed and predicted values [33]. For the ideal model, the slope $a = 1$ and the intercept $b = 0$. To calculate the squared correlation coefficients r_0^2 and $r_0^{'2}$ between the actual and observed values we build two regressions of y versus $\hat{y}$ and $\hat{y}$ versus y, i.e. $y^{r_0} = k\hat{y}$ and $\hat{y}^{r_0} = k'y$, where

$$k = \frac{\sum_i^n y_i \hat{y}_i}{\sum_i^n \hat{y}_i{}^2} \quad \text{and} \quad k' = \frac{\sum_i^n y_i \hat{y}_i}{\sum_i^n y_i{}^2}, \tag{2.10}$$

and are the slopes of ideal QSAR models. Then the correlations of determination are defined as follows:

$$r_0^2 = \frac{\sum_i^n (\hat{y}_i - y_i^{r_0})^2}{\sum_i^n (\hat{y}_i - \bar{\hat{y}})^2}, \tag{2.11}$$

$$r_0^{'2} = \frac{\sum_i^n (y_i - \hat{y}_i^{r_0})^2}{\sum_i^n (y_i - \bar{y})^2} \tag{2.12}$$

In other words, the calculation of the correlation between observed and predicted activities is performed. The new regression models are created and we can check how close they are to the ideal model. In this case, slopes k and k' should be close to 1.

There are five conditions introduced by authors in [33] to assess the predictive power of QSAR models. They are widely applied in the model validation platforms such as [44, 46, 80]. To consider the QSAR model predictive, the following condition

must be satisfied [113, 114]:

$$\begin{cases} r^2 > 0.6 \\ q^2 > 0.5 \\ \frac{r^2 - r_0^2}{r^2} < 0.1 \quad or \quad \frac{r^2 - r_0'^2}{r^2} < 0.1 \\ k \in [0.85, 1.15] \quad or \quad k' \in [0.85, 1.15] \\ |r_0^2 - r_0'^2| < 0. \end{cases} \tag{2.13}$$

where r^2 is a correlation coefficient between the predicted and observed values, r_0^2 and $r_0'^2$ (defined by formulas (2.11) and (2.12)) are coefficients of determinations [97] predicted vs. observed and observed vs. predicted, respectively. The values k and k', defined by formula (2.10) represent slopes of regression line through the origin [33].

The q^2 value describes the benefit of using the generated model (predictor). In contrast to r^2, $q^2 \in (-\infty, 1]$. If $PRESS \to \infty$ then in the validation set there is the high variation of observed activities according to the expected mean. In this case $q^2 \to -\infty$ and the mean of observed values $\bar{y}$ is the better estimator than the predictor. In a case when $q^2 = 0$ there is no difference between using the mean and predictor. Otherwise, a generated model gives better predictions.

Another most common statistics used to compare models is the root mean square error of prediction (RMSE):

$$RMSE = \sqrt{\frac{\sum_i^n (\hat{y}_i - \bar{y}_i)^2}{n}}, \tag{2.14}$$

and mean absolute error (MAE):

$$MAE = \frac{\sum_i^n |\hat{y}_i - \bar{y}_i|}{n}, \tag{2.15}$$

We can use both RMSE and r^2 to compare regression models that use the same dependent variable. The higher r^2, the lower RMSE. RMSE could be a good indicator for model quality if it could be trusted. However, the uncertainty of RMSE may be introduced by model misspecification in the stage of model evaluation or over-fitting during the model development stage. Thus, it is important to provide the goodness-of-fit and predictivity statistics to assess model quality.

2.4.5 Predictive Toxicology Systems

Mastering high quality models and data has become an important task for predictive toxicology. Various organisations and institutions are interested in collections of high quality data and models. The number of toxicological data and available models has increased dramatically in recent time. This is why data and model collections within the predictive toxicology information management system can help in knowledge exchange and information utilisation processes. Despite this observation, there is a need to develop a flexible and interoperable framework for predictive toxicology which aims to address the following challenges: data integration, data quality assessment, automated model development, model validation reporting, model aggregation. In this section the most important existing predictive toxicology applications are discussed according to data integration and exchange, model development, collection, validation and reporting.

2.4.5.1 Ambit

Ambit, developed by IdeaConsult Ltd. [43, 44], is an open-source chemoinformatics data management system. It was designed to store a large amount of data and to provide tools for data mining and analysis. The main goal for the Ambit designers was to develop the extensible QSAR decision support system. Interoperability, flexibility and transparency are its three design principles. The XML representation of chemical compounds is used to exchange data within various organisations. Modular database design supports the flexibility and possibility for the future extension of the system functionality. Publicly available sources support the transparency principle. Ambit stores more than 450,000 chemical structures together with their various identifiers like: names, CAS number, InChI.

The Ambit data management system includes the Ambit relational database and several libraries of chemoinformatics applications. The database stores: different chemical compound structures and their chemical and physical properties which are collected from different resources, including information about toxicity endpoints, experiments and literature references.

Ambit Discovery provides statistics and data mining techniques for the chemical grouping and applicability domain estimation. The CDK library [103] is used

for chemical compound descriptors calculations. The QMRF Repository supports the model reporting according to the OECD validation principles [77]. Collected models can be easily searchable within the Ambit framework.

Ambit XT, a set of applications, consists of a number of modules which lead users to achieve their goals and plans for predictive analysis. The workflow architecture is used to support data provenance. It includes chemical compound updates history and user actions recording; storage of toxicity data for different endpoints. Data quality assessment is achieved by comparison across different resources.

The ToxMatch and the ToxTree libraries, also developed by IdeaConsult, are built into the Ambit system. ToxTree is a decision making application for a toxic hazard assessment. For toxicity estimation, decision tree techniques are used with a set of built-in rules for classification schemes such as Cramer rules [18], Verhaar scheme [117], Skin irritation prediction [118], Eye irritation prediction, Benigni and Bossa rules for mutagenicity and carcinogenicity [4]. Moreover, ToxTree supports a development of new rules. Using java environment, user can easily build his own model within the Ambit workflow framework.

ToxMatch is an open-source application which provides statistical and data mining methods for grouping chemicals based on their similarities, prediction and assessment of physiochemical properties, toxicity, environmental fate and ecotoxicity [44, 56, 86]. the new compound toxicity is assessed from reading across different datasets. ToxMatch supports pair-wise similarity search between compounds, similarity estimation between a single compound and a group of chemicals. The ToxMatch similarity assessment is based on the well known similarity measures [121]. Additionally, ToxMatch supports various structure representations of a chemical compound, including: descriptors, fingerprint, atom environment and the descriptors generation. This software was developed in collaboration with European Commission Joint Research Centre (JRC).

The Ambit system meets most of the requirements for recent predictive toxicology. It is a host for data and models. It provides a lot of predictive toxicology applications which build components for the decision support system. The Ambit system is focused on data integration, applicability domain estimation, similarity assessment and chemical compound structure conversion. Moreover, the Ambit workflow architecture is an excellent environment for toxicity assessment and new rules development. Currently, the QMRF format is widely used for model reporting.

2.4.5.2 OCHEM

The Online Chemical Modelling Environment (OCHEM) [78] is a web-based platform to support QSAR modelling. There is a back-end database (which contains quality-controlled chemical and experimental data) integrated into the modelling framework, with an aim to support a series of steps of building QSAR models. These steps range from dataset search and construction, descriptor calculation, descriptor selection, through model building, simple search, validation, to model storage and reporting [105]. OCHEM allows users to select descriptor calculation, data pre-processing and modelling building methods from pre-defined lists. The training datasets and developed models are easily accessed, manipulated, combined and reused within the OCHEM framework. However, apart from some configuration parameters, there are no representation schemas available that would allow users to export the built models to external formats. In addition, some intermediate results (e.g. the calculated descriptor values) are not available and transparent to users. Practically speaking, it is quite difficult to include all possible methods/tools in one single platform, especially some in-house developed methods. Therefore, the current approach limits the freedom of users to (re-)build the models in other platforms and using different tools. In terms of model management, OCHEM enables users to perform model combination, such as using the output of a certain model as the input of another one. However, this process is purely user-oriented; users have to make their own decisions. Currently, there is no (semi-)automated decision support available to users to identify the most suitable models when given the problem at hand.

2.4.5.3 JRC QSAR Database

For regulatory purposes, the OECD has developed a standardised QSAR Model Report Format (QMRF) [54] which aims to capture required information to meet the OECD principles [77] for the validation of QSAR models. Besides this, representing and managing QSAR models has attracted increasing attention in recent years. In order to support the identification and retrieval of suitable QSAR models, the Joint Research Center (JRC) has developed a publicly available web-based model database that stores high-quality documentation for QSAR models. The model information is captured using QMRF and only the approved QMRF documents are included in the JRC QSAR

model database [54]. To facilitate the use of QMRF, a stand alone application, named *QMRF Editor*, was developed to support the preparation of QMRF documents. It has also been employed to document QSAR models in other systems (e.g. AMBIT [44], OpenTox [80] and CAESAR [48]). The main content of QMRF is carefully designed to reflect the five OECD principles. In addition, some general information about the model, such as training and validation dataset files, is included in the QMRF document. The web-based inventory allows users to search QMRF documents and structures from the database, submit new (generated from scratch via the on-line QMRF editor) or existing QMRF documents, and review submitted documents. The stored QMRFs can be searched via QMRF number, free text, predefined lists of endpoints, algorithms, software and authors, by using either AND or OR operators. The retrieved documents can be downloaded in various formats, including PDF, MS Excel, XML and HTML. Furthermore, the JRC model inventory enables users to search all included substances based on either exact or similarity modes, by using CAS numbers, formulas, chemical names, aliases, SMILES codes. When searching for a substance from the database, the associated QMRF documents and their relations (e.g. presence of the substance in the training/validation dataset) will also be displayed. QMRF was designed to summarise information about QSAR models, with the particular purpose of regulatory report use. It is useful to retrieve QSAR model documents, however it does not include the executable model files. In addition, only the model performance is recorded rather than the model predictions. This makes it difficult to re-produce a model of interest.

2.4.5.4 QSARDB

QSAR DataBank (QSARDB) is a web-based system which aims to store QSAR models and their associated information [54], and is currently still under development. A web-based Graphical User Interface (GUI) is provided to visually represent model information and its prediction results. Six domain objects, namely *compound*, *property*, *descriptor*, *model*, *prediction* and *workflow* are defined in QSARDB and each domain object is associated with a corresponding XML representation file. Currently, QSARDB contains over 100 QSAR models (mainly classification and regression models) which are categorised by properties, species and endpoints. This enables users to effectively retrieve models by browsing through different categories. In contrast to

other existing QSAR model representation schemas, QSARDB employs the Predictive Model Markup Language (PMML) standard [21] to represent the actual predictive models rather than just the model metadata. Additionally, PMML files are embedded in their corresponding model web pages and XML files. In terms of prediction, one model can have many predictions, including using different validation methods on training, internal/external validation datasets. The predicted values are stored in separate plain text files and embedded as attachments in a single prediction registry XML file. QSARDB intends to include all necessary data to build QSAR models and allows users to download the whole data repository in ZIP files. However, some actual data (e.g. training/testing datasets, descriptor values, PMML model files and predicted values) are included as *Cargos* (attachment files in QSARDB) in the corresponding object domain folders. By doing this, it requires users to have a clear understanding of how the system organises data, if they would like to make use of such information in other platforms. In addition, apart from allowing users to browse all available models based on different categories, there is no functionality provided to support more complex model management tasks, such as model identification and model ranking.

2.4.5.5 OpenTox

OpenTox is an open framework which integrates a wide range of techniques to support toxicity prediction [80]. It currently provides two applications for model development and toxicity estimation. There are two methods, namely lazar (Lazy Structure–Activity Relationships) classification and lazar regression [41], available in *ToxCreate* to generate and validate new QSAR modelling from given experimental data. *ToxPredict* allows users to predict a toxicity endpoint for a given chemical compound. Being supported by other OpenTox concepts, including *dataset, feature, algorithm* and *report*, a RDF/XML model representation schema is proposed to store QSAR models in the OpenTox framework. The model representation mainly contains model metadata (e.g. model ID, name, timestamp, algorithm, training dataset, parameters, dependent/independent variables). However, the actual model is not included in the RDF/XML representations. For prediction, OpenTox uses a separate validation schema to report model performance results by using up to four different validation methods. Only the performance matrix results are stored, the actual simulated values are omitted from the

current representation. Similar to other QSAR frameworks, there is no decision support information (e.g. applicability domain verification or model rating mechanism) available for users to choose the most appropriate model for a given task.

2.4.5.6 Inkspot

Inkspot [46] is a cloud-based system, which hosts data, models and documents, and allows for a collaboration between various organisations and institutions. Inkspot provides data management system to store data and tools for its analysis. It provides also a workflow architecture to help users in easy implementation of new predictive toxicology applications within the framework. To support collaboration and grouping users in specialised communities, InkSpot developed the social network architecture. Every user has his own user space where he can store his data and share information with other members in the same community. It is worth noting that the Inkspot architecture does not support data integration and data curation. In addition, information stored within the InkSpot service can be multiplied in different user spaces and this may cause data inconsistency.

Inkspot includes the Discovery Bus component for automated QSAR modelling [13]. Discovery Bus is a multi-agents management software for automated QSAR model development without expert intervention. An agent or a class of agents are a piece of software being able to respond to a particular query. The idea of multi-agent systems is to split a given request into small sub-tasks across many agents. There are two main groups of agents: workers and controllers. Controllers are responsible for the planning, scheduling and controlling of a given task. The central agent (planner) assigns tasks to worker agents. Every time, a worker agent responds to a sub-query, its reply is stored in a discovery database. Installers and reapers support worker agents. Installers are responsible for installing and activating new agents whereas reapers take care of the agents which failed during their performance. Controller and scheduler agents are responsible for setting the execution priority and scheduling tasks. Pathways of the problem solving strategy are stored in a plan database by the central planner agent. They can be retrieved any time the similar query is submitted.

In the context of QSAR modelling, classes of agents include calculation of the molecular descriptors, feature selections, data preparation and machine learning tech-

niques for accessing the toxicity. Each descriptor agent calculates a different set of chemical compound descriptors. Later, sets can be combined across all descriptor agents. The dataset is divided into training and testing sets. The training dataset is separated in 10 cross-validation sets. The testing set is used for the further model validation. Different strategies are used for feature selection by feature agents. Each selected features subset is available for the model building agent. Statistical and machine learning techniques are used to build models. For every feature selection the best model is chosen across different agents based on the cross-validation statistics. Then, external validation is performed to assess the quality of the predictive model. External validation statistics [33, 34, 114] are used to calculate model performance metrics. It is done by the Open-QSAR component built-in the Dictionary Bus [79]. This matrix can be used in the model comparison and model selection approach to choose the best methods to predict required values.

2.5 Model Reuse

The process of product development, such as drug design, cosmetics or plant/crop protection, takes usually up to ten years and companies spend hundreds of millions of dollars on developing a new product. This process is divided into four phases: discovery, profile, evaluation and support. In the first phase, from millions of chemical compounds, thousands are selected according to their biological, chemical or physical properties. This chemical compounds group is profiled against various targets (e.g biochemical and physiological targets related to metabolism, growth, development, nervous communication) and tens of them pass to the evaluation phase. After the evaluation phase usually only very limited number of chemicals are selected as a product that can be introduced into the market. Sometimes, the final product does not meet the safety regulation and a company can only register and not sell it. In this situation, the product must to be return to the development or evaluation phase for further development or rejected. The identification of chemicals that may fail in the evaluation phase become crucial for many companies. Thus, many organisations focus on better information organisation and reuse in order to reduce the cost of testing and manufacturing in the product development phase.

The usage of predictive models for new chemicals evaluation process has become

a key strategy in various organisations. In [113] the authors demonstrated that QSAR models can be used as virtual screening tools if they are robust and properly validated. According to Annex XI of the REACH legislation [93], results of QSAR modelling may be used instead of testing when QSAR model was:

- scientifically validated,
- substance falls within its applicability domain,
- results are adequate for the purpose of classification and labelling and/or risk assessment,
- documentation of the applied method is provided.

Having such a wealth of previously developed models at our disposal can bring a number of benefits if we are able to make effective reuse of them. Trained models usually represent a significant investment of time, and may contain high-impact insights into the relationships between particular chemical attributes and specific toxicological effects. In the past, published models in the literature were often unused and unseen within communities because they were not publicly available or not annotated to be suitable for reuse. They are often difficult to restore to a useful form as the published details are either incomplete or the supporting information is missing. Lack of a standard description format for model representation, and the lack of stringent reviewing and authors' carelessness have been identified as the main causes for incomplete model descriptions [71, 76]. Reproducing work to reach the same conclusions is obviously an inefficient use of time in the best case, and in the worst case a different and possibly incorrect conclusion may be reached. In such situations the knowledge that was previously discovered and encapsulated within a predictive model may be lost. To avoid this, the knowledge should be captured together with the human experience of knowledge itself and its use, and the proper management of such knowledge is required [100].

In the previous section, a number of various toxicological system have been discussed in an attempt to address the aforementioned problems. To make models more reusable sources of information, various model representations and ontologies have been implemented for each toxicology system. This allows users to build models or

workflows and reuse them only within a particular system To make use of existing models, users are required to register with the system and also submit data that they use for predicting a given endpoint. This discourages modellers form using such predictive toxicology systems to some extent. Often, the data in use is confidential and modellers do not fully trust the existing systems. Additionally, model exchange across different platforms is challenging, due to the various model representation formats.

Models that are stored in model databases can be reused to predict toxicity of new chemical compounds. Unfortunately, this involves a manual process of model identification. A potential user is required to make a comparison of model applicability domains and their predictivity for a given activity in order to decide if the model can make reliable predictions for a given chemical compound. Model comparison is a difficult task since models are generated using various subsets or various chemical compound descriptors. Consequently, models can be trained and validated on different datasets. For regression models, the model performance can be described by the predictive squared correlation coefficient q^2. Since the sizes and contents of modelling and validation datasets may differ for various models, the value of q^2 is not sufficient for model comparison [33]. Several model performance matrices were analysed in the context of model validation and model selection [114]. They are applied in automated model development process where models are validated by the same dataset. In the case where two models come from different sources, model comparison becomes challenging. This requires predictive models to be validated across the entire chemical space, which is very difficult as the list of available chemicals and assays is limited. Based on this observation there is a need to develop a framework for automated model identification for new chemicals. Such a framework should include a comparison model performances as well as their applicability domains.

The last element related to the efficient model reuse is model mechanistic interpretation. Understanding why models makes particular decisions and knowing the mechanism that leads towards the predicted outcome, increase a trust in the model prediction. It is also the fifth requirement in OECD principles for QSAR model validation. Unfortunately, not all models can be easily interpreted. It is possible for linear models, thanks to the availability of the model parameters and their statistical significance. For non linear models this information is hidden within the model structure and often it is difficult to extract it. Thus, model reuse does not only include the identification

which model would be the most suitable for a new chemical compound but also the interpretation of the model decision and analysis of this interpretation. The framework reviewed in the previous section do not include such functionality.

2.6 Summary

Accurate and appropriately shared models can bring a number of benefits if we are able to make effective use of the existing expertise. This chapter presented the predictive toxicology domain and discussed main challenges related to the data and model integration and quality assessment. This chapter also reviewed current practices and methods for model development, validation and model reuse. This review included a OECD principles for QSAR model validation, the REACH regulation for model reuse as an alternative to animal testing. The lesson that has been learnt is that once a model has been built, it can be effectively reused for new chemicals evaluation. Global models can be officially approved and built within predictive toxicology systems. Local models can be used for new chemicals only if they are properly annotated, stored and validated.

A number of existing predictive toxicology systems has been reviewed with regard to model representation format and functionality that allow efficient model reuse. All of them efficiently support only the model development process. Clearly, there is a need for automated techniques for mining model repositories such us model quality control, data and model integration, model comparison, model identification and interpretation. The main question here is: How can we reuse existing models for new data? This is why this thesis focuses on model governance, model identification and interpretation to build theoretical framework and methods that allow for better model management and reuse. The following chapters open a new domain of research and raise questions about model reuse in predictive toxicology.

Chapter 3

Model Governance

Effective data governance can enhance quality, availability and integrity of organisational data through cross-organisational collaboration, and implementation of structured policy-making. Currently, models are also recognised as important information assets which can support decision-making and business strategies. This is so in domains such as pharmacy, cosmetics or agriculture where there is a need to reduce costs of a new product development and to increase the safety of the product introduced to the market. Implementation of the data governance principles to model management can help to balance factional silos with organisational interest by lowering costs, reducing risks and increasing data confidence. This chapter extends the data governance principle to define a framework for model governance. Six rules for defining minimum information about QSAR model and the XML schema are proposed. Parts of this chapter were published in [83].

3.1 Introduction

In the literature, governance is defined as a set of processes and strategies that address the problem of formal management of important information assets. The terminology of data governance was firstly established for the IT sector where governance was focused on information technology systems, their performance and risk management [107]. The well known Weill and Ross [120] governance framework for IT assets provides the following decision domains: IT principles, IT infrastructure strategies,

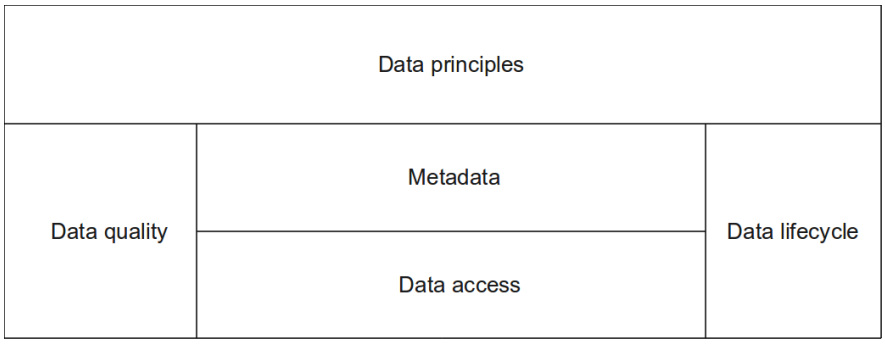


Figure 3.1: Decision domains for data governance [62].

IT architecture, Business application needs and IT investments to define the IT scope. This domain defines the role that IT plays in the organisation and decisions for the IT architecture to meet business application needs.

Constantly increasing amount of available information and the growth of digital data made many organisation aware of the importance of their data [22]. There was a need to develop a framework for governance of data assets. The framework for data governance was proposed in [62] where the authors differentiated between IT assets and information assets. They also proposed a set of five data decision domains such as data principles, data quality, metadata, data access and data life-cycle, and defined what governance type is needed for each domain (see Figure 3.1). According to the authors, data principles establish the extent to which data is an enterprise wide asset, and thus what specific policies, standards and guidelines are appropriate. Data quality refers to its ability to satisfy data usage requirements. Quality has multiple dimensions which were discussed in the previous chapter (see Section 2.3). Metadata, defined as data about data, describes what the data is about and provides a mechanism for a consistent description of data representation, thereby helping interpret the meaning or semantics of data. Data access is premised on the ability of data beneficiaries to assign a value to different categories of data. The data access standards can be based on the definition of unacceptable uses of data and external requirements for the ability to track who/what has accessed/modified data. Understanding how data is used, and how long it must be retained allows organisations to develop approaches that map usage patterns to the optimal storage media, thereby minimizing the total cost of storing data over its life cycle.

Data governance includes also the human aspect of data: who owns data, who stewards it, who defines, produces and uses data across organisations. A good understanding of it can increase an efficient implementation of regulatory and business requirements in order to reduce the cost of information management across organisations. This improves information quality and security, supports decision making processes, and helps to bring organisations into regulatory compliance [10]. These actions are supported by compliance monitoring, standards in information representation and inventories, information management, information risk management and valuation.

3.2 Definition of Model Governance

Predictive models have become business intelligence tools that allow the prediction of specific outcome to support business decisions in various domains. They should be properly validated, reviewed and accepted by regulatory bodies or management boards before it can be used within an organisation. Such acceptance increases trust in model correctness and allows its safe usage. In the real world, there are many good models that can support local decisions and they should be properly captured and managed. According to the authors in [65] a model can now be considered as a valuable informational asset, and its proper usage and application is a candidate subject for governance.

Models can also represent a new source of risk of incorrect decisions. For example, let consider a model that was used to predict a toxic effect of chemicals from a given chemical space. If this model was used incorrectly or has huge error bias then it can cause incorrect decisions on moving some chemicals to the development phase, which causes financial loss related with a cost of their manufacturing and laboratory testing. Another example can represent the situation when a product, discovered using *in-silico* methods, was registered but can not be put into market, because it does not meet human or environmental safety requirements. This can be caused by insufficient documentation or lack of reported validation procedures for used methods in the discovery process. To minimize such situations, governance procedures for models should be strengthened through: assurance that the model is properly used; model improvements to validate and maintain its effectiveness, and understanding the model weaknesses (e.g where the model can be applied safety, how reliable it is). These

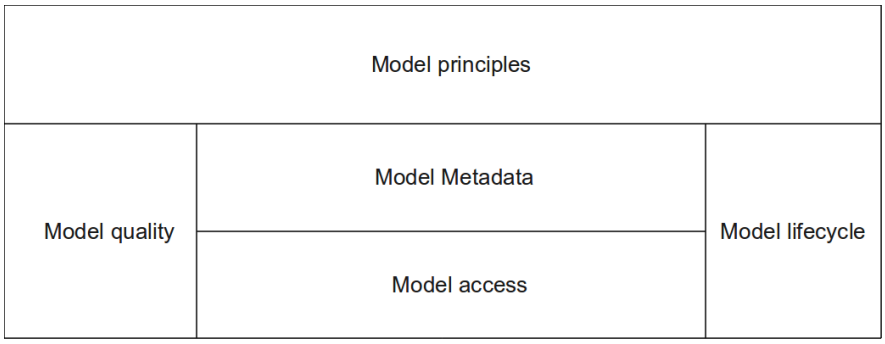


Figure 3.2: Decision domains for model governance.

observations allow us to define the following principles for model governance:

Definition 1. *Model governance is a set of strategies that:*

- *help to ensure models achieve their purposes,*
- *establish model reliance and its importance for organisational use,*
- *establish levels of controls and validation.*

The decision domains for model governance are built extending the framework presented in [62]. They include: model principles, model quality, metadata of a model, access to a model, model life-cycle (see Figure 3.2). Model principles includes polices, standards and guidelines that help to establish which models can become the valuable information assets. Model quality defines the boundary of chemical space where a model gives reliable prediction. It does not include only the assessment of applicability domain and model predictivity, but also an estimation of model reliability for new data. The rules on the assessment of applicability domain as well as model accuracy were extensively discussed in Section 2.4.3. Meta-data about a model provides additional information about model authors, development stages, reviewers, supporting literature, where and how the model has been used. Access to the model defines who and in which situations has access to the model in terms of model development, validation or revision. The life-cycle domain defines the model valuation and verification in order to establish the boundary of the model applicability and its correctness.

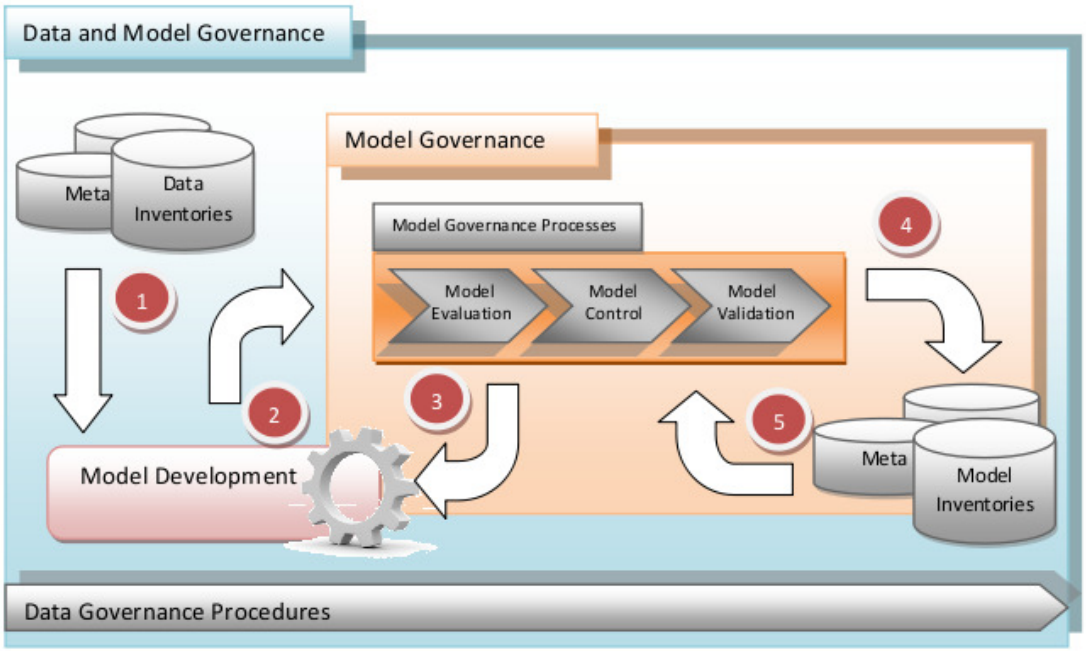


Figure 3.3: Model governance processes within a data and model governance framework.

3.3 Model Governance Processes

Organisations design and implement policies according to internal business goals and regulatory compliance, which follow data governance [20] and model governance (Definition 1) principles. These policies standardise the meaning of model representation and model evaluation. As discussed in the previous section, a model can become a new source of risk of incorrect decisions. This is why the evaluation of a model usage and its proper governance is crucial to increase the reliance on the model and its organisational reuse. These three above mentioned strategies of model governance can be seen as counterparts to the key data governance principles: management, quality and security for the model information context.

The model governance processes are built within a data and model governance framework. Figure 3.3 presents such a framework and described flows of data and model objects and their stewardship within the organisation. The collection of data is standardised and organized according to the internal implementation of data gov-

ernance processes (1). A management board defines the focus areas for model development. Models are developed on high quality data (2). Then model governance processes must be applied to reviewed models within life-cycle management for model development (3). Once reviewed model is accepted by the management board or regulatory body, it is stored in the model repository (4). Stored models in model inventories are the subjects of the continuous verification and validation (5) to maintain their effectiveness in reuse.

For each model object there are three model governance processes: *model evaluation, model control, and model validation* (see Table 3.1). Model evaluation is a process to determine: 1) if the policies for model representation and levels of model controls and validation are suitable for the organisational level of model reuse and control, 2) if validation procedures used for an individual model development process comply with established policies, 3) if the internal model inventories comply with data quality principles. Polices are reviewed according to: model object definition, procedures for model development and validation processes. Inventories are reviewed with respect to: model representation schemas and their consistency, accuracy and completeness among object stored in model inventories.

Model control is a process to determine: 1) if model documentation adequately complies with the established polices, 2) if the model is easily accessed for organisational reuse and it is operating, 3) if security and change control procedures comply with the established polices. During model control processes an unauthorised user access or model reuse is controlled. This is important to verify if model was used within its applicability domain. In situation where the model was not applicable, the alert message should be sent to user or management board. In a case of any updates to the model, the model should be flagged as not validated, and passed again to the model evaluation phase. The documentation from model validation procedures should be checked according to the established policies and regulatory bodies requirements.

Model validation is a process to determine: 1) if methods used in model development phase have mechanistic interpretation, 2) if the implemented methods are accurate and reliable, 3) if model inputs and results are integrated, and 4) how effectively the model is operating. The validation phase provides mechanisms for the verification of model development processes, as well as a data used to generate a model. The review of model results determines how effectively a model is operating.

Table 3.1: Model Governance Processes.

Processes	Rules	Action
Model Evaluation	Policies	1) review model definition 2) review model reuse 3) review model control 4) review model validation
	Inventories	1) review model inventories 2) review representation schemas 3) review model availability 4) review authorisation schemas
Model Control	Changes and Security	1) control access for authorized users 2) control unsafe model reuse 3) control model updates 4) prevent unauthorised access
	Documentation	1) review documentation according to the established policies 2) analyse model limitations and potential weaknesses
Model Validation	Verification	1) review model mechanistic interpretation 2) review model results to determine how effectively a model is operating 3) test model integrity

3.4 Information Management System for Data and Model Governance

Information management system for data and model governance framework is proposed in Figure 3.4. It defines a number of actions for each model governance process. To provide an efficient oversight throughout an organisation, the management board defines policies and strategies that implement current guidelines and regulations established by the regulatory bodies (e.g. REACH). Line management provides adequate controls over data and models and tests model control practices and model validation procedures to ensure compliance with established policies and procedures [10]. Staff and external parties (collaborators) are involved in order to validate that the model is working as intended.

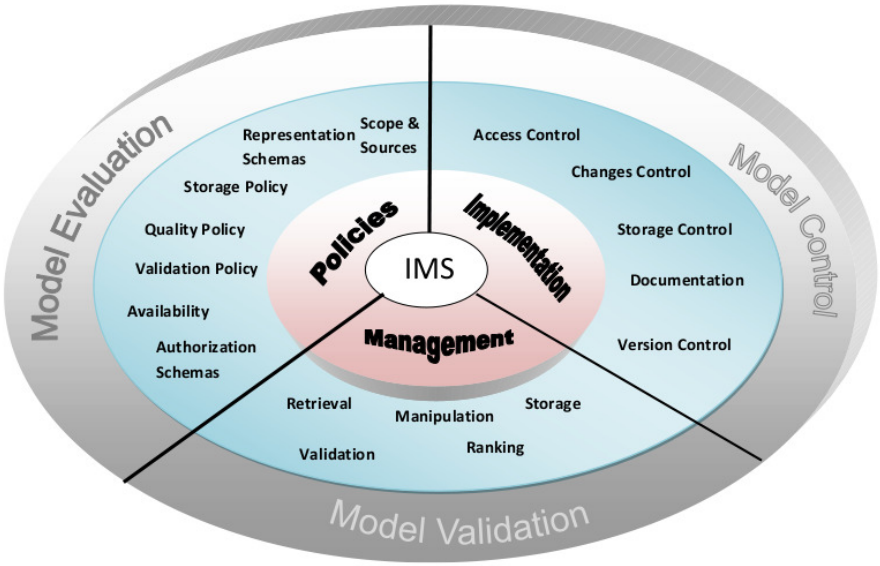


Figure 3.4: Information Management System (IMS) for Model Governance.

3.4.1 Policies

Policies are principles or rules which are defined by regulatory bodies and/or driven by internal business needs in order to: 1) ensure data and models deposited in the database are able to serve their intended purposes, 2) to provide guidelines for user-level control and information management and 3) to deliver proportional oversight throughout the organisation. Multiple sub-domains are considered in terms of policies, including scope & sources, representation schemas, storage policy, quality policy, availability policy and authorisation schemas.

Scope and Sources define what data should and can be held by the database and where the data comes from. This is closely related to the purposes of the database and the availability of resources. *Representation Schemas* provide standards for information context (metadata) and information records representation. This will help to build the databases and interpretable frameworks for information exchange within an organisation. *Storage Policy* gives the procedures that help to decide when the data or models become valuable assets to the organisation. For example, in bio-science laboratories, large quantity of chemicals are profiled on a daily basis and a huge number of them are profiled against the same endpoint in various projects. Often, there is no connec-

tion between these projects, so identification of interesting chemicals or their activity is limited and valuable information can be lost. *Quality Policy* offers several levels of quality filters in order to guarantee the stored data meets its required data needs with respect to different applications requirements. Due to its importance, quality policy is particularly discussed in the next subsection in detail. *Information Availability* concerns how users can access organisational information while maintaining the data integrity. It includes in what way and on what device the data can be accessed, and the possible export formats of data. *Authorisation Schemas* control user access to private and sensitive data based on user privileges. The data might have multi-level user access supported by the pre-defined authorisation policies. The authorisation policy indicates what parts of the data can be accessed/manipulated by whom [30].

3.4.2 Implementation

The implementation aspects of a governance framework for models and data are difficult to define and specify due to their highly individual nature. Each organisation that implements the governance principles will require a unique implementation that is based upon their hardware, software and practices. Specific framework implementations have a significant impact upon the verification of data and model usage, in light of policies, and ensuring data integrity. These policies provide a means for adequate control over data and models objects and their use. Particularly, they allow the evaluation of security and change control procedures, as well as a centralized, systematized method for data and model quality control.

Access control systems implement the availability policies and the authorisation schemas specific to a particular organisation. The detection of unauthorized activity is a powerful tool in protecting the system. This allows system monitoring in order to provide a multi-level defensive capability for the system. Implementation of appropriate limits on users, applications, hardware sources that support operational controls that affect information reliability, accessibility, and timeliness are necessary and important aspects of a complete and usable system. *Change control* is a process used to ensure that changes in data or models are introduced in a controlled and coordinated manner. The new versions of models are reported and monitored by the version control modules and, together with the change control system, they reduce the possibility

of unnecessary changes. *Storage control* implements the information representation schemas and ensures completeness of the stored information according to the quality definitions. The information context should be encoded for the datasets from the experimental protocol and the original data should be stored in the record format. For the models, the information context represents the annotations of the steps in model development and validation. The implementation controls systems together with the documentation framework are components of the quality control system.

3.4.3 Management

Management can be generally defined as making use of information within a governance framework, and ensuring such use is effective, efficient and in line with specified objectives. Figure 3.4 highlights the five main aspects of a information management approach: storage, retrieval, manipulation, ranking and validation. These functional aspects form the core interactive elements for an end user interacting with data and model repositories under a governance framework and subsequently are the key to an effective and useful framework. *Storage* functionality allows users to store or create new information, structured and defined by the organisational polices, using an organisation-wide standardised format. *Retrieval* supports chemical compound identification across various projects or model identification for new chemicals. Model identification involves model selection from existing model collections. This discovered information (data or model) can become a new source for a future business strategies. *Manipulation* supports the result reproduction, maintaining information content by authorized users, reporting information weaknesses (where model provides inaccurate prediction or incorrect content definition for data and models) by end users. *Ranking* functionalities support information comparison (models or data) to support the decision on the various organisational levels. These comparison methods and matrices are defined in quality policies and allow descriptive information comparison (based on the metadata content), and usage information comparison (based on data or model quality measures). *Validation* supports the model governance polices to validate the model representation, quality, usage and storage policies with the business aims and scopes. This also include a validation of the model development processes according to the regulatory body guidances.

3.5 Model Governance in Predictive Toxicology

Data governance is identified as a new challenge in predictive toxicology. There are already existing rules related to scientific experiments, data harvesting or data quality metrics provided by the regulatory bodies. These rules can be implemented in the organisational procedures. Moreover, data governance principles require the development of interoperable and transparent framework that allow the standardisation of the data representation. This become an advantage in data exchange and data integration across various organisations. In [30], the authors discussed gaps in the development of current management frameworks according to data governance principles. They reviewed seven widely used predictive toxicology data sources and applications, with a particular focus on their data governance aspects, including: data accuracy, data completeness, data integrity, metadata and its management, data availability and data authorisation. The authors reveals the current problems and desirable needs of predictive toxicology. Models in predictive toxicology have been considered as an alternative for animal testing. They have become business intelligence tools that allow the prediction of a toxic effect of chemicals on living organisms. This means that a model has to be validated and accepted by regulatory bodies or management boards before it can be used within an organisation (e.g. models in the environmental regulatory decision process [16]).

In Chapter 2 the five OECD principles for QSAR model validation were introduced. These principles are accepted by regulatory bodies and widely implemented in the predictive toxicology domain. They become a base for defining processes for model governance. They also should be incorporated within the information management system for data and model governance framework. One can notice that information access and manipulation is obviously heavily dependent on the underlying data and model representations, and shortcomings in terms of their quality. This will have a significant impact on the ability of a user to effectively reuse of this stored informations. Therefore, the first step towards effective model governance is to create a sufficiently flexible and accessible data and model warehouse. The main aim of such a model inventory is to make original information (such as data and models) available for further data mining usage, analytical processing and decision support. This information is cleaned, transformed and catalogued into data inventories. The content of

the stored information is extracted into metadata representations and loaded into information repositories. It is a crucial step in information integration and information exchange processes for predictive toxicology. Additionally, model governance ensures the quality of collected information. The quality management system embedded within an implementation of a model governance framework is responsible for storing, querying, updating and managing reliable information in efficient ways. It implements a set of structures, processes and strategies which support information quality assurance and quality control. This involves implementation of various quality measurements to define a level of information accuracy, consistency, availability and provenance.

3.6 QSAR Model Representation

The standards of QSAR model exchange format have been studied in [34, 39, 77] and there is not an efficient QSAR model representation. This is caused by the variety of software that allows users the calculation of chemical compound properties and a lack of an uniform descriptor representation [102]. In this section we review the common QSAR model formats existing in the literature and we propose six rules that define minimum necessary information about QSAR model representation.

3.6.1 Exchanged Model Representation Formats

To represent any predictive model the Predictive Model Markup Language (PMML) can be used. It is an XML-based language which provides a way for applications to define statistical and data mining models, to exchange and share models between PMML compliant applications [21]. This format consists of the following elements: header, data dictionary, data transformations, model and mining schema. Header contains general information about the PMML document including: copyright information for the model, software name and version, time-stamp which can be used to specify the date of model creation. Data Dictionary includes definitions for all the possible fields used by the model. This allows the definition of data types and values. Data Transformations consists of transformations that allow for the mapping of user data into a more desirable form which can be used by the mining model. PMML defines several kinds of data transformations. Model contains the definition of the data mining

model. Mining Schema lists all fields used in the model. This can be a subset of the fields defined in the data dictionary. The list may include: attribute name, type of the attribute, outliers treatment, missing values replacement and treatment. Targets allow for post-processing of the predicted value in the format of scaling if the output of the model is continuous.

Another example of the exchange model representation format is QSAR Model Reporting Format (QMRF) [55]. QMRF was released in 2007 and became a standard for the model information representation. It is also used by Ambit and OpenTox. It consists of a set of information about the model development and validation processes in accordance with the OECD QSAR validation principles [77]. This information is encoded also in XML format and includes:

- Model: model title and its identification, pointers to the other relevant models, software used for model development,
- Report: date and authors reporting the model, date of model development, reference to the relevant papers,
- Endpoint: species, target, measurements units, experiment protocol, dependent variable, data quality measurements,
- Algorithm: type of the model, the algorithm description, descriptors used to build a model, feature selection techniques, algorithms and descriptors generation together, software details,
- Applicability domain - model descriptors and their ranges, methods and software used to assess the applicability domain and its threshold,
- Internal validation: dataset descriptions, statistics of "model goodness-of-fit" and model robustness,
- External validation: data set description, statistics of model predictivity.

The QMRF, based on the XML DTD (Document Type Definition) Schema, defines the elements that may be included in the model report document, which attributes these elements have, and the ordering and nesting of the elements. This format does not represent the relation between data used for the model generation and validation and

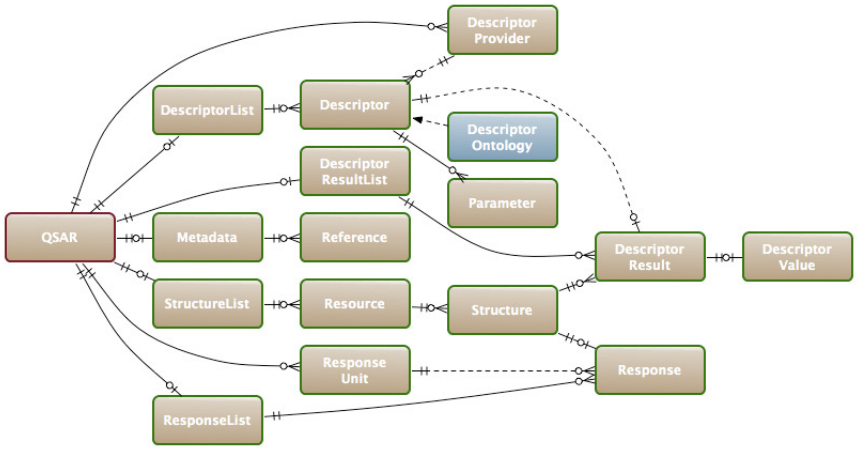


Figure 3.5: The QSAR_ML structure.

the model itself. This is meta-data representation of a model, that can be further used in the provenance analysis.

The latest model representation format QSAR-ML was proposed in [102]. It is an exchange format for QSAR that includes chemical structures, descriptors, software implementations and response values. The framework allows the user to easy set up new QSAR analysis, add molecules, select descriptors and implementations with optional parameters, import and add response values. The structure is illustrated in Figure 3.5 and is build by the following components: Structures - define the chemical structure (this includes InChI to provide the data integrity). Resource - a file referenced by path or URL (it contains a checksum to verify the integrity of the files). StructureList - a list of structure references. Descriptors - define chemical descriptors. DescriptorList - a list of model descriptors. Parameters - a list of model parameters/settings. DescriptorProvider - a version of the software used to calculate descriptors. Responses - define are the measured QSAR endpoints (response variable). ResponseUnit - defines the measured unit like IC50 or LD50. ResponseList - defines a list of measured QSAR points. DescriptorResults - are the results of a descriptor calculation on a structure, and links a DescriptorValue to a Descriptor-Structure. DescriptorResultList - a list of DescriptorResults. Metadata - includes information about authors, license, description, and also contains optional References.

The above discussed formats were designed to provide transparent, interoperable model representation that facilitate the model exchange procedure between the various

toxicity platforms. The PMML model format is the most standard one, but this does not cover the meta information about model provenance and the modelling endpoint. QMRF focuses on the OECD principles without annotating information about model itself. In this case, the unambiguous algorithm must be provided but there no meta information about model structure. And finally QSAR-ML is defining the data used to generate or validate model with the list of the model descriptors and the model predictions. In the next section a model representation format is provided that combines the core elements from the above discussed model formats.

3.6.2 Minimum Information About a QSAR Model Representation (MIAQMR)

Minimum Information about a QSAR Model Representation (MIAQMR), proposed in this section, is a set of rules, that helps to define a model representation satisfying model governance framework requirements as discussed in Section 3.4. These rules encode the minimum necessary information required to define model objects in the governance framework including: model provenance, model description, dataset provenance, model development, model reliability and predictivity. All these rules comply with the current OECD requirements [77] for *in-silico* modelling and are inspired by these principles. They also form the basis for the XML model representation development called MIAQMR-ML. The general schema including the conceptual and data layer is presented in Figure 3.6. The data layer is a simplified diagram of the model representations. Each element corresponds to a rule that defines the minimum required information. There are in total six rules discussed below.

RULE I: Model Provenance
A model must comply with the following standards: authors must be clearly defined; timestamp of model creation, submission and updates must be provided; one or more referenced sources that describe the model development process should be included.

Model provenance defines in a very general way when and by whom the model was created or developed. Listing 3.1 defines the XML schema for the provenance element. The model representation must include an author list, contact information

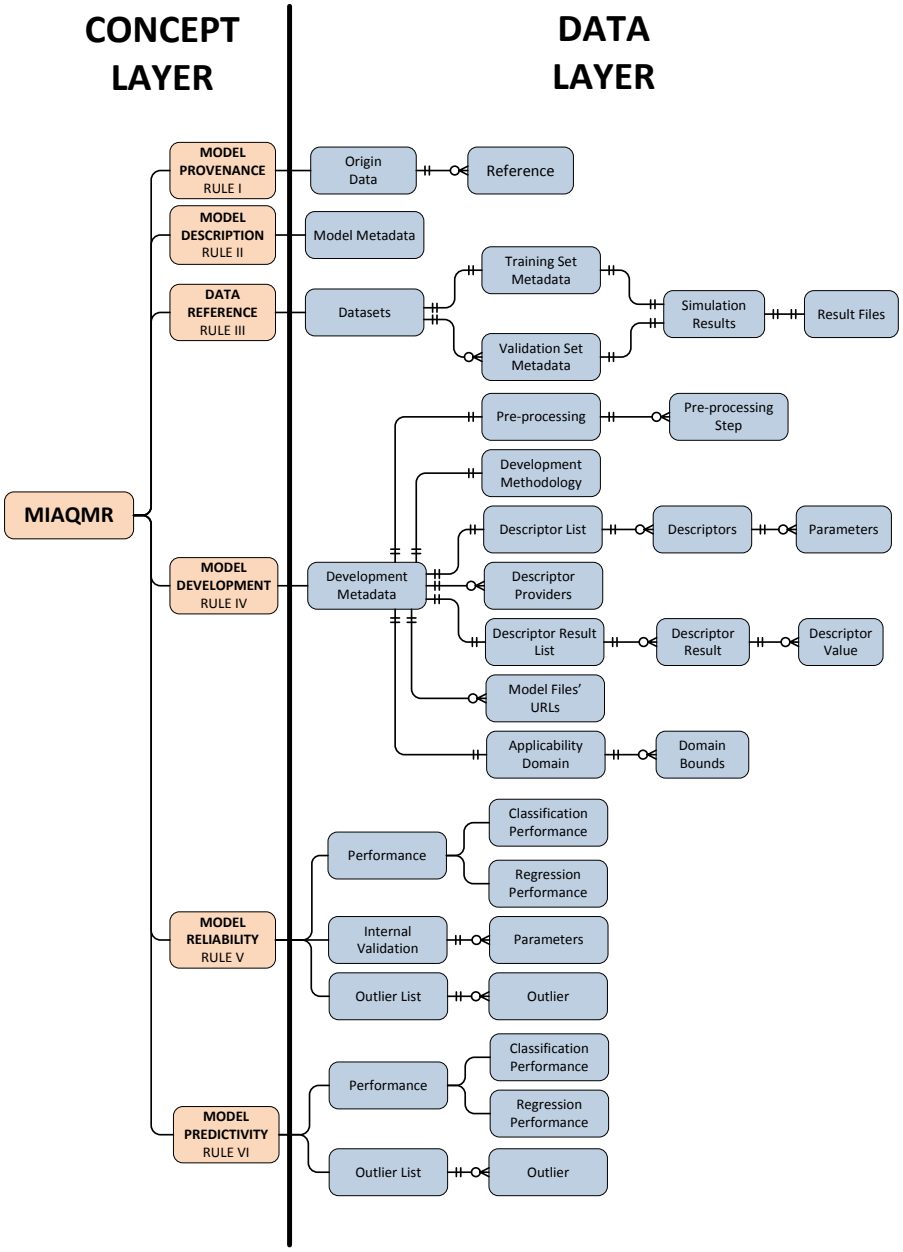


Figure 3.6: MIAQMR-ML schema.

and the submission date. This assists with model maintenance and validation, as other users may report errors or unexpected model behaviour directly to the authors. The model version and timestamp must be included to support users who wish to know about the model provenance. Model development and last modification timestamps help users to monitor any changes that may be made by authors. Too many changes in a short time may impact on the degree of trust in model predictivity. Additionally, a model should not be available to users before a stable version is released. The quality and version control system will monitor the updates and error reports and mark models for an external review. A model must include corresponding references for model quality evaluation. These references should include the data collections and the model development procedures written as an organisational documentation, white or scientific papers.

Listing 3.1: Model provenance element of the MIAQMR-ML.xsd

```
<complexType name="ProvenanceType">
  <sequence>
    <element name="Reference" ></element>
  </sequence>
  <attribute name="Authors"/>
  <attribute name="ContactDetails"/>
  <attribute name="SubmissionDate" />
  <attribute name="Version" />
  <attribute name="LastModifycationDate"/>
  <attribute name="CreateDate"/>
</complexType>
```

RULE II: Model Description
A model must comply with the 1st OECD QSAR model validation principle and have a clearly defined endpoint name and type.

According to Rule II, model representation must include at least minimal assay information such as: endpoint, units and species (see Listing 3.2). The endpoint and species should be standardised to avoid inconsistency in data representation. The model description does not include the full experiment protocol description only the

exposure time should be recorded. Such information is required to report the model to the regulatory bodies but in the governance framework this information can be recovered from the dataset or assays descriptions. The model representation must include type and name. The name of the model should be standardised over the systems. Two models for the same endpoint should have the same endpoint name. This allows for unique models identification for a given endpoint. The model representation must include the dependent variable. This is because the dependent variable being modelled can be different from the measured endpoint. The model type annotation and a clear definition of the dependent variables will be used in the model selection and comparison process.

Listing 3.2: Model description element of MIAQMR-ML.xsd

```
<complexType name="DescriptionType">
 <attribute name="ModelName"/>
 <attribute name="Species"/>
 <attribute name="Endpoint"/>
 <attribute name="Unit"/>
 <attribute name="ExposureTime"/>
 <attribute name="ModelType">
 <attribute name="DependentVariable"/>
</complexType>
```

RULE III: Data Reference
A model must have references provided for datasets used in model development and validation phases, supporting compliance with the 3rd OECD QSAR model validation principle.

According to Rule III, model representation must have associated training and validation datasets (see Listing 3.3). In the case when cross-validation methods such as LOO, LMO, K-fold etc. are used, the validation dataset can be empty. This, however, can cause a reduction in model quality. Conversely, a model may have more than one validation dataset. Both datasets must include dataset provenance (including authors, timestamp and references) and number of chemicals. The dataset reference is necessary to recover information about the assay and the experimental protocol and the

number of chemicals used for statistical model comparison. Together with the information about applicability domain, this may provide information about how well the model can be applied across a wide chemical sub-space. These datasets should be submitted in file format, together with the model, or be accessible via a provided link. The simulation results must be included for model submission. This is required in order to facilitate model identification and toxicity assessment without applying a model for new chemicals. To clarify, the predicted toxicity measurements are only stored in the files and used for the model performance analysis. The chemicals are related to the models via datasets.

Listing 3.3: Model datasets element of MIAQMR-ML.xsd

```
<complexType name="DatasetType">
  <sequence>
    <element name="TrainingDataset"> </element>
    <element name="ValidationDataset"> </element>
   </sequence>
</complexType>
<complexType name="DatasetProvenanceType">
  <sequence>
    <element name="SimulationResult"></element>
  </sequence>
  <attribute name="DatasetAuthors" />
  <attribute name="DatasetLastModifyDate" />
  <attribute name="DatasetURL" />
  <attribute name="NumberOfChemicals" />
  </complexType>
<complexType name="ResultType">
  <attribute name="FileName"/>
  <attribute name="FileURL" />
</complexType>
```

RULE IV: Model Development
A model must refer to the model development process to comply with the 2nd and 3rd OECD QSAR model validation principles, and to support compliance with the 5th OECD QSAR model validation principle.

Rule IV ensures transparency in the model algorithm within the model governance framework. This information has to be available for the quality control system to establish model performance according to the quality policies. This also allows the reproduction of predictions as it includes information about data pre-processing, the algorithm, attributes/features used in the model development process and their provider, the access to the model executable file or to the tools that may be used for the further predictions and the applicability domain. Often, model supporting information published in the literature is not sufficient to reproduce model results. Thus the model must provide information about whether the dataset has been pre-processed or not. If the dataset has been pre-processed, the feature selection algorithm that was used should be included (see Listing 3.4). The method name must be defined. The name of the software and its version must be specified. The method description must be provided including a list of parameters. Such information will help users to reproduce the model using the same or different modelling tools.

Listing 3.4: Model preprocessing element of MIAQMR-ML.xsd

```
<complexType name="PreprocessingType">
  <sequence>
    <element name="PreprocessingStep"> </element>
  </sequence>
</complexType>
<complexType name="PreprocessingStepType">
  <attribute name="Name"/>
  <attribute name="Description"/>
</complexType>
```

The model representation must define model features (descriptors). This imposes a requirement to provide the descriptor list used in the model development process. Descriptors can be calculated from molecular structure, e.g. molecular weight, or measured such as logP. In this case, when descriptors values were calculated from a structure, the software (including version and parameters) from which the descriptor values are derived is required (see Listing 3.5). Such captured information allows users to reproduce results or to find a collection of models for a given endpoint. These models may be based on different feature sets and their combination can improve a future prediction for new chemical compounds.

Listing 3.5: Model descriptor type element of MIAQMR-ML.xsd

```
<complexType name="DescriptorListType">
  <sequence>  <element name="Descriptor"> </element>
  </sequence>
  <attribute name="Name"/>
  <attribute name="Description"/>
</complexType>
<complexType name="DescriptorType">
  <sequence> <element name="Parameter"> </element>
  </sequence>
  <attribute name="Name"/>
  <attribute name="Type">
  <attribute name="Provider" />
  <attribute name="Description"/>
</complexType>
<complexType name="ParameterType">
  <attribute name="Key"/>
  <attribute name="Value"/>
</complexType>
<complexType name="DescriptorProviderType">
  <attribute name="URL"/>
  <attribute name="Name" />
  <attribute name="Vendor"/>
  <attribute name="Version"/>
</complexType>
<complexType name="DescriptorResultListsType">
<sequence> <element name="DescriptorResult"> </element>
  </sequence>
</complexType>
<complexType name="DescriptorResultType">
  <sequence> <element name="DescriptorValue"> </element>
  </sequence>
  <attribute name="SubstanceID"/>
  <attribute name="DescriptorID"/>
</complexType>
<complexType name="DescriptorValueType">
  <attribute name="Index"/>
  <attribute name="Label" />
  <attribute name="Value" />
</complexType>
```

The model representation must have defined an applicability domain, where the model produces reliable predictions. This includes information about the applicability domain estimation process and representation of applicability domain (see Listing 3.6). The model has to be represented in a machine readable format such as files for MS Excel, PMML, ARFF, R, Java, C etc. It is required to submit actual model files when submitting a model to the system.

Listing 3.6: Model applicability domain element of MIAQMR-ML.xsd

```
<complexType name="ApplicabilityDomainType">
  <sequence>
    <element name="ADelement" </element>
  </sequence>
  <attribute name="Process"/>
</complexType>
<complexType name="ADelementType">
  <attribute name="Label"/>
  <attribute name="LowerBound" />
  <attribute name="UpperBound" />
</complexType>
```

RULE V: Model Reliability
The model reliability must define the model ability to make predictions from the training dataset. Models must have a defined type, the model performance matrices, and a list of outlier chemicals clearly defined.

According to Rule V, the internal validation method must be specified, such as cross validation, LOO, LMO, boosting etc (see Listing 3.7). For classification models, the following performance matrices are required: confusion matrix, precision/recall, overall accuracy, sensitivity, specificity and errors. For regression models, the following performance matrices are required: R-squared (r^2), Q-squared (q^2) [114] and errors. The model must provide a list of outliers. While the simulation results can be stored, these values do not have to be provided by authors. Nevertheless, they can be calculated from the model and stored in the model representation format.

Listing 3.7: Model reliability element of MIAQMR-ML.xsd

```
<complexType name="ReliabilityType">
  <sequence>
    <element name="Performance"> </element>
    <element name="InternalValidation" > </element>
    <element name="OutlierList" ></element>
  </sequence>
</complexType>
<complexType name="InternalValidationType">
  <sequence>
    <element name="Parameter"></element>
  </sequence>
  <attribute name="ValidationType">
      <simpleType>
   <restriction base="string">
     <enumeration value="CrossValidation"/>
     <enumeration value="LOO"/>
     <enumeration value="LMO"/>
     <enumeration value="Boosting"/>
     <enumeration value="Other"/>
   </restriction>
      </simpleType>
    </attribute>
</complexType>
<complexType name="OutlierListType">
  <sequence>
    <element name="Outlier" ></element>
  </sequence>
    <attribute name="NomberOfOoutlier"/>
</complexType>
<complexType name="OutlierType">
    <attribute name="SubstanceID" type="string"/>
</complexType>
<complexType name="PerformanceType">
  <choice>
    <element name="ClassificationPerformance"> </element>
    <element name="RegressionPerformance"> </element>
  </choice>
</complexType>
```

RULE VI: Model Predictivity
The model predictivity must define the model ability to make predictions from outside the training data. Models must have a defined type, model performance statistics, and a list of the chemicals defined as outliers.

According to RULE VI and similar to the reliability rule, the model representation should include information about the model performance for the validation dataset (see Listing 3.8). For classification models, the following performance matrices are required: confusion matrix, precision/recall, overall accuracy, sensitivity, and specificity. For regression models, the following performance matrices are required: R-squared (r^2), Q-squared (q^2) and errors such as mean squared error, etc. The model must also provide a list of outliers. If a model has been validated on multiple validation datasets, each should have their own performance matrices. These values can be calculated from the simulation result, but it is important to encode it in a model representation to ensure the model representation consistency.

Listing 3.8: Model predictivity element of the MIAQMR-ML schema

```
<complexType name="PredictivityType">
   <sequence>
       <element name="Performance"> </element>
            <element name="Outlierlist"> </element>
   </sequence>
   <attribute name="ADextrapolationComment" type="string"/>
</complexType>
```

3.7 Model and Data Farm (MADFARM) Prototype

To provide a proof-of-concept for model governance framework, the Model and Data Farm (MADFARM) prototype has been developed. MADFARM is a web application to access models and datasets for predictive toxicology. This application has been implemented for internal use in Syngenta [83] and is still under development to provide the whole functionality that was discussed in this chapter. As of 2013, it includes an implementation of the MIAQMR-ML format for model representation.

3.7.1 MADFARM Design Principles

The design for the MADFARM pilot platform carefully considers the OECD principles for QSAR acceptance in predictive toxicology. As an information system, the design steps were: data representation, database design based on the data implementation, and user interface creation. Additionally, the proposed framework is built upon the following key design principles which guide both the design and implementation:

- Flexibility - employs a multi-level user access mechanism which allows for a variety of user scenarios, use cases and requirements. In addition, the system is delivered in a web application format, the users can easily access the system within the organisation. This also provides the flexibility to make changes or updates to the system, no extra efforts is required from the users.

- Extensibility - the design is guided by an object-oriented approach, so that it is easy to add new components/objects into the existing framework.

- Transparency - proposes a detailed metadata representation schema which is based on the OCED principles. This helps to increase the transparency and interpretability of such framework.

- Reliability - there is a quality check mechanism to ensure the stored datasets and models are of high quality. Any included model has to go through the model evaluation - model control - model validation process. The metadata of dataset and model are well captured and stored. Multi-level user access and authorisation is implemented to protect sensitive data to make the framework more reliable and trustful. Version control and change control also ensures that the users can only access the permitted information.

- Reusability - users can rebuild a model by using the stored metadata and parameters. The representation schema is represented in XML format which can easily be exchanged over the Internet. Most existing frameworks enable users to re-build stored models within their own frameworks, but not using external tools. The MADFARM framework not only stores model metadata, but also the executable model files. In addition, the training/validation datasets are available.

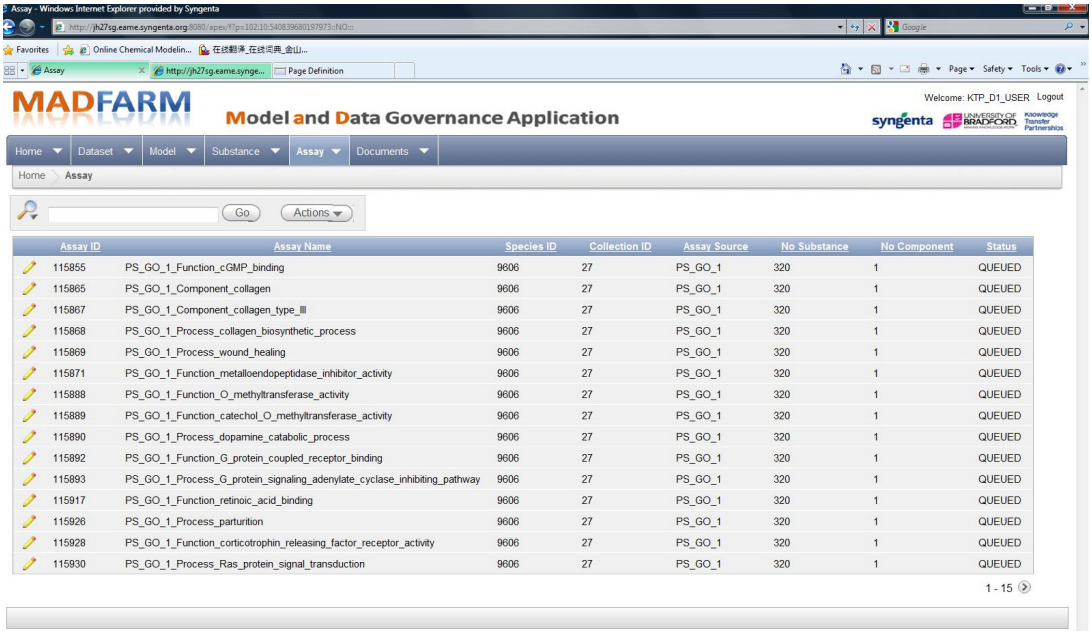


Figure 3.7: MADFARM - Browse Assay Interface.

3.7.2 MADFARM Web Interface

Currently, 3486 substances, 5482 assays, 5571 assay components, 9104 species and 71 endpoints are stored in MADFARM. Data comes from various databases such as: ToxRefDB, DSSTOX, TETRATOX, CAESAR and Syngenta's internal databases. The system allows users to browse information and search for chemical compounds, assays and datasets. Users can submit new assays and substances but, before data is integrated quality procedures are applied to ensure their consistency and correctness. Figure 3.7 presents the search engine for an assay.

From the collection of assays, users can create their own datasets for further analysis or modelling. Once the dataset is created, it can be stored in MADFARM (see Figure 3.8). During submission, the metadata about a dataset are captured according to RULE III (see above section). MADFARM provides a specially designed entry tool, where all required information about a dataset has to be provided. The dataset should be submitted as supporting information for a model if users wish to submit the model into the system. Currently, there are two models for polar/nonpolar narcosis for TETRATOX taken from JRC QSAR DB and two models for BCF and developmental

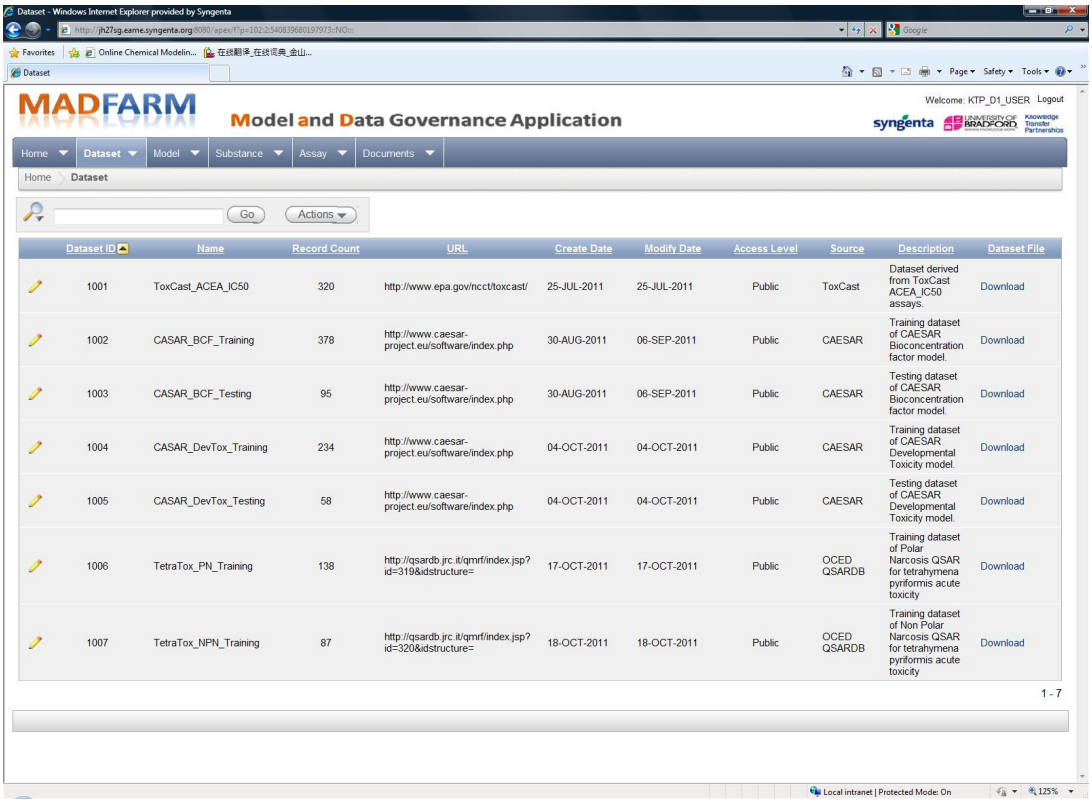


Figure 3.8: MADFARM - Browse Dataset Interface.

toxicity from the CAESAR system. Additionally in MADFARM there are 31 models developed in Syngenta for the ToxCast dataset. Figure 3.9 presents the model search engine. The user can browse existing models in MADFARM and use them to predict toxicity for new chemical compounds.

The model submission page has been built according to the rules discussed in the previous section. There is a specially designed template where the user is required to provide all information about authors, models, datasets and simulation results. To ensure the completeness of provided information, the user is asked to follow the submission protocol. Once the model is submitted, it has to be reviewed according to the model quality policies to ensure completeness of information before it becomes available to other users.

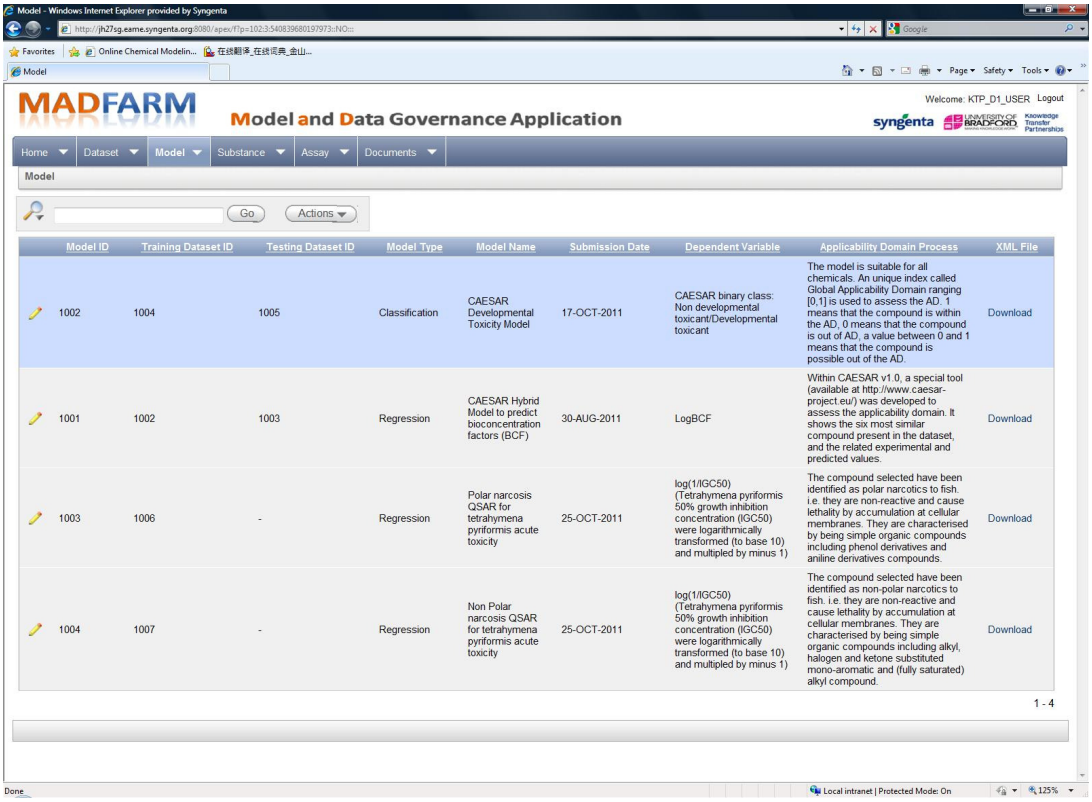


Figure 3.9: MADFARM - Browse Model Interface.

3.8 Summary

Models are powerful tools supporting decision making processes in various domains such as economics, finance, pharmacy, and biosciences to name a few. They are accepted as intelligent decision making tools and are commonly viewed as valuable business-wide assets. However, in the real world there are (local) models designed to work well within defined narrow applicability domains. They are too specific in their application to become general business tools, but are useful enough to become valuable assets within organisations. Despite this, very often they are lost within communities instead of being validated and subsequently used to support future decisions.

In this chapter a new concept of model governance in the predictive toxicology domain was proposed. The model governance processes were discussed in detail, and the information management framework for data and model governance was introduced.

The main challenge is how to bring models to life to enhance and improve their availability and to provide the means to assess their potential utility for future decision making. To achieve this, a suitable model representation is required. In this chapter the minimum information about a QSAR model representation (MIAQMR) was defined. This representation combines and extends elements included in existing model representations to provide a transparent and sufficient model object representation. An XML schema was proposed for the MIAQMR mark-up language. As a proof of concept, the prototype of the MADFARM system was presented. The MADFARM has been developed in collaboration with Dr. Xin Fu and Dr. Richard Marchese Robinson that were KTP Research Associates in Syngenta.

In MADFARM, a process of model reuse is manual. Users have to compare models to find the most suitable ones. There is a need to provide automated or semi-automated methods that support decision on a model usage and its analysis. The following chapters will address these issues.

Chapter 4

Model Identification

Reuse of information and existing models in predictive toxicology is important due to the current focus on the reduction of animal testing and the cost of new product development. The decision to use a model is left to users and it is based on their trusts that the model is accurate. This chapter introduces a new concept of semi-automated model identification. Having a collection of good quality models we would like to identify the most suitable model for a given chemical compound. To solve this problem we propose a partitioning model. To construct such model two method are introduced: the first based on the nearest neighbour and the second based on a Pareto neighbourhood. This chapter defines a theoretical framework and proposes algorithms for the model identification process. Experimental work shows that the proposed approach provides good results. The work presented in this chapter was published in [84] and [122].

4.1 Introduction

There is an extensive literature associated with the best practice for model generation and data integration (see Section 2.4), but management and identification of relevant models from available collections of models is still an open problem. In recent years a large number of highly predictive models, having various applicability domains, has become publicly available. Some of them, tested on a wide chemical space, have become officially approved tools, e.g. KOWWIN (estimates the log octanol-water partition coefficient) or BCFBAF (estimates fish bioconcentration factor) built into Es-

timation Program Interface (EPI) Suite [28]. There is also a large number of quality models that are applicable only for a narrow chemical space. Some of them are annotated according to the OECD principles and publicly available in databases like the JRC QSAR Models Database [54]. Models that are stored in model databases can be reused to predict toxicity of new chemical compounds. Unfortunately, this involves a manual process of model selection. A potential user is required to make a comparison of model applicability domains and their predictivity for a given activity in order to decide if the model can make reliable predictions for a given chemical compound. Model comparison is a difficult task since models are generated using various subsets or various chemical compound descriptors. Consequently, models can be trained and validated on different datasets. Several model performance matrices have been analysed in the context of model validation and model selection in [66]. They are applied in automated model development where models are validated by the same dataset [124]. In the case where two models come from different sources, model comparison becomes challenging. This requires predictive models to be validated across the entire chemical space, which is very difficult as the list of available chemicals and assays can be limited. Clearly, there is a need for automated techniques for mining model repositories including methods for model quality control, data and model integration, model identification, and model interpretation. To partially address these issues, this chapter proposes a mathematical framework for model identification in predictive toxicology.

In engineering, the term "*model identification*" refers to the system identification that uses statistical methods to build mathematical models of dynamic systems from measured data. In this chapter, the term "*model identification*" is used to cover the whole range of problems related to model selection from a collection of existing models (for a given endpoint) developed on various datasets and in the different time. In the extreme case, datasets (and specified applicability domains) for two models can be disjoint (ie. non-overlapping). Model identification is a much harder problem than the well known model selection problem [64], i.e. choosing a model from a set of candidate models with the same applicability domain. Therefore, various methods applied in traditional model selection [49, 66, 73, 111] cannot be directly applied to model identification. In contrast to model selection, model identification cannot take into account all model variables or parameters since some model variables cannot be easily accessed for new chemical compounds.

4.2 Partitioning Model

The idea of model identification is simply based on a partition of a chemical space into groups according to a similarity of elements in this space and model performances. Each group is assigned with a model that gives reliable prediction for all elements from such group. In this section the basic definitions for a mathematical framework for model identification are introduced.

Definition 4.1. *A chemical space X is defined as a set of pairs $x = (x^d, x^f)$, where $x^d \in \mathbb{R}^{K_1}$ represents descriptors, $x^f \in \{0,1\}^{K_2}$ is a fingerprint, and $K_1 + K_2$ is the dimension of the chemical space.*

Descriptors represent various topological, geometrical, physiochemical properties of a chemical compound. A fingerprint is a binary vector whose coordinates define the presence or absence of predefined structural fragments within a molecule. A fingerprint is also a one dimensional representation of the molecular descriptor and it is widely used for chemical similarity search in large databases. Detailed discussion of chemical representation was presented in Section 2.2.1. It is also worth noting that a fingerprint is not a unique chemical compound representation because it encodes only a fragment of a molecule. There can be two different molecules having the same fingerprint representation.

Definition 4.2. *A predictive model M is a mapping $X \to Y$, where $x \subset \mathbb{R}_d$ is a chemical space and $Y \subset \mathbb{R}$ is the output space.*

The output space Y might, for example, represent a particular biological, physical or chemical activity of a chemical compound. For such defined models, an input data is represented by the pairs:

$$(x_i, y_i) \in X \times Y, \qquad i = 1, \ldots, n,$$

where x_i is an element of the chemical space and y_i is the measured activity of that element. There is also a set of m predictive models $\mathcal{M} = \{M_1, \ldots, M_m\}$ associated with the activity Y. These models were generated using various statistical or data mining techniques and they have different applicability domains and performances. To

identify the most predictive model from the collection of models $\mathcal{M}$ for a new chemical compound $x \in X$, a partitioning model is defined.

Definition 4.3. *A* partitioning model $\hat{M}$ *is a mapping* $X \to Y$ *given by the following formula:*

$$\hat{M}(x) = \begin{cases} M_1(x), & x \in D_1, \\ M_2(x), & x \in D_2, \\ \vdots, & \vdots, \\ M_m(x), & x \in D_m, \end{cases}$$

where

- $D_1, \ldots, D_m \subseteq X$ *are disjoint,*
- $\bigcup_{i=1}^{m} D_i = X$.

A computer representation of the partitioning model is a demanding task due to the size of the chemical space – one has to store the sets $D_1, \ldots, D_m$. The partitioning model aims at dividing the chemical space in such a way that every element $x \in X$ is assigned to the model, from the set of available models, with the highest predictive power. This task is clearly infeasible as the set X is large whereas available information is limited. Therefore, there is a need to concentrate on approximate solutions to build the partitioning model.

The construction of the partitioning model is a similarity-based classification problem, that assigns a given chemical compound to the most predictive model. The similarity-based classifier estimates the class label of a test item using similarities between the test item and a set of labelled training items [15]. While most learning methods derive a set of classification rules from training data, in this work the classification is obtained by applying a pre-defined classification function on a given dataset. This function is a combination of the chemical compounds similarity and model performance. According to Definition 4.3, the partitioning model splits the chemical space in groups in order to maximize the similarity of their chemical compounds and to minimize the error of a model associated with this group. Let us call such a group - *a model group*. It is easy to notice that this is a bi-criteria problem and the solutions have to represent a trade-off between optimality of these criteria (the so-called Pareto

Algorithm 1 Double Min-Score Algorithm

Input: A dataset T, a family of models $\mathcal{M}_T$ and a new data x.
Output: The most predictive model M.
Step 1: Calculate the error $e_{i,j}$ for every model M_j and every item x_i in dataset T.
Step 2: Split the dataset T into m disjoint model groups.
Step 3: Calculate the nearest neighbourhood of x.
Step 4: Select the model M_j assigned with the nearest neighbour of x.

points[23]). Pareto optimality is a multi-criteria optimisation problem widely used in decision-making. The usage of Pareto points for model identification in predictive toxicology will be presented later in this chapter (see Section 4.4.2).

4.3 Double Min-Score Algorithm

The construction of the partitioning model, as mentioned in the previous section, is a bi-criteria problem. For simplicity, this problem can be reduced to be a one-criteria problem based on the chemical compound similarity hypothesis [51] which states that similar compounds have similar properties. The mapping between the chemical space and a set of model indexes is defined using the Double Min-Score (DMS) algorithm (see Algorithm 1).

Let's consider a dataset T of pairs $(x_i, y_i) \in X \times Y$, where $i = 1, \ldots, n$, and the family of predictive models $\mathcal{M}_T$. In Step 1 of the DMS algorithm presented above, the error $e_{i,j}$ of the model M_j for the i-th data item is defined as follows:

$$e_{i,j} = |y_i - M_j(x_i)|, \tag{4.1}$$

where $i = 1, \ldots, n$ and $M_j \in \mathcal{M}_T$ for $j = 1, \ldots, m$.

In the next step a mapping of the chemical space into a set of model indexes $D : X \leftarrow \{1, 2, \ldots, m\}$ is defined. Firstly, a mapping D on the dataset T is considered in such a way that for each $x_i \in T$:

$$D(x_i) = \min\{j \in \{1, 2, \ldots, m\} : e_{i,j} = \min\{e_{i,l} : l = 1, \ldots, m\}\} \tag{4.2}$$

In this step, a class (a model index) is defined for elements in the dataset T. In Step

2 of Algorithm 1 a dataset is divided into m disjoint sets. According to formula (4.2), each data item $x_i \in T$ is assigned to the model that has the minimal error defined by formula (4.1) over all available models. In case where more than one model has the same predictive error, the model with the lowest index is chosen.

In the next step, the mapping D is extended to the whole chemical space X in the following way: for $x \in X$

$$D(x) = \min\{D(x_i) : \rho(x^f, x_i^f) = \min\{\rho(x^f, x_k^f) : k = 1, \ldots, n\}, i = 1, \ldots, n\} \tag{4.3}$$

where $D(x_i)$ is defined by formula (4.2) and ρ is the fingerprint-based similarity coefficient (widely used in chemical similarity searching [121]). The DMS algorithm uses only the molecular similarity of chemicals and does not require knowledge of the model applicability domain. In this stage (Steps 3-4, Algorithm 1) the nearest neighbourhood of x is calculated. Then, the element x is assigned to the model group of its nearest neighbour x_i according to formula (4.3). The selected model can be applied on x to predict its activity y.

It is worth noting that the automated model selection framework can also be used for the applicability domain estimation. The partitioning model groups chemicals according to the model performance, and then ranges for model descriptors can be easily obtained from the chemical space X.

4.4 Algorithms Based on Pareto Order

The identification of the most reliable model from the collection of models for new chemicals is a challenging task. As was mentioned in Section 4.2 it is a bi-criteria optimisation problem. The solution must to be a trade-off between the chemical similarity and the model performances. This means that there can be more than one solution.

This section has three main subsections. The first introduces new properties of the Pareto order. The second presents a new method for finding the set of optimal Pareto sets of candidate models. In the last section the Pareto neighbourhood is discussed and two methods for unambiguous model identification are presented.

4.4.1 Pareto Optimality

Pareto optimality, often called Pareto efficiency, is named after 20-th century Italian economist Vilfredo Pareto, who studied income distribution and the analysis of individuals' choices. He introduced the concept of Pareto efficiency, widely used in economics and engineering, to find solutions for multi-criteria optimisation problems [24]. Multi-criteria optimisation involves more than one objective function to be optimized simultaneously and the methods are applied where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. In this concept, the maximization of the similarity within a model group and minimization of the model error associated with the group are the main two objective functions.

In engineering, a set of solutions/choices that are Pareto efficient is very often called the Pareto frontier or Pareto set. In this work the Pareto set is used to define a set of candidate models for new chemicals. The following sections include a recall of the Pareto order and proposes its new properties.

4.4.1.1 Pareto Set

Let us consider a vector $v = [f_1, f_2, \ldots, f_K]$ in the K-dimensional space. Let $\pi_j(v) = f_j$ denote a j-th coordinate of vector v and V be a finite set of vectors in $\mathbb{R}^K$.

Definition 2 (Domination)**.** *A vector $v \in \mathbb{R}^K$ is dominated by a vector $w \in \mathbb{R}^K$, which is denoted by $v \preceq w$, if*

$$\pi_j(v) \leq \pi_j(w), \quad \forall j = 1, \ldots, K. \tag{4.4}$$

One can say that v is strictly dominated by w, $v \prec w$, if $v \preceq w$ and $v \neq w$, i.e.

$$\forall j = 1, \ldots, K \quad \pi_j(v) \leq \pi_j(w), \quad \exists_{j=1,\ldots,K} \quad \pi_j(v) < \pi_j(w). \tag{4.5}$$

Definition 3 (Comparison)**.** *Vectors $v, w \in \mathbb{R}^K$ are incomparable, which we denote by $v \sim w$, if neither $v \preceq w$ nor $w \preceq v$.*

Notice that $v \sim w$ if and only if there exist $i, j \in \{1, \ldots, K\}$, $i \neq j$, such that

$$\pi_i(v) < \pi_i(w) \quad \text{and} \quad \pi_j(v) > \pi_j(w). \tag{4.6}$$

Definition 4 (Pareto set)**.** *A set $\Gamma \subset V$ of minimal vectors with respect to $\preceq$ is called a Pareto set for V.*

Note that Γ consists of incomparable vectors. Then one can define Γ equivalently by the formula

$$\Gamma = \{v \in V : \forall_{w \in V} \quad v \preceq w \vee v \sim w\}. \tag{4.7}$$

The above definitions and basic properties of the Pareto set can be found in [110]. Below there are defined some properties of Pareto sets and Pareto order that are used in the following sections. First, a convenient notation is introduced. Let

$$f_j^{min} := \min\{\pi_j(v) : v \in V\}, \quad j = 1, \ldots, K, \tag{4.8}$$

and

$$V_j := \{v \in V : \pi_j(v) = f_j^{min}\}, \quad j = 1, \ldots, K. \tag{4.9}$$

The set V_j consists of all vectors in V with minimal value on the j-th coordinate.

Lemma 1. *Let Γ_j be the set of all minimal vectors in V_j. Then $\Gamma_j \subset \Gamma$, where Γ is the Pareto set for V.*

Proof. One can prove this lemma by contradiction. Let's $j \in \{1, \ldots, K\}$ and choose $v \in \Gamma_j$. Assume that $v \notin \Gamma$, which is equivalent to saying that there exists $w \in V$ that is strictly dominated by v, i.e. $w \prec v$. This means that $\pi_j(w) = \pi_j(v)$ and $w \in V_j$. By the definition of Γ_j we know that v is a minimal vector in V_j, so $v \preceq w$, which contradicts $w \prec v$. □

Let $I\Gamma = \bigcup_{j=1,\ldots,K} \Gamma_j$ and

$$f_j^{max} := max\{\pi_j(v) : v \in I\Gamma\}, \quad j = 1, \ldots, K. \tag{4.10}$$

In particular one can notice that $I\Gamma$ is a subset of Γ and it is called an *initial Pareto set*. The next lemma establishes the dependence of the conditions for incomparability with vectors in this initial Pareto set.

Lemma 2. *If a vector $v \in V$ is incomparable with all vectors in $\mathrm{I}\Gamma$, then there exist at least two indices $j \in \{1, \ldots, K\}$ such that*

$$\pi_j(v) \in (f_j^{min}, f_j^{max}). \tag{4.11}$$

Proof. Let $v \in V$. First notice that $\pi_j(v) \geq f_j^{min}$, $j = 1, \ldots, K$. If $\pi_j(v) \notin (f_j^{min}, f_j^{max})$ for all j then $\pi_j(v) \geq f_j^{max}$ for all j and $w \preceq v$ for $w \in \mathrm{I}\Gamma$. If there exists exactly one $j \in \{1, \ldots, K\}$ such that $\pi_j(v) \in (f_j^{min}, f_j^{max})$, then for each index $l \neq j$ we have $\pi_l(v) \geq f_l^{max}$ and there exists a vector $w \in \Gamma_j$ such that $w \preceq v$. Therefore, if v is incomparable with vectors in $\mathrm{I}\Gamma$, none of the above cases can take place, and the proof is completed. □

4.4.1.2 Pareto Order in Two Dimensions

This subsection is devoted to study of the two-dimensional case, i.e. $K = 2$. We shall use the notation introduced above.

Lemma 3. *The set $\mathrm{I}\Gamma$ has at most two elements.*

1. *If $|\mathrm{I}\Gamma| = 1$, then $\mathrm{I}\Gamma$ is the Pareto set for V.*
2. *If $|\mathrm{I}\Gamma| = 2$, then a vector $v \in V$ is incomparable with vectors in $\mathrm{I}\Gamma$ if and only if*

$$\forall_{j=1,2} \quad \pi_j(v) \in (f_j^{min}, f_j^{max}). \tag{4.12}$$

Proof. Notice first that each Γ_j, $j = 1, 2$, consists of one element, because the Pareto order $\preceq$ induces a linear order on the sets V_j. Therefore, $\mathrm{I}\Gamma$ consists of at most two elements. Assume that $\mathrm{I}\Gamma$ has one element, which we denote by w. From the construction of $\mathrm{I}\Gamma$ we have:

$$\pi_1(w) = f_1^{min}, \qquad \pi_2(w) = f_2^{min}.$$

Consequently, w is dominated by every vector of V, so it is the only minimal vector in V. Assume now that $\mathrm{I}\Gamma$ consists of two vectors: w_1 and w_2.

($\Rightarrow$) After renumbering, $\Gamma_1 = \{w_1\}$ and $\Gamma_2 = \{w_2\}$. Hence, we obtain from equations (4.8)-(4.10)

$$f_1^{min} = \pi_1(w_1), \qquad f_1^{max} = \pi_1(w_2),$$
$$f_2^{min} = \pi_2(w_2), \qquad f_2^{max} = \pi_2(w_1).$$

Due to (4.6) the set of vectors $v \in V$ incomparable with $I\Gamma$ satisfies (4.12).
($\Leftarrow$) Let $v \in V$ for which inclusion (4.12) holds, then using renumbering of set Γ_j, $j = 1, 2$, from the above implication, we obtain:

$$\pi_1(v) > f_1^{min} = \pi_1(w_1), \qquad \pi_1(v) < f_1^{max} = \pi_1(w_2),$$
$$\pi_2(v) < f_2^{max} = \pi_2(w_1), \qquad \pi_2(v) > f_2^{min} = \pi_2(w_2).$$

According to Definition 3 and Formula (4.6), $v \sim w_1$ and $v \sim w_2$. Since $I\Gamma = \{w_1, w_2\}$, then v is incomparable with the vectors w_1 and w_2. □

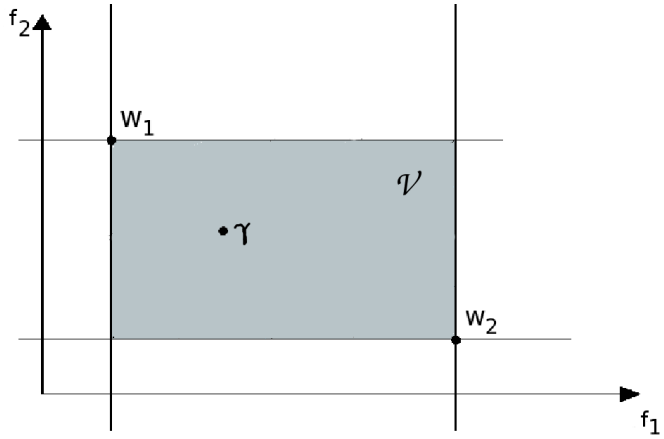
As shown in Figure 4.1a and Figure 4.1b, when $I\Gamma$ consists of two elements w_1 and w_2, a set of vectors incomparable with $I\Gamma$ is given by the rectangle $\mathcal{V}$. Let γ be a vector incomparable with $I\Gamma$, i.e. $\gamma \in \mathcal{V}$. The introduction of v_0 divides the rectangle $\mathcal{V}$ into three areas:

- A' and A'' is a set of vectors incomparable with $I\Gamma \cup \{\gamma\}$,
- B is a set of vectors smaller then γ,
- C is a set of vectors bigger then γ.

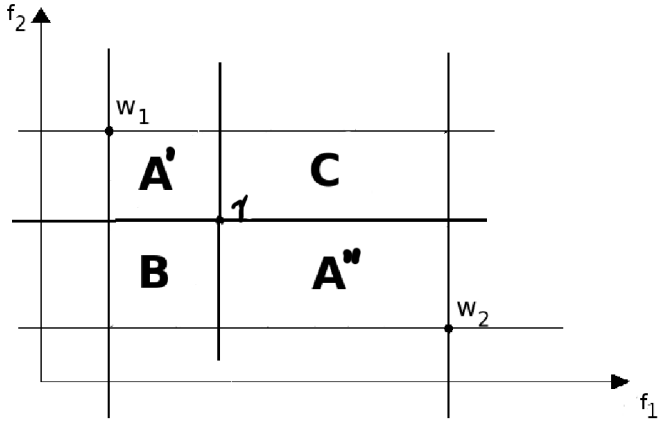
The above properties of $I\Gamma$ and vectors incomparable with $I\Gamma$ allow us to limit the search space $\mathcal{V}$ to find Pareto solutions.

4.4.1.3 Finding a Pareto Set in 2D Vector Space

This section proposes an algorithm for finding a Pareto set in two-dimensional space (see Algorithm 2). FIND-PARETO-SET(V) is a recursive algorithm that finds all Pareto points in the rectangle $\mathcal{V}$ defined by two points in the initial Pareto set $I\Gamma$ (see Lemma 1); this rectangle contains all points from V. The algorithm starts from



(a) A space $\mathcal{V}$ of incomparable vectors bounded by coordinates vectors $w_1, w_2 \in \Pi\Gamma$



(b) A partial space $\mathcal{V}$ when a new vector γ is introduced.

Figure 4.1: Search space for Pareto solutions

finding a point γ that does not dominate any other points in V (line 4). This point splits the area $\mathcal{V}$ into four rectangles (see Figure 4.1b). According to Lemma 2 and 3, $\mathbf{B} \cap \mathbf{V} = \emptyset$, $\mathbf{C}$ does not contain Pareto points, whereas points in rectangles $\mathbf{A}'$ and $\mathbf{A}''$ are incomparable with γ. The above procedure is recursively repeated for $\mathbf{V} \cap \mathbf{A}'$ and $\mathbf{V} \cap \mathbf{A}''$ (Q_1 and Q_2).

The algorithm sketched above calls FIND-PARETO-POINT($\bar{V}$) (see Algorithm 3) to find a Pareto point in the set $\bar{V}$. This procedure works in the pessimistic time $O(n^2)$, where n is a number of elements in $\bar{V}$ (when all solutions are comparable, i.e., to form a chain it may take n iterations to find a Pareto point). However, the expected running time is much shorter thanks to the random selection of points.

Algorithm 2 FIND-PARETO-SET(V)

if $V = \emptyset$ **then**
  return $\emptyset$
end if
$\gamma \leftarrow$ FIND-PARETO-POINT(V)
$Q_1 = \big(V \ \{\gamma\}\big) \cap \big((-\infty, f_1(\gamma)] \times [f_2(\gamma), \infty)\big)$
$Q_2 = \big(V \ \{\gamma\}\big) \cap \big([f_1(\gamma), \infty) \times (-\infty, f_2(\gamma)]\big)$
$\Gamma = \{\gamma\} \quad \cup$ FIND-PARETO-SET(Q_1) $\cup$ FIND-PARETO-SET(Q_2)
return Γ

Algorithm 3 FIND-PARETO-POINT($\bar{V}$)

if $\bar{V} = \emptyset$ **then**
  return $\emptyset$
end if
select $\hat{v}$ randomly from $\bar{V}$
while $\hat{v}$ dominates points from $\bar{V} \setminus \{\hat{v}\}$ **do**
  $\bar{V} \leftarrow \{v \in \bar{V} \setminus \{\hat{v}\} : v \preceq \hat{v}\}$
  select $\hat{v}$ randomly from $\bar{V}$
end while
return $\hat{v}$

4.4.2 Pareto Algorithms

Following the similarity hypothesis researchers build models for groups of chemicals that have a common molecular fragment or common properties. These models are more reliable and give better predictions for chemicals that lie in the model applicability domains. Further, high quality models developed for a small subset of chemical space can be combined in a global model that covers larger chemical space using various ensemble techniques.

The chemical space X is a set of chemical compounds represented by the combination of all possible existing chemical descriptors (see Definition 4.1), and for a given endpoint there is a collection of existing models $\mathcal{M}$. For each chemical compound $x \in X$, model predictions $Y' = \{y'_1, \ldots, y'_m\}$ for models from $\mathcal{M}$ are known. To identify a model for a given query chemical compound q, the set of chemicals from X and their model performances is converted into a set of pairs (d_i, e_{im}), where d_i represents the distance between q and the *i*-th chemical compound from the chemical space. The error $e_{im} = |y(x_i) - y'_m(x_i)|$ defines model performance for the m-th model from $\mathcal{M}$

Algorithm 4 MODEL-IDENTIFY(T, q)

1: $V \leftarrow$ INIT(T,q)
2: $\Gamma \leftarrow$ FIND-PARETO-SET(V)
3: **if** $|\Gamma| = 1$ **then**
4: **return** modelId of the sole element of Γ
5: **else**
6: **return** FIND-MODEL-ID(Γ)
7: **end if**

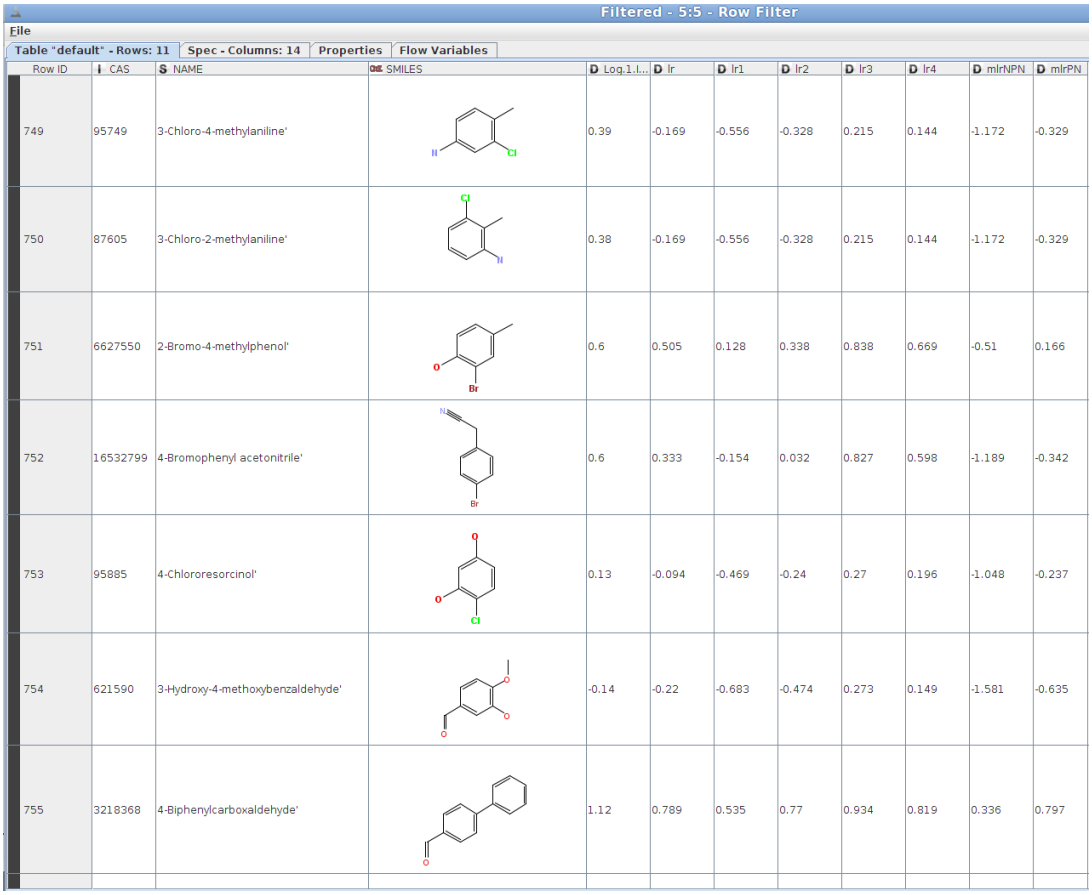
Algorithm 5 INIT(T, q)

1: $V \leftarrow \emptyset$
2: **for** $i = 0$ to rows(T) **do**
3: **for** $j = 0$ to models(T) **do**
4: calculate the distance d_{qi} and error e_{ij}
5: $V = V \cup \{(d_{qi}, e_{ij})\}$
6: **end for**
7: **end for**
8: **return** V

applied to the i-th chemical compound. In a set of such pairs, one can find models that have a low predictive power for the most similar chemical compounds whereas the other gives better predictions. This illustrates the situation often encountered in multi-criteria optimization problems: there is no solution that outperforms the others with respect to all criteria. Hence, instead of having one solution we have a set of solutions that cannot be compared to each other. The above task is a Pareto problem: one has to balance similarity to existing chemical compounds and accuracy of predictions offered by available models.

The model identification procedure (see Algorithm 4) can be described as follows: for a query chemical compound q and a given chemical space – 1) create the set V of pairs (d_i, e_{im}), 2) find the Pareto set for V, and 3) select the most suitable model for q. To create a set V we start from the array T (see Figure 4.2) that contains a structural representation of the chemical compound, its measured activity (for a given endpoint) and predictive performance of each model from $\mathcal{M}$.

After executing MODEL-IDENTIFY(T, q), in line 1, the array T is converted into a list of vectors V using procedure INIT(T, q) (see Algorithm 5). Every vector $v_i \in V$ is defined as a pair of the distance between q and the i-th chemical compound from T, and



Filtered - 5:5 - Row Filter

File

Table "default" - Rows: 11 | Spec - Columns: 14 | Properties | Flow Variables

Row ID	CAS	NAME	SMILES	Log.1.l...	lr	lr1	lr2	lr3	lr4	mlrNPN	mlrPN
749	95749	3-Chloro-4-methylaniline'		0.39	-0.169	-0.556	-0.328	0.215	0.144	-1.172	-0.329
750	87605	3-Chloro-2-methylaniline'		0.38	-0.169	-0.556	-0.328	0.215	0.144	-1.172	-0.329
751	6627550	2-Bromo-4-methylphenol'		0.6	0.505	0.128	0.338	0.838	0.669	-0.51	0.166
752	16532799	4-Bromophenyl acetonitrile'		0.6	0.333	-0.154	0.032	0.827	0.598	-1.189	-0.342
753	95885	4-Chlororesorcinol'		0.13	-0.094	-0.469	-0.24	0.27	0.196	-1.048	-0.237
754	621590	3-Hydroxy-4-methoxybenzaldehyde'		-0.14	-0.22	-0.683	-0.474	0.273	0.149	-1.581	-0.635
755	3218368	4-Biphenylcarboxaldehyde'		1.12	0.789	0.535	0.77	0.934	0.819	0.336	0.797

Figure 4.2: Collection of models for the IGC50 prediction for *Tetrahymena pyriformis*.

the error of the j-th model from $\mathcal{M}$ for the compound i. The distance $d_{qi} = 1 - ST_{qi}$ is calculated using Tanimoto coefficient ST, which is the most frequently used similarity measure in chemoinformatics [121]. This coefficient works with fingerprints (binary representation of molecules) and is defined as a ratio between the number of bits set on the same position in two fingerprints and the sum of bits set on different positions. The model error e_{ij} is defined as a distance between the true activity for compound i and the value computed by model j. We treat V as a set of all possible solutions for model identification for a given query molecule q and known chemical sub-space.

In line 2 of MODEL-IDENTIFY(T, q), we call FIND-PARETO-SET(V) to find the set of all Pareto points Γ in V. Then, we analyse points in Γ in order to choose

the most predictive model for q. In the case when $|\Gamma| = 1$, there is only one candidate, so the choice is trivial. This case is comparable to the DMS algorithm proposed in Section 4.3 which selects the most predictive model for the most similar chemical compound of q. In the case when Γ consists of many Pareto points, the model identification becomes a difficult task: the Tanimoto similarity coefficient (as well as other fingerprint similarity measures) between chemical compounds may not be correlated enough with their activity partially contradicting the similarity hypothesis [51] (see the end of this section for a detailed example). To identify a model using Pareto points, first we define the n-Pareto Neighbourhood as follows:

Definition 5. *The n-Pareto Neighbourhood is a set with at most n Pareto points from Γ which are at distance less than τ from the element q where $\tau > 0$ and $n > 0$.*

This Pareto neighbourhood defines at most n Pareto points that represents the neighbouring chemicals to the query chemical compound q and their the most predictive models in the same time. The threshold τ is selected by experiment and depends on the chemical similarity within a given chemical space. As was mentioned above, similar chemical compounds might have very different measurements of activity. To demonstrate this, we analysed the TETRATOX dataset which contains growth inhibition concentration (IGC50) for *Tetrahymena pyriformis* [45, 99]. Chemical compounds were compared in pairs. Their Tanimoto similarity coefficient and differences in measured activity were collected. Summarised results are displayed in Table 4.1. Column headers hold differences in the measured activity between two chemicals, while row headers describe molecule similarity threshold. A single cell of this array represents the number of pairs of chemical compounds for which the distance is smaller than the row identifier and the difference in the activity is smaller than the column identifier. The TETRATOX dataset contains over one thousand chemical compounds and the biggest difference between measured values of IGC50 is equal to 5.3. Notice that the number of pairs of chemicals that are similar, based on both the fingerprint similarity and the activity, is very small. There is only one pair of chemical compounds that have the same activity and maximal similarity (1 row, 1 column). On the other hand, there are many chemicals which are similar fingerprint-wise but have different activities. This makes model identification challenging. There is not a single model identified that could be the most reliable for a query chemical compound. This in-

Table 4.1: Analysis of chemical compound similarities in order to highlight the difference of the chemical activity for the TETRATOX dataset

$f_{sim}/\ dif\, f_{activ}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0	1	2	2	2	2	2	2	2
0.1	3	13	27	44	51	62	70	79
0.2	6	112	220	335	431	512	585	655
0.3	16	318	617	933	1213	1474	1719	1928
0.4	32	720	1402	2081	2701	3297	3840	4328
0.5	66	1380	2726	4042	5227	6437	7536	8547
$f_{sim}/\ dif\, f_{activ}$	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
0	2	2	2	2	2	2	2	2
0.1	84	90	93	96	99	103	104	104
0.2	700	753	782	801	827	842	849	858
0.3	2106	2278	2412	2507	2621	2715	2784	2821
0.4	4763	5160	5526	5837	6119	6360	6575	6724
0.5	9481	10362	11167	11840	12488	13082	13589	14004

volves the analysis of the model performances over Pareto neighbourhood set that is discussed in detail in the following subsections.

4.4.2.1 Average Pareto Model Identification

Having defined the Pareto neighbourhood for a given chemical compound q, two methods for model identification are proposed below. The first one is called n-Average Pareto (see Algorithm 6). The threshold τ is used to remove these chemical compounds which are dissimilar to the query compound q. Next, we average model errors for the chemicals represented by Pareto points and then the model with the smallest average error is selected. Algorithm 4 that uses n-Average Pareto method is called n-Average

Algorithm 6 n-Average Pareto

FIND-MODEL-ID(Γ, T, n, τ)

1: n-PN $\leftarrow$ n-Pareto neighbourhood for a given n and the threshold τ
2: $X' \leftarrow$ all chemical compounds linked to points in n-PN (use T to accomplish this task)
3: compute for each model average error on chemical compounds from X'
4: **return** Id of the model with smallest average error

Algorithm 7 n-Centroid Pareto

FIND-MODEL-ID(Γ, T, n, τ)

1: n-PN $\leftarrow$ n-Pareto neighbourhood for a given n and the threshold τ
2: for all points from n-PN calculate the centroid c
3: for each point from n-PN calculate the Euclidean distance to the centroid
4: **return** Id of the model having the Pareto point with the smallest distance to the centroid.

Pareto Model Identification (n-APMI). The usage of Pareto neighbourhood in comparison with the standard nearest neighbourhood is more sensitive to model performance and allows for the rejection of similar chemical compounds on which models perform badly.

4.4.2.2 Centroid Pareto Model Identification

The second method is called n-Centroid Pareto (see Algorithm 7). For all Pareto points from the n-Pareto Neighbourhood the centroid Pareto point c is calculated according to the formula:

$$c = (d_c, e_c) = \left(\frac{\sum_{p \in n-PN} d_p}{|n - PN|}, \frac{\sum_{p \in n-PN} e_p}{|n - PN|}\right), \tag{4.13}$$

where d_c is the average of distances and e_c is the average of model errors for all Pareto points from the neighbourhood $n - PN$. In the next step, the Euclidean distance between Pareto points and the centroid is computed. The model that is associated with the Pareto point for which the Euclidean distance to the centroid is minimal, is selected. Algorithm 4 that uses n-Centroid Pareto is called n-Centroid Pareto Model Identification (n-CPMI). According to the definition, both n-APMI and n-CPMI are partitioning models that split chemical space into disjoint groups and allow unambiguous model identification.

4.5 Experimental Results

Three experiments were proposed in order to demonstrate the advantages of model identification for predictive toxicology. Each experiment has two phases. In the first phase we treat model identification as a classification problem to study the performance of proposed methods in comparison with the other classification algorithms. The class

represents the name of the most predictive model from the collection of existing models and it is assign to each chemical compound. In the second phase, for each chemical compound we apply the identified model to predict the following: the growth inhibition concentration (IGC50), partition coefficient (LogP), and chemical persistence in the soil. Finally, we compared these results with the original model performances.

4.5.1 IGC50 Prediction for *Tetrahymena pyriformis*

A dataset (*Tetrahymena pyriformis* Toxicity – TPT) of 1129 chemicals was obtained from the INCHEMICOTOX webpage [45]. This dataset is composed of toxicity data for the unicellular ciliate protozoan *Tetrahymena pyriformis* (see [99]) and was published in [125]. The measure of toxicity is the 50% growth inhibition concentration (IGC50). Two QSAR regression models were obtained from INCHEMICOTOX. These models are also reported in the JRC QSAR Models Database [89]. The first, non-polar narcosis (NPN) QSAR [26], was originally trained on 87 chemicals identified as non polar narcotics with $q^2 = 0.95$. The linear regression model was defined as follows:

$$\log(1/IGC50) = 0.83 \log P - 2.07,$$

where $\log P$ is the octanol-water partition coefficient. The second, polar narcosis (PN) QSAR model [27] for *Tetrahymena pyriformis*, was trained on 138 polar narcosis chemicals with $q^2 = 0.75$ and defined as follows:

$$\log(1/IGC50) = 0.62 \log P - 1.00.$$

Training datasets for both models were obtained from the JRC QSAR Models Database. We compared these sets with the *Tetrahymena pyriformis* dataset and we confirmed that 204 (136 from the PN model and 68 from the NPN models) training chemicals were present in the TPT dataset. We did not perform any data curation for this dataset. We implemented both models with the $\log P$ value calculated using the CDK library [103] and we used them to predict toxicity for the TPT datasets.

First, we considered the model identification problem as a classification problem to predict which model will be the most reliable for a given chemical compound. Having a dataset of the predicted IGC50 for both models and the measured value, we used

Table 4.2: Comparison of classification algorithms according to a number of correctly classified elements, false positive, false negative and the classifiers accuracies. The polar narcosis model label was defined as the positive class.

Method	Correct class	False Positive	False Negative	Accuracy
SMO	899	122 (10.8%)	106 (9.4%)	0.80
Part	904	123 (10.9%)	101 (8.9%)	0.80
NaiveBayes	845	191 (19%)	90 (7.9%)	0.75
J48	905	123 (10.9%)	100 (8.9%)	0.80
IBK(1)	905	121 (10.7%)	102 (9%)	0.80
IBK(3)	901	133 (11.7%)	94 (8.3%)	0.79
IBK(5)	889	149 (13.2%)	93(8.2%)	0.78
BayesNet	756	264 (23%)	108 (9.5%)	0.67
DMS	901	115 (10.1%)	112 (9.9%)	0.79
3-CPMI	902	136 (12%)	90 (7.9%)	0.79
5-CPMI	897	137 (12%)	94 (8.3%)	0.79
10-CPMI	863	168 (14.8%)	97 (8.5%)	0.76
3-APMI	**918**	**99 (8.7%)**	**111 (9.8%)**	**0.81**
5-APMI	891	115 (10%)	122 (10.8%)	0.78

a priori information (called "oracle model") about the best selected model for each chemical compound and we applied various classification methods. To simulate the model identification for new chemical compounds we used the leave-one-out (LOO) method. Table 4.2 includes results from the comparison of n-CPMI and n-APMI proposed in this chapter with the DMS algorithm, and with standard classification algorithms such as: NaiveBayes, BayesNet decision trees (PART and J48), nearest neighbour (IBK) or support vector machine (SMO) implemented in WEKA [36]. We used the default parameter settings for all classifiers. To generate these classification models we used binary descriptors (1024 - bit fingerprints calculated using CDK library) and the model errors. We compared all classifiers according to the number of the correctly classified chemicals and the classifier accuracies. The 3-APMI method gives the highest number of correctly classified elements and relatively low numbers for false positive and false negative - especially comparing this method to IBK(3). The 3-APMI uses the 3-Pareto neighbourhood whereas IBK(3) uses the 3-nearest neighbourhood for classification. This shows that model identification using Pareto points is as good as or can be better than other well known classification algorithms.

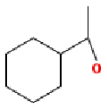
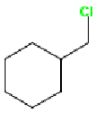
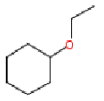
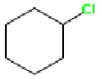
Name	CAS.N...	SMILES	Model....	oracle...
sec-Phenethyl alcohol	98-85-1		PN	NPN
Benzyl chloride	100-44-7		PN	NPN
Benzene	71-43-2		PN	NPN
Ethoxybenzene	103-73-1		PN	NPN
Chlorobenzene	108-90-7		PN	NPN

Figure 4.3: Chemical compounds wrongly associated with the PN model.

The decision which model is chosen, relies on the distance to the Pareto points. Figures 4.3 and 4.4 show misclassification examples for 3-APMI algorithm. In Figure 4.4 for *3*-phenyl-1-propanol the NPN model was identified. Its Pareto neighbourhood included three chemicals: *4-chloro-3-methylphenol*, *methylbenzene* and *4-dimethylbenzene* with the distances and models errors shown in Table 4.3. The 3-APMI averages model errors for all Pareto points and selects the one with the smallest error, in this case the NPN model. One can notice that the best model for this Pareto neighbourhood is the NPN model for *4-dimethylbenzene* whereas this chemical compound is not the most similar to the query chemical compound.

To demonstrate a correct classification example, we selected *benzylamine* that was associated correctly with the PN model. Its Pareto neighbourhood included two chemicals: *2-chloroaniline* and *(+/-)-1,2-diphenyl-2-propanol* with distances and model per-

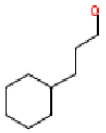
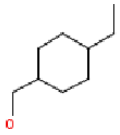
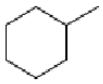
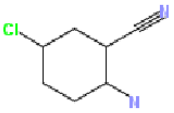
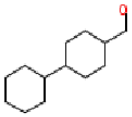
Name	CAS.N...	SMILES	Model....	oracle...
3-Phenyl-1-propanol	122-97-4		NPN	PN
4-Ethylbenzylalcohol	768-59-2		NPN	PN
Methylbenzene	108-88-3		NPN	PN
2-Amino-5-chlorobenzonitrile	5922-60-1		NPN	PN
4-Biphenylmethanol	3597-91-9		NPN	PN

Figure 4.4: Chemical compounds wrongly associated with the NPN model.

Table 4.3: Model performances and distance comparison of the 3-Pareto neighbourhood of the *3-phenyl-1-propanol*.

Name	distance	PN	NPN
methylbenzene	0.33	0.37	0.28
4-dimethylbenzene	0.36	0.54	0.08
4-chloro-3-methylphenol	0.30	0.61	1.14

Table 4.4: Model performances and distance comparison of the 3-Pareto neighbourhood of the *benzylamine*.

Name	distance	PN	NPN
2-chloroaniline	0.08	0.30	0.38
(+/-)-1,2-diphenyl-2-propanol	0.11	0.041	0.59

Table 4.5: Analysis of model prediction accuracies for IGC50 for *Tetrahymena pyriformis*

Method Name	r^2	RSE	q^2	MAE	RMSE
NPN	0.58	0.66	0.15	0.69	0.94
PN	0.58	0.66	0.58	0.50	0.66
DMS	0.68	0.56	0.62	0.43	0.62
3-CPMI	0.67	0.58	0.60	0.43	0.63
5-CPMI	0.66	0.59	0.59	0.44	0.65
10-CPMI	0.65	0.60	0.57	0.44	0.66
3-APMI	**0.69**	**0.56**	**0.65**	**0.41**	**0.60**
5-APMI	0.68	0.57	0.62	0.42	0.62
Oracle	0.75	0.50	0.71	0.35	0.54

formances shown in Table 4.4. These distances to the query chemical compound are small and for both chemicals the PN model gives the most reliable prediction. Again the 3-APMI identifies the PN model that has the minimal average error amongst the Pareto neighbours.

In the next step, from the entire TPT dataset, we selected chemicals included in the original training datasets for both models. We identified 4 out of 68 chemicals that were used to train the NPN model but the oracle model associated them with the PN model. The results from 3-APMI are shown in Figure 4.5. We repeated the same analysis for the training dataset of the polar narcosis model and we identified 9 out of 136 chemicals that were associated with the NPN model by the oracle model (see Figure 4.6).

To predict IGC50 for the TPT dataset we used the identified model for each chemical compound in this dataset. The results obtained for the entire dataset are shown in Table 4.5. The statistics used are: r^2 - correlation coefficient for the observed and predicted values, RSE - root-squared error, q^2 - predictive squared correlation coefficient,

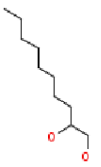
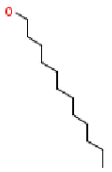
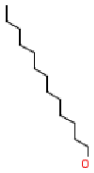
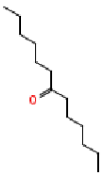
S "Name"	S "CAS.N..."	"SMILES"	D "Log.1..."	S "oracle"	S Model...
1,2-Decanediol	1119-86-4		0.764	PN	NPN
1-Dodecylalcohol	112-53-8		2.161	PN	NPN
Tridecylalcohol	112-70-9		2.45	PN	NPN
Di-n-hexyl ketone	462-18-0		1.521	PN	NPN

Figure 4.5: Chemical compounds wrongly associated with the PN model by the oracle model. These chemicals were originally used to train the NPN model.

MAE - mean absolute error and RMSE - root mean square error. The oracle model has the knowledge of the best model for each chemical compound. Its predictivity is low because we used only two existing models from JRC QSAR database that were designed based on mode-of-action (polar/non polar narcosis) for chemicals from TPT. The 3-APMI method provides the best prediction among non-oracle models. The first two rows present prediction statistics for PN and NPN models. They are lower than for all other models. Notice, however, that their r^2 and RSE statistics are identical. This is due to the fact that both models are affine functions of one and the same explanatory variable. An affine function can, therefore, transform one model into another. This is what happens when regression is applied to compute r^2 and RSE. Notice that other two measures of q^2 and predictive errors are different for these models.

As another example, we considered only a small subset of the whole initial TPT dataset that contains only 376 chemical compounds. This dataset includes all training chemicals used in PN and NPN models plus over 100 additional chemicals from the

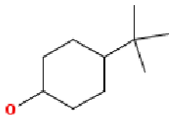
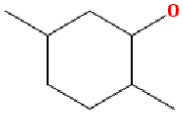
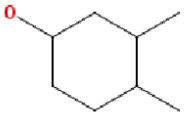
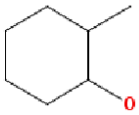
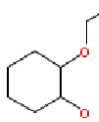
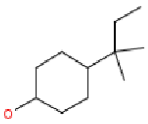
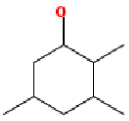
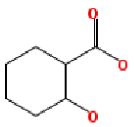
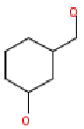
"Name"	"CA..."	"SMILES"	"Log.1..."	"oracle"	Model.Name
4-tert-Butylphenol	98-54-4		0.91	NPN	NPN
2,5-Dimethylphenol	95-87-4		0.08	NPN	PN
3,4-Dimethylphenol	95-65-8		0.12	NPN	PN
2-Methylphenol	95-48-7		-0.29	NPN	PN
2-Ethoxyphenol	94-71-3		-0.36	NPN	PN
4-tert-Pentylphenol	80-46-6		1.23	NPN	NPN
2,3,5-Trimethylphenol	697-82-5		0.36	NPN	NPN
Salicylic acid	69-72-7		-0.51	NPN	NPN
3-Hydroxybenzylalcohol	620-24-6		-1.04	NPN	NPN

Figure 4.6: Chemical compounds wrongly associated with the NPN model by the oracle model. These chemicals were used to train the PN model.

Table 4.6: Comparison of classification algorithms according to a number of correctly classified elements, false positive, false negative and the classifiers accuracies. The polar narcosis model label was defined as the positive class.

Method	Correct class	False Positive	False Negative	Accuracy
SMO	296	47(12%)	33(8.7%)	0.787
Part	303	34(9%)	39(10.3%)	0.805
NaiveBayes	281	67(17%)	28(7.4%)	0.747
J48	296	44(11.7%)	36(9.5%)	0.787
IBK(1)	307	42(11.1%)	27(7.1%)	0.816
IBK(3)	300	42(11.1%)	34(9%)	0.797
IBK(5)	299	46(12.2%)	31(8.2%)	0.795
BayesNet	273	76(20.1%)	27(7.1%)	0.726
DMS	297	48(12.7%)	31(8.2%)	0.719
3-CPMI	**316**	**29 (7.7%)**	**31(8.2%)**	**0.844**
5-CPMI	305	33(8.7%)	38(10.1%)	0.811
10-CPMI	288	41(10.9%)	47(12.5 %)	0.766
3-APMI	306	33(8.7%)	37(9.8%)	0.813
5-APMI	300	41(10.9%)	35(9.3%)	0.797

TPT dataset. We included chemicals for which the absolute error of the oracle model is less than 0.4 and they are in the applicability domain of both models. The value of $\log P \in [-0.5, 6.2]$ and the toxicity value is in the range $[-2.5, 3.05]$. Again we compared various classifiers that were used for model identification (see Table 4.6). In this case the best method is 3-CPMI that from the 3-Pareto neighbourhood selects model for which the Pareto point is the closest to the neighbourhood centroid. This method gives better results if compared with the DMS method that selects the model with the smallest error for the nearest neighbour. Comparing the regression models for IGC50 (see Table 4.7), 3-CMPI method give us better prediction than DMS, PN and NPN models.

The above examples show a great potential of the proposed model identification methods. Model identification can be considered as an ensemble technique to build consensus models in predictive toxicology.

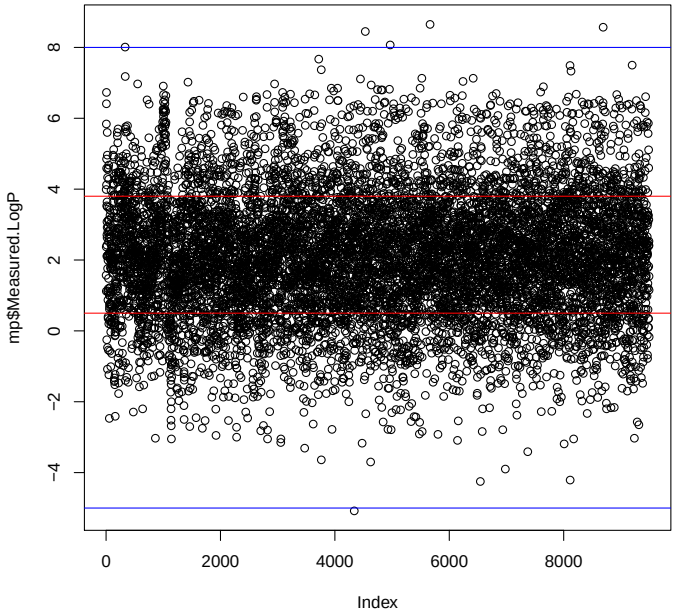
Table 4.7: Analysis of model prediction accuracies for IGC50 for Tetrahymena pyriformis

Method Name	r^2	RSE	q^2	MAE	RMSE
NPN	0.84	0.37	0.60	0.44	0.57
PN	0.84	0.37	0.75	0.33	0.46
DMS	0.89	0.30	0.88	0.20	0.32
3-CPMI	**0.92**	**0.25**	**0.91**	**0.16**	**0.26**
5-CPMI	0.90	0.28	0.89	0.18	0.29
10-CPMI	0.88	0.32	0.86	0.21	0.33
3-APMI	0.91	0.27	0.90	0.18	0.29
5-APMI	0.90	0.28	0.89	0.19	0.30
Oracle	0.98	0.10	0.98	0.09	0.11

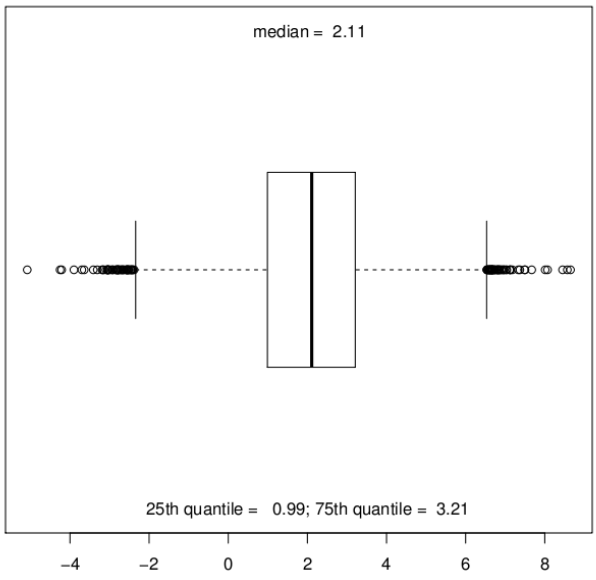
4.5.2 LogP Prediction for Syngenta Dataset

The octanol/water Partition coefficient (LogP) is a measure of the lipophilicity of chemical compounds and is an important descriptive parameter in bio-studies [32]. It describes the ability of a chemical compound to dissolve in fats, oils, lipids, and non-polar solvents. Currently, there are various methods that allow the estimation of LogP: fragment-based methods (CLOGP, KOWWIN), atom contribution methods (TSAR, XLOGP), topological indices (MLOGP), molecular properties (BLOGP). The initial dataset contained about 9000 chemical compounds and their measured LogP values. This measure was collected from experiments that have been run in Syngenta's laboratories. The measured value of LogP is in the range $[-5.08, 8.65]$ (see Figure 4.7a). There was around 6300 chemicals between the first and third quantile (see red lines in Figure 4.7a and Figure 4.7b). There was no additional data curation apart from the curation provided by Syngenta.

For such defined dataset, two experiment were prepared. In the first experiment, three models to predict LogP were considered. The first one called CLOGP was developed in Syngenta, the second model called KOWWIN from EPI Suite [28] and finally MLOGP model from Dragon software [109]. These models were run on entire dataset and the model predictions were collected for each chemical compound in this dataset.



(a) measured LogP



(b) summary

Figure 4.7: Syngenta's measured LogP dataset.

Table 4.8: Analysis of model prediction accuracies for a LogP estimation

nr chemicals	Mod.Name	q^2	MAE	RMSE
1000	CLOGP	0.83	0.38	0.74
	MLOGP	0.57	0.84	1.19
	KOWWIN	0.79	0.47	0.83
	3-APMI	0.84	0.38	0.74
2000	CLOGP	0.76	0.41	0.78
	MLOGP	0.44	0.85	1.2
	KOWWIN	0.69	0.50	0.88
	3-APMI	0.78	0.39	0.72
2333	CLOGP	0.37	1.21	1.54
	MLOGP	0.39	1.13	1.52
	KOWWIN	0.41	1.01	1.49
	3-APMI	0.64	0.80	1.16

The 1000 randomly selected chemicals out of 9000 was used as an unknown dataset and the remaining 8000 chemicals as a known chemical space for the partitioning model. The 3-APMI method was used as the most accurate in the previous experiment. The performance of CLOGP, KOWWIN, MLOGP, and 3-APMI models were compared for these 1000 randomly selected chemicals. The same experiment was repeated for 2000 randomly selected chemicals. In this case, the remaining set of 7000 chemicals was used as a known chemical space. Finally, from the initial dataset those chemical compounds for which the oracle model has absolute error > 0.7 were selected and used to test the partitioning model. We obtained a set of 2333 chemical compounds and we used remaining part as a known chemical space. In this experiment, 3-APMI algorithm was run only once for each case: 1000, 2000, 2333 chemicals. Table 4.8 displays the accuracy of model predictions. As LogP is a continuous value, the validation statistics for the linear regression model were used to compare model performances (q^2, MAE, RMSE). One can observe that there is one best model – CLOGP. The other models: KOWWIN, MLOGP even if they are widely accepted models and many researches use them to calculate LogP, are not very accurate. These results show that the 3-APMI method is generally at least as good as the best model (CLOGP). In the case of 1000 randomly selected chemicals CLOGP was hard to beat, although for 2000

Table 4.9: Analysis of model prediction accuracies for a LogP estimation for 1000 randomly selected chemicals in 10-CV

Mod.Name	Run	R2	RSE	Q2	MAE	RMSE	Run	R2	RSE	Q2	MAE	RMSE
CLOGP		0.847	0.694	0.826	0.412	0.739		0.805	0.767	0.774	0.449	0.825
XLogP		0.621	1.094	0.458	0.927	1.306		0.546	1.170	0.279	1.022	1.473
MLOGP	1	0.563	1.174	0.546	0.843	1.195	2	0.561	1.151	0.559	0.857	1.152
KOWIN_LOGP		0.831	0.729	0.804	0.469	0.786		0.776	0.821	0.728	0.524	0.904
3-APMI		0.867	0.647	0.859	0.359	0.665		0.864	0.641	0.856	0.380	0.659
Oracle		0.940	0.435	0.939	0.207	0.438		0.923	0.481	0.921	0.228	0.486
CLOGP		0.775	0.833	0.743	0.462	0.890		0.805	0.723	0.768	0.421	0.787
XLogP		0.583	1.135	0.330	0.984	1.437		0.572	1.071	0.276	0.966	1.392
MLOGP	3	0.557	1.170	0.552	0.867	1.175	4	0.563	1.083	0.554	0.820	1.092
KOWIN_LOGP		0.759	0.864	0.718	0.517	0.933		0.798	0.737	0.755	0.467	0.809
3-APMI		0.839	0.705	0.828	0.383	0.728		0.859	0.615	0.849	0.341	0.636
Oracle		0.911	0.525	0.909	0.233	0.531		0.921	0.460	0.918	0.207	0.468
CLOGP		0.788	0.779	0.747	0.439	0.850		0.805	0.752	0.776	0.434	0.806
XLogP		0.549	1.136	0.257	1.008	1.456		0.566	1.122	0.300	1.001	1.423
MLOGP	5	0.534	1.154	0.531	0.858	1.157	6	0.561	1.128	0.556	0.835	1.134
KOWIN_LOGP		0.767	0.816	0.722	0.501	0.891		0.788	0.785	0.753	0.493	0.846
3-APMI		0.847	0.662	0.839	0.376	0.679		0.864	0.629	0.856	0.355	0.647
Oracle		0.918	0.485	0.916	0.229	0.491		0.931	0.448	0.929	0.213	0.453
CLOGP		0.843	0.662	0.822	0.397	0.705		0.818	0.739	0.796	0.419	0.782
XLogP		0.604	1.052	0.419	0.911	1.273		0.586	1.115	0.379	0.954	1.364
MLOGP	7	0.565	1.103	0.557	0.837	1.111	8	0.544	1.171	0.535	0.859	1.181
KOWIN_LOGP		0.779	0.786	0.736	0.484	0.859		0.791	0.792	0.762	0.482	0.844
3-APMI		0.876	0.589	0.869	0.349	0.603		0.883	0.591	0.878	0.341	0.604
Oracle		0.945	0.392	0.944	0.194	0.394		0.934	0.444	0.933	0.210	0.449
CLOGP		0.823	0.750	0.791	0.446	0.814		0.823	0.750	0.800	0.423	0.796
XLogP		0.602	1.123	0.338	0.965	1.448		0.607	1.116	0.385	0.985	1.394
MLOGP	9	0.587	1.145	0.583	0.843	1.149	10	0.559	1.182	0.553	0.861	1.188
KOWIN_LOGP		0.781	0.834	0.739	0.522	0.909		0.795	0.806	0.762	0.497	0.867
3-APMI		0.871	0.640	0.862	0.374	0.660		0.870	0.641	0.865	0.351	0.653
Oracle		0.938	0.442	0.936	0.220	0.450		0.939	0.439	0.938	0.206	0.444

randomly selected chemicals one can clearly see the benefit of using 3-APMI (higher q^2 and lower MAE). The biggest gain is, however, observed for those chemicals whose activity is difficult to predict (the last case). In case where all available models have a poor predictivity, the proper model identification and the usage of the identified model for predictions can lead to increased model accuracy.

To test 3-APMI method and to show its robustness, the second experiment was established. For the same LogP dataset, four models were considered: CLOGP - developed in Syngenta, KOWWIN - from EPI Suite [28], MLOGP - from Dragon software [109] and XLOGP from CDK library [92]. The same 3-APMI algorithm was used for a model identification in a cross-validation test. In the first case 1000 chemicals were

randomly selected in each of 10 runs. Table 4.9 presents the model performances for each run. The R2 and RSE were obtained from the fitted linear regression representing a relation between observed and predicted values. The Q2, MAE and RMSE were calculated using (2.6), (2.15), (2.14). One can notice that in this case CLOGP model is the best model among the available models and using the partitioning models to choose the best possible model for a new data we are able to increase the model predictivity (see Figure 4.8). This figure demonstrates that using the 3-APMI methods increased the prediction in comparison to the best model (CLOGP).

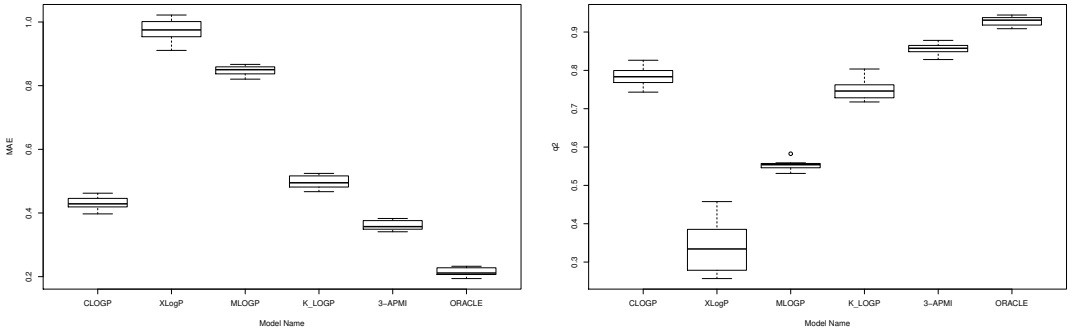


Figure 4.8: Aggregated minimum absolute error (MAE) and predictive squared coefficient correlation (Q2) for the 3-APMI 10-cross validation. In each test 1000 chemical were selected.

In second test 2000 chemicals were randomly selected out of 9000 available chemical compounds in each of 10 runs. Model performances were collected and presented in Table 4.10 in the same way as in the previous case. The results demonstrate that the usage a proper model identification method for unknown data can lead to the increased model predictivity. In each of 10 runs 3-APMI gives better predictions than the CLOGP model (see Figure 4.9). The analysis of oracle model's performance is an important element in this study. The oracle model represents situation where we exactly know which model should be used for new data. The goal of model identification methods is to minimize a distance between a chosen method and the oracle model performances. One can observe that, the chosen method (3-APMI) in this experiment gives predictions that are between the best model and oracle model. The results were consistence between different runs, demonstrating the robustness of the model identification method based on the chemical space partitioning.

Table 4.10: Analysis of model prediction accuracies for a LogP estimation for 2000 randomly selected chemicals in 10-CV

Mod.Name	Run	R2	RSE	Q2	MAE	RMSE	Run	R2	RSE	Q2	MAE	RMSE
CLOGP	1	0.819	0.742	0.790	0.426	0.798	2	0.771	0.825	0.735	0.462	0.887
XLogP		0.595	1.109	0.387	0.954	1.364		0.533	1.178	0.230	1.023	1.512
MLOGP		0.576	1.134	0.567	0.840	1.147		0.537	1.173	0.532	0.865	1.179
KOWIN_LOGP		0.788	0.803	0.743	0.501	0.882		0.728	0.899	0.675	0.534	0.982
3-APMI		0.861	0.649	0.853	0.361	0.668		0.818	0.736	0.805	0.390	0.761
Oracle		0.938	0.435	0.937	0.209	0.439		0.898	0.551	0.894	0.243	0.560
CLOGP	3	0.807	0.759	0.783	0.438	0.804	4	0.835	0.711	0.814	0.401	0.756
XLogP		0.553	1.155	0.293	1.013	1.452		0.611	1.092	0.435	0.948	1.315
MLOGP		0.561	1.145	0.558	0.846	1.147		0.591	1.120	0.588	0.822	1.123
KOWIN_LOGP		0.770	0.828	0.736	0.501	0.887		0.826	0.730	0.803	0.458	0.777
3-APMI		0.861	0.644	0.857	0.363	0.653		0.871	0.629	0.867	0.354	0.639
Oracle		0.927	0.467	0.926	0.217	0.471		0.936	0.443	0.935	0.203	0.446
CLOGP	5	0.821	0.727	0.795	0.425	0.779	6	0.824	0.737	0.799	0.431	0.787
LogP		0.588	1.104	0.370	0.948	1.365		0.602	1.107	0.390	0.968	1.371
MLOGP		0.538	1.169	0.530	0.853	1.179		0.566	1.157	0.558	0.846	1.167
KOWIN_LOGP		0.781	0.805	0.741	0.489	0.874		0.806	0.773	0.776	0.488	0.831
3-APMI		0.855	0.655	0.847	0.363	0.673		0.874	0.624	0.868	0.361	0.637
Oracle		0.932	0.450	0.929	0.212	0.458		0.934	0.452	0.932	0.212	0.458
CLOGP	7	0.812	0.753	0.789	0.430	0.797	8	0.795	0.787	0.765	0.446	0.842
XLogP		0.581	1.123	0.375	0.941	1.371		0.534	1.186	0.289	1.001	1.465
MLOGP		0.549	1.165	0.542	0.849	1.173		0.558	1.156	0.556	0.849	1.15
KOWIN_LOGP		0.780	0.814	0.750	0.493	0.867		0.767	0.839	0.729	0.503	0.904
3-APMI		0.853	0.664	0.848	0.371	0.677		0.831	0.714	0.820	0.392	0.736
Oracle		0.926	0.470	0.925	0.219	0.474		0.912	0.515	0.910	0.231	0.522
CLOGP	9	0.829	0.707	0.806	0.419	0.754	10	0.813	0.723	0.782	0.422	0.780
XLogP		0.585	1.101	0.366	0.945	1.361		0.581	1.082	0.354	0.930	1.344
MLOGP		0.536	1.166	0.527	0.865	1.175		0.545	1.128	0.538	0.834	1.136
KOWIN_LOGP		0.791	0.781	0.754	0.491	0.847		0.802	0.744	0.765	0.474	0.810
3-APMI		0.862	0.636	0.854	0.367	0.653		0.867	0.610	0.859	0.349	0.628
Oracle		0.928	0.460	0.926	0.219	0.465		0.931	0.441	0.929	0.205	0.446

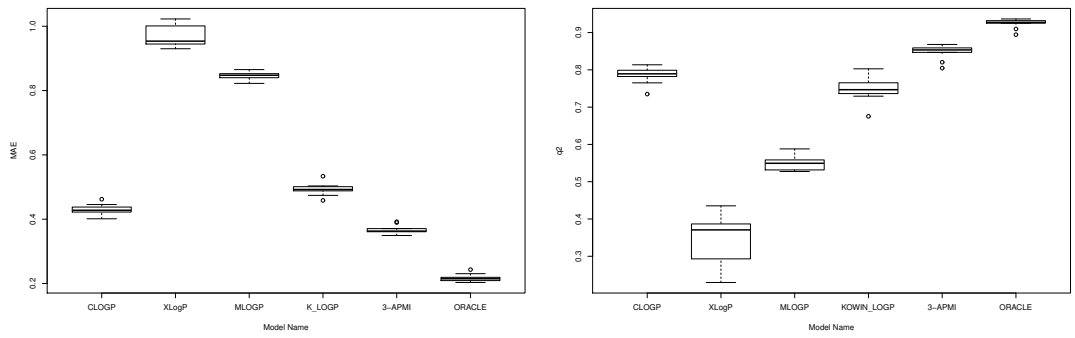


Figure 4.9: Aggregated minimum absolute error (MAE) and predictive squared coefficient correlation (Q2) for the 3-APMI 10-cross validation. In each test 2000 chemical were selected.

4.5.3 Prediction of Chemical Persistence in Soil for Syngenta Dataset

The length of time a chemical compound remains active in soil is called ”soil persistence,” or ”soil residual life”. The longer a chemical persists, the higher the potential for human or environmental exposure to it. The rate of degradation of a chemical in soil is a critical aspect of its environmental profile. It is desirable to control e.g weeds during the application time but after this time chemicals should not persist and affect growing crops. There is a lot of experimental data that includes soil-based screens for the chemical persistence, but there would still be a benefit in being able to predict soil persistence. The prediction of the degradation rate in soil is hard due to a complex combination of a number of different physical, chemical and biological processes. This is why finding a relation between various chemical attributes that can affect persistence in soil is a difficult problem.

In 2012 Syngenta published the challenge (prediction of soil persistence) on the IdeaConection [42] platform to identify potential new approaches. There were 268 chemicals from Syngenta's various projects, in the overall dataset. The dataset provided included log base 10 of the degradation rate for whole soil where $\log(k) \in [-3.8, 0.84]$ and soil water $\log(k) \in [-2.7, 4.7]$ and $k = 0.693/\text{half-life}$. For each chemical compound, there were 4000 descriptors such as: constitutional indices, ring descriptors, connectivity and information indices, 2D matrix-based descriptors, burden eigenvalues, geometrical descriptors, molecular properties, drug-like indices, functional group counts, gateway descriptors, WHIM descriptors, 3D-MoRSE descriptors, RDF descriptors, 3D matrix-based descriptors, topological descriptors calculated by Dragon software [109]. From the initial dataset a training dataset of 134 was published. The remaining part was hidden for competitors and used for the external validation of new QSAR models.

The challenge published by Syngenta was to build a model that accurately predicts the degradation rate in soil for each chemical. There were five teams in the competition, providing more than one model. Only four of them submitted models that could be validated on the external dataset. The statistics R^2, q^2 and R^2_{all} (see Formula 2.3-2.6) were used to identify the winning models and award teams. Table 4.11 and Table 4.12 presents the result of the validation process. The wining models were Team 4's model based on Burden Eigenvalues for soil-water and Team 6's model for whole-soil.

Table 4.11: Validation of soil-water models.

Model Name	R^2_{tr}	q^2	R^2_{all}
Team: 4,G I Model Based on Burden Eigenvalues	0.77	-0.03	0.38
Team: 4, G I Model Based on All Descriptors	0.82	-0.22	0.32
Team: 3, Model 2	0.67	-0.37	0.17
Team: 6, Model	0.79	-0.65	0.09
Team: 3, Model 1	0.86	-2.44	-0.74
Team: 4, G I Model Based on RDF Descriptors	0.74	-0.45	0.16
Team: 4, G III Model	0.59	-0.26	0.18
Team: 4, G I Model Based on 3D Matrix Descriptors	0.57	-0.22	0.18
Team: 4, G I Model Based on GETAWAY Descriptors	0.56	0.26	0.42
Team: 4, G II Model Based on Ring Descriptors, Drug-like Indices and Molecular Properties	0.31	-0.20	0.06
Team: 4, G II Model Based on All Descriptors	0.30	-1.22	-0.43
Team: 4, G II Model Based on Constitutional Indices, Geometrical and GETAWAY Descriptors	0.04	0.22	0.13

Table 4.12: Validation of whole-soil models.

Model Name	R^2_{tr}	q^2	R^2_{all}
Team: 6, Model	0.71	-0.57	0.10
Team: 3, Model 2	0.67	-0.79	-0.03
Team: 3, Model 1	0.68	-0.86	-0.06
Team: 4, G I Model Based on All Descriptors	0.79	-1.10	-0.12
Team: 4, G I Model Based on 3D Matrix Descriptors	0.72	-1.27	-0.23
Team:5, Linear Regression Approximation to Neural Network	-1.86	-2.09	-1.96
Team: 4, G III Model	0.72	-0.28	0.24
Team: 4, G I Model Based on RDF Descriptors	0.70	-1.26	-0.24
Team: 4, G I Model Based on GETAWAY Descriptors	0.63	-0.29	0.19
Team: 4, G I Model Based on Burden Eigenvalues	0.59	-0.12	0.25
Team: 4, G II Model Based on Constitutional Indices, Geometrical and GETAWAY Descriptors	0.43	-1.36	-0.42
Team: 4, G II Model Based on All Descriptors	0.13	-2.84	-1.29
Team: 4, G II Model Based on Ring Descriptors, Drug-like Indices and Molecular Properties	-0.15	-0.76	-0.44

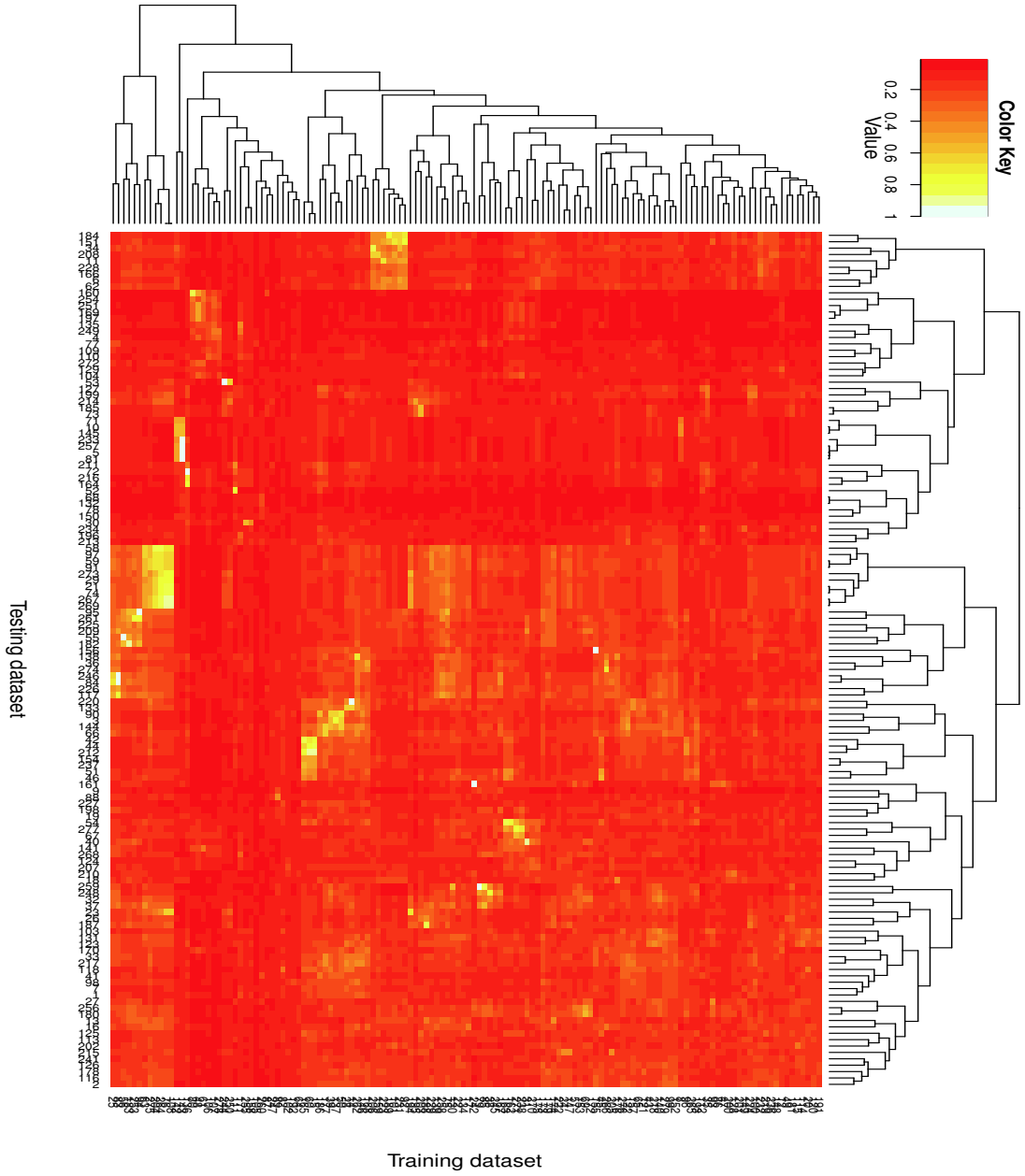


Figure 4.10: Heatmap of the chemical structure similarity between training and testing datasets.

One can notice that models have an acceptable range of the coefficient of determination R^2_{tr} for a training dataset, whereas the q^2 values went to be very low. The negative number means that using the average as a predictor will give better prediction comparing to the usage of original models. This happened because the chemicals used in the initial dataset were very diverse. Only a portion of test chemicals has similar chemicals in training dataset. Figure 4.10 presents the heat-map of the chemical compound structural similarities. The columns represent chemicals from training dataset and rows represent chemicals from testing dataset. The dendograms are a replacement for similarity functions in hierarchical clustering for each dataset separately. The chemical similarity was calculated based on the "extended" fingerprint using Tanimoto measure (see Formula 2.1) and ranged from 0 to 1. Similarity 1 means that chemical structures are identical or very similar.

The validation results were very surprising, most of the models can give reliable prediction for chemicals used in training process but only a few give positive statistics for the test chemicals. The interesting question here was if model identification can be used in such a case and if the obtained results can be better than results of the original provided models. To answer these questions two tests were performed for the soil-water endpoint. For each test a few models were selected and DMS, APMI, CMPI model identification methods were applied. The prediction accuracy was compared with the original selected models, oracle model and the AVG models. The AVG model uses the nearest neighbourhood and selects the model that gives the minimal average error for chemicals in this neighbourhood.

In the first test, three models that have good accuracies for the training dataset were selected: Team: 4, G I Model Based on All (MI) Descriptors, Team: 4,G I Model Based on Burden Eigenvalues (MII) and Team: 4, G I Model Based on RDF Descriptors (MIII). These models were used for model identification and compared with the oracle model. Table 4.13 presents the results of model accuracies using the LOO approach for the training dataset. Table 4.14 presents the results of model accuracies for the testing dataset and the training dataset is used as a known chemical space. The columns represent the n-neighbourhood. One can notice that usage of oracle model will lead to the accuracy 0.95 for the training dataset and 0.64 for testing dataset. In this case, the partitioning model selected based on the training dataset validation is as good as the best provided model MI (see 4-APMI) and the results are comparable with using

Table 4.13: Model identification applied to three models for training dataset of soil-water endpoint.

Mod Name	2	3	4	5	6	7	8	9	10
MI	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
MII	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
MII	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74
Oracle	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
DMS	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
APMI	0.80	0.80	**0.84**	0.84	0.84	0.84	0.84	0.834	0.84
CPMI	0.80	0.79	0.79	0.79	0.78	0.80	0.82	0.82	0.81
AVG	0.81	0.80	0.83	0.82	0.82	**0.84**	0.84	0.83	0.82

Table 4.14: Model identification applied to three models for testing dataset of soil-water endpoint.

Mod Name	2	3	4	5	6	7	8	9	10
MI	-0.22	-0.22	-0.22	-0.22	-0.22	-0.22	-0.22	-0.22	-0.22
MII	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
MIII	-0.45	-0.45	-0.45	-0.45	-0.45	-0.45	-0.45	-0.45	-0.45
Oracle	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
DMS	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27
APMI	-0.23	-0.24	-0.33	-0.27	-0.28	-0.28	-0.26	-0.23	-0.26
CPMI	-0.27	-0.33	-0.35	-0.27	-0.24	-0.25	-0.23	-0.25	-0.22
AVG	-0.22	-0.24	-0.27	-0.35	-0.36	-0.42	-0.34	-0.28	-0.43

the standard nearest neighbourhood approach (see 7-AVG in Table 4.13). The results for the testing dataset are influenced by the big residuals in all three models. Model MII have the smallest negative q^2 statistic and the model MIII the biggest. The most frequent selected model was model MI and the accuracies for the partitioning models for both the nearest and Pareto neighbourhoods are close to -0.22.

In the second test, only two models with positive q^2 were selected: Team: 4, G II Model Based on Constitutional Indices, Geometrical and GETAWAY Descriptors (MI) and Team: 4, G I Model Based on GETAWAY Descriptors (MII). Table 4.15 presents the comparison of the model accuracies. one can notice that there is one most predictive model, MII with accuracy 0.56. The accuracies of partitioning model that uses Pareto approaches as well as the DMS algorithm are in the same range as model MII. The 3-APMI method uses model MII with comparable accuracy 0.6.

Table 4.15: Model identification applied to two models for training dataset of soil-water endpoint.

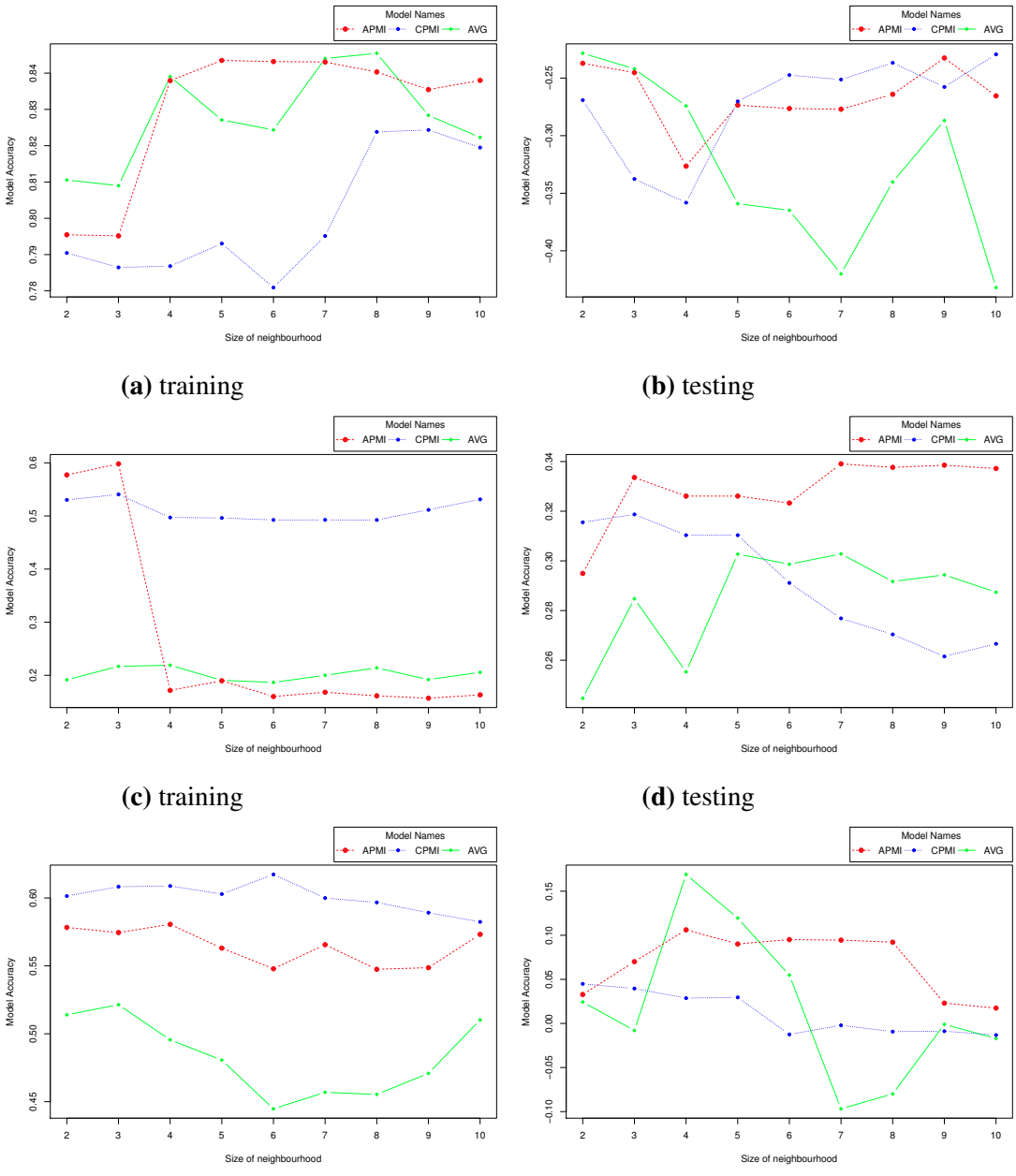
Mod Name	2	3	4	5	6	7	8	9	10
MI	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
MII	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
Oracle	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
DMS	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
APMI	0.58	**0.6**	0.17	0.19	0.16	0.17	0.16	0.16	0.16
CPMI	0.53	0.54	0.49	0.49	0.49	0.49	0.49	0.51	0.53
AVG	0.19	0.21	0.21	0.19	0.18	0.2	0.21	0.19	0.20

Table 4.16: Model identification applied to two models for testing dataset of soil-water endpoint.

Mod Name	2	3	4	5	6	7	8	9	10
MI	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
MII	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
Oracle	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
DMS	**0.34**	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
APMI	0.29	0.33	0.32	0.33	0.32	**0.34**	0.34	0.34	0.34
CPMI	0.32	0.32	0.31	0.31	0.29	0.28	0.27	0.26	0.27
AVG	0.24	0.28	0.25	0.30	0.29	0.30	0.29	0.29	0.28

Interesting observation was made when model identification algorithms were applied to the testing dataset (see Table 4.16). Both original models have positive q^2. Having knowledge which model should be used, the accuracy could be increased to 0.56 (see oracle model). In this case, application of the nearest neighbour (DMS) and Pareto approaches win compared to both models. The highest accuracy is 0.34 for DMS and 7-APMI.

In this work, the analysis of the partitioning model accuracies according to the size of the neighbourhood were studied. Figure 4.11 presents how accuracy is changed with the increasing number of neighbours. In this case Pareto neighbourhood (APMI and CPMI) is compared with the standard n-nearest neighbourhood (AVG). The considered number of neighbours is from 2 to 10. The results summarised Tables 4.13–4.16 plus additional analysis for applying model identification to all models for soil-water endpoint (see Figure 4.11e and Figure 4.11f). The red line represent APMI, blue CPMI and green AVG methods, respectively.



(a) training **(b)** testing

(c) training **(d)** testing

(e) training **(f)** testing

Figure 4.11: Model accuracies vs size of neighbourhood for soil-water models.

The similar observation was made for the models for whole-soil endpoint. In the first case where all selected models have $R^2_{tr} > 0.7$ and negative q^2, the model identification approach is not possible. In a case when the models are not good but the q^2 is positive the model identification approach increases the predictivity of such model combination.

All methods proposed in the chapter were implemented in R [19]. The $\log P$ value, fingerprints and Tanimoto similarity were calculated using the R CDK library [92]. A number of tests were run to define the threshold τ. It is important to notice that the n-Pareto neighbourhood defines the set of at most n-Pareto points. Therefore, for the 3-Pareto neighbourhood we found chemicals that have 1, 2, or 3 Pareto neighbours for $\tau = 0.4$ for the entire TPT dataset. For the 5-Pareto neighbourhood $\tau = 0.7$ and for the 10-Pareto neighbourhood we considered all Pareto neighbours. This shows that the size of the Pareto neighbourhood depends on a size of the available chemical space and may vary for different endpoints. Also, looking at the results for APMI in Tables 4.8-4.10 one can notice that it is not worth considering all Pareto points, and that the size of the Pareto neighbourhood depends on chemical compound similarities.

4.6 Summary

The large volume of publicly available toxicity information and the increasing number of good quality models managed properly can become useful sources of information. Models and data can be further reused within *in-silico* modelling to speed up the process of high-throughput screening. Decision on usage of a particular model for new chemicals is a result of model analysis and validation. In this chapter we developed a framework for intelligent model management and mining that automatizes the decision making process on the choice of the best model.

Firstly, the concept of the partitioning model in terms of model identification was proposed. The main idea proposed here is to split the chemical space into disjoint model groups. Each group is assigned with a particular predictive model in order to maximize the similarity of chemicals and to minimize the model error within a group. This is clearly a bi-criteria classification problem. To construct a partitioning model, three algorithms were proposed. In the Double Min Score (DMS) algorithm, the assumption was that the model performance is equal for similar chemicals according to

the similarity hypothesis. A new item is classified to the particular group based on the nearest neighbour. Two other proposed methods (APMI and CPMI) identify a suitable model for a query chemical compound based on the model performance in a Pareto neighbourhood. The concept of Pareto optimality was recalled and lemmas were proposed for properties of Pareto points. These properties were used to build a simple yet effective method for finding a Pareto set in 2D space.

The experimental results demonstrated the advantage of the proposed approaches and indicated that the automated model identification is a promising research direction with many practical applications. An additional interesting problem is the estimation of the identified model's reliability for a new chemical compound. To address this question, a method for model interpretation is proposed in the following chapter.

Chapter 5

Model Interpretation

Model interpretation is one of the key aspects of the model evaluation process. The explanation of the relationship between model variables and the output is relatively easy for such statistical models as linear regressions thanks to the availability of model parameters and their statistical significance. For Random Forest models, this information is hidden within the model structure. This chapter presents an approach for computing feature contributions for Random Forest classification models and introduces methods for their analysis. The extensive analysis leads to a discovery of the standard behaviour of the model and allow for an additional assessment of model reliability for new data. The methodology presented in this chapter was published in [81] and [85].

5.1 Introduction

Models are used to discover interesting patterns in data or to predict a specific outcome, such as drug toxicity, client shopping purchases, or car insurance premium. They are often used to support human decisions in various business strategies. In Chapter 2, the process of model development was extensively discussed. Implementation of this process together with capturing information on how the data was harvested, how the model was built and how the model was validated, allows us to trust that the model gives reliable predictions. But, how to analyse the relation between predicted values and the training dataset? Which features contribute the most to classifying a specific instance? How to assess the reliability of a models prediction?

Answers to these questions are considered particularly valuable in such domains as chemoinformatics, bioinformatics or predictive toxicology [95]. Linear models, which assign instance-independent coefficients to all features, are the most easily interpreted. Regression coefficients represent the mean change in the response variable for one unit of change in the predictor variable while other predictors are held constant in the model. However, in the recent literature, there has been considerable focus on interpreting predictions made by non-linear models which do not render themselves to straightforward methods for the determination of variable/feature influence. In [12], the authors present a method for local interpretation of Support Vector Machine (SVM) and Random Forest models by retrieving the variable corresponding to the largest component of the decision-function gradient at any point in the model. Interpretation of classification models using local gradients is discussed in [2]. A method for visual interpretation of kernel-based prediction models is described in [38]. Another approach, which is presented in detail later, was proposed in [67] and aims at shedding light on the decision-making process of regression Random Forests.

Of interest to this chapter is the popular Random Forest model proposed by Braiman [6]: this model does not provide information on individual trees. This is why it is sometimes referred to as a "black box", a system who's internal workings are not visible and is only defined by its inputs and outputs, what makes its interpretation a difficult task [87]. Its author suggests two measures for the significance of a particular variable: the variable importance and the Gini importance [7]. The variable importance is derived from the loss of accuracy of model predictions when values of one variable are permuted between instances. Gini importance is calculated from the Gini impurity criterion used in the growing of trees in the Random Forest. However, in [104], the authors showed that the above measures are biased in favor of continuous variables and variables with many categories. They also demonstrated that the general representation of variable importance is often insufficient for the complete understanding of the relationship between input variables and the predicted value.

Following the above observation, Kuzmin et al. proposed in [67] a new technique to calculate a feature contribution, i.e., a contribution of a variable to the prediction, in a Random Forest model. Their method applies to models generated for data with numerical observed values (the observed value is a real number). Unlike the variable importance measures [7], feature contributions are computed separately for each

instance/record. They provide detailed information about relationships between variables and the predicted value: the extent and the kind of influence (positive/negative) of a given variable. This new approach was positively tested in [67] on a Quantitative Structure-Activity (QSAR) model for chemical compounds. The results were not only informative about the structure of the model but also provided valuable information for the design of new compounds.

The procedure from [67] for the computation of feature contributions applies to Random Forest models predicting numerical observed values. This chapter aims to extend it to Random Forest models with categorical predictions, i.e., where the observed value determines one from a finite set of classes. The difficulty of achieving this aim lies in the fact that a discrete set of classes does not have the algebraic structure of real numbers which the approach presented in [67] relies on. Due to the high-dimensionality of the calculated feature contributions, their direct analysis is not easy. In this chapter, three techniques for discovering class-specific feature contribution "patterns" in the decision-making process of Random Forest models are proposed: the analysis of median feature contributions, of clusters and log-likelihoods. This facilitates interpretation of model predictions as well as allowing a more detailed analysis of model reliability for new data.

5.2 Random Forest

A Random Forest (RF) model introduced by Breiman [6] is a collection of tree predictors. Each tree is grown according to the following procedure [7]:

1. the bootstrap phase: select randomly a subset of the training dataset – a local training set for growing the tree. The remaining samples in the training dataset form a so-called out-of-bag (OOB) set and are used to estimate the RF's goodness-of-fit.

2. the growing phase: grow the tree by splitting the local training set at each node according to the value of one variable from a randomly selected subset of variables (called the best split method) using the classification and regression tree (CART) method [8].

3. each tree is grown to the largest extent possible. There is no pruning.

The bootstrap and growing phases require an input of random quantities. It is assumed that these quantities are independent between trees and identically distributed. Consequently, each tree can be viewed as sampled independently from the ensemble of all tree predictors for a given training dataset.

For prediction, an instance is run through each tree in a forest down to a terminal node which assigns it a class. Predictions supplied by the trees undergo a voting process: the forest returns a class with the maximum number of votes. Draws are resolved through a random selection.

Before a feature contribution procedure can be presented, there is a need to develop a probabilistic interpretation of the forest prediction process. Denote by $C = \{C_1, C_2, \ldots, C_K\}$ the set of classes and by Δ_K the set

$$\Delta_K = \big\{(p_1, \ldots, p_K) : \sum_{k=1}^{K} p_k = 1 \text{ and } p_k \geq 0\big\}.$$

An element of Δ_K can be interpreted as a probability distribution over C. Let e_k be an element of Δ_K with 1 at position k – a probability distribution concentrated at class C_k. If a tree t predicts that an instance i belongs to a class C_k then we write $\hat{Y}_{i,t} = e_k$. This provides a mapping from predictions of a tree to the set Δ_K of probability measures on C. Let

$$\hat{Y}_i = \frac{1}{T} \sum_{t=1}^{T} \hat{Y}_{i,t}, \tag{5.1}$$

where T is the overall number of trees in the forest. Then $\hat{Y}_i \in \Delta_K$ and the prediction of the Random Forest for the instance i coincides with a class C_k for which the k-th coordinate of $\hat{Y}_i$ is maximal.

5.3 Feature Contributions for Binary Classifiers

The set Δ_K simplifies considerably when there are two classes, $K = 2$. An element $p \in \Delta_K$ is uniquely represented by its first coordinate p_1 ($p_2 = 1 - p_1$). Consequently, the set of probability distributions on C is equivalent to the probability weight assigned to class C_1.

To understand the feature contribution method, the knowledge of the tree growing process is important. In the first step of this process, the training dataset is selected and it is positioned in the tree root node. A splitting variable (feature) and a splitting value are selected and the set of instances is split between the left and the right child nodes of the root node. The procedure is repeated until all instances in a node are in the same class or further splitting does not improve prediction. The class that a tree assigns to a terminal node is determined through majority voting between instances in that node.

We will refer to instances of the local training set that pass through a given node as the training instances in this node. The fraction of the training instances in a node n belonging to class C_1 will be denoted by Y^n_{mean}. This is the probability that a randomly selected element from the training instances in this node is in the first class. In particular, a terminal node is assigned to class C_1 if $Y^n_{mean} > 0.5$ or $Y^n_{mean} = 0.5$ and in this case the draw is resolved in favor of class C_1.

The feature contribution procedure for a given instance involves two steps: 1) the calculation of local increments of feature contributions for each tree and 2) the aggregation of feature contributions over the forest. A local increment corresponding to a feature f between a parent node (p) and a child node (c) in a tree is defined as follows:

$$LI^c_f = \begin{cases} Y^c_{mean} - Y^p_{mean}, & \text{if the split in the parent is performed over the feature } f, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly to the tree growing procedure the local training set is run down a tree and at each node except the root node the local increments for a splitting feature f are calculated. They represent the change of the probability of being in class C_1 between the child node and its parent node provided that f is the splitting feature in the parent node. It is easy to show that the sum of these changes, over all features, along the path followed by an instance from the root node to the terminal node in a tree is equal to the difference between Y_{mean} in the terminal and the root node.

The contribution $FC^f_{i,t}$ of a feature f in a tree t for an instance i is equal to the sum of LI_f over all nodes on the path of instance i from the root node to a terminal node.

The contribution of a feature f for an instance i in the forest is then given by

$$FC_i^f = \frac{1}{T}\sum_{t=1}^{T} FC_{i,t}^f. \tag{5.2}$$

The feature contributions vector for an instance i consists of contributions FC_i^f of all features f.

Notice that if the following condition is satisfied:

(U) for every tree in the forest, local training instances in each terminal node are of the same class

then $\hat{Y}_i$ representing forest's prediction (5.1) can be written as

$$\hat{Y}_i = \left(Y^r + \sum_f FC_i^f,\ 1 - Y^r - \sum_f FC_i^f\right) \tag{5.3}$$

where Y^r is the coordinate-wise average of Y_{mean} over all root nodes in the forest. If this unanimity condition **(U)** holds, feature contributions can be used to retrieve predictions of the forest. Otherwise, they only allow for the interpretation of the model.

5.3.1 Example of Feature Contributions Calculation

In this section, to demonstrate the calculation of feature contributions on a toy example, a subset of the UCI Iris Dataset [116] is used. From the original dataset, ten records were selected – five for each of two types of the iris plant: versicolor (class 0) and virginica (class 1) (see Table 5.1). A plant is represented by four attributes: Sepal.Length (f1), Sepal.Width (f2), Petal.Length (f3) and Petal.Width (f4). This dataset was used to generate a Random Forest model with two trees, see Figure 5.1. In each tree, the local training dataset (LD) in the root node collects those records which were chosen by the Random Forest algorithm to build that tree. The LD sets in the child nodes correspond to the split of the above set according to the value of a selected feature (it is written between branches). This process is repeated until reaching terminal nodes of the tree. Notice that the condition **(U)** is satisfied – for both trees, each terminal node contains local training instances of the same class: Y_{mean} is either 0 or 1.

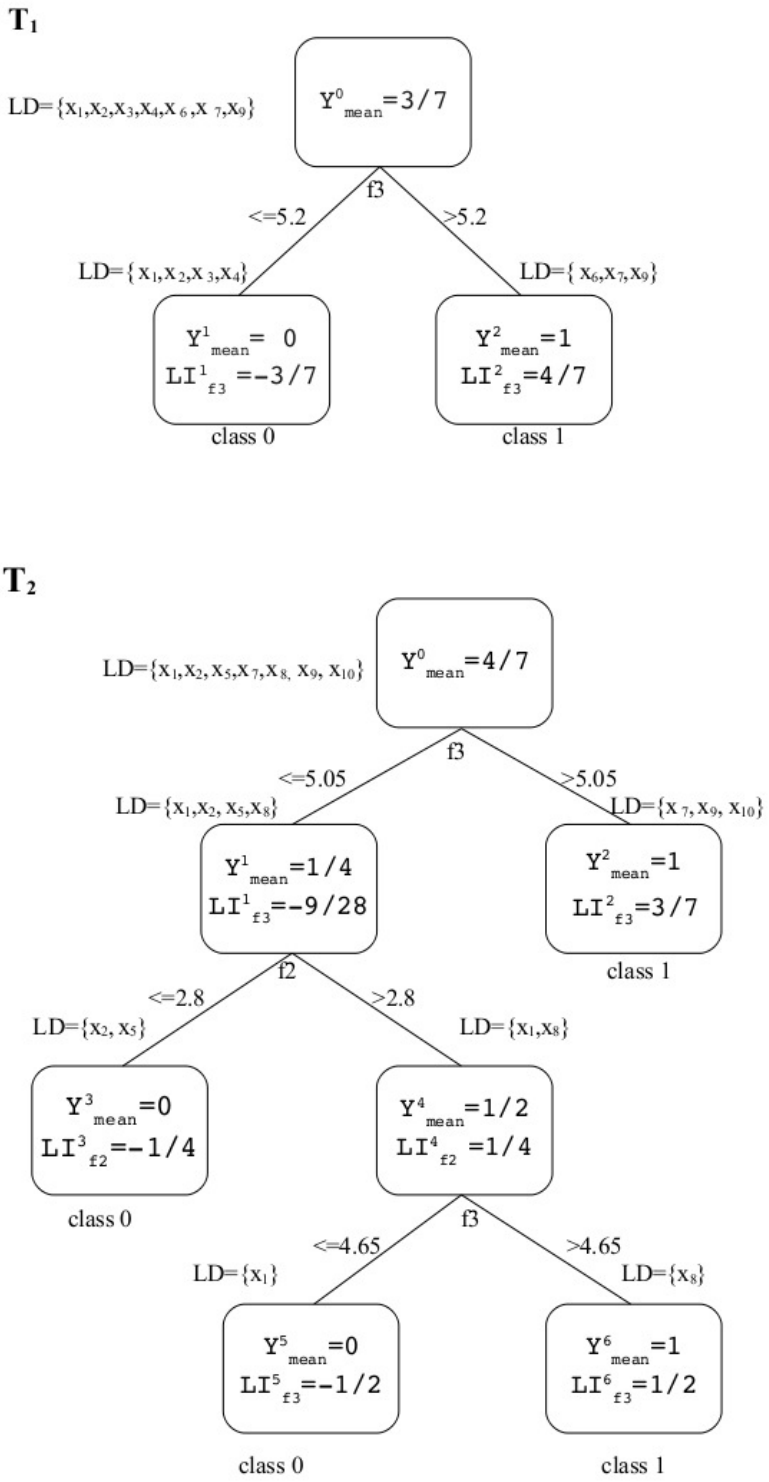


Figure 5.1: A Random Forest model for the dataset from Table 5.1.

Table 5.1: Selected records from the UCI Iris Dataset. Each record corresponds to a plant. The plants were classified as iris versicolor (class 0) and virginica (class 1).

	Sepal		Petal		
	Length (f1)	Width (f2)	Length (f3)	Width (f4)	Class
x_1	6.4	3.2	4.5	1.5	**0**
x_2	6.3	2.5	4.9	1.5	**0**
x_3	6.4	2.9	4.3	1.3	**0**
x_4	5.5	2.5	4.0	1.3	**0**
x_5	5.5	2.6	4.4	1.2	**0**
x_6	7.7	3.0	6.1	2.3	**1**
x_7	6.4	3.1	5.5	1.8	**1**
x_8	6.0	3.0	4.8	1.8	**1**
x_9	6.7	3.3	5.7	2.5	**1**
x_{10}	6.5	3.0	5.2	2.0	**1**

The process of calculating feature contributions runs in 2 steps: the determination of local increments for each node in the forest (a preprocessing step) and the calculation of feature contributions for a particular instance. Figure 5.1 shows Y^n_{mean} and the local increment LI^c_f for a splitting feature f in each node. Having computed these values, we can calculate feature contributions for an instance by running it through both trees and summing local increments of each of the four features. For example, the contribution of a given feature for the instance x_1 is calculated by summing local increments for that feature along the path $p_1 = n_0 \to n_1$ in tree T_1 and the path $p_2 = n_0 \to n_1 \to n_4 \to n_5$ in tree T_2. According to Formula (5.2) the contribution of feature f2 is calculated as

$$FC^{f2}_{x_1} = \frac{1}{2}\Big(0 + \frac{1}{4}\Big) = 0.125$$

and the contribution of feature f3 is

$$FC^{f3}_{x_1} = \frac{1}{2}\Big(- \frac{3}{7} - \frac{9}{28} - \frac{1}{2}\Big) = -0.625.$$

The contributions of features f1 and f4 are equal to 0 because these attributes are not used in any decision made by the forest. The predicted probability $\hat{Y}_{x_1}$ that x_1 belongs

to class 1 (see Formula (5.3)) is

$$\hat{Y}_{x_1} = \underbrace{\frac{1}{2}\left(\frac{3}{7} + \frac{4}{7}\right)}_{\hat{Y}^r} + \underbrace{\left(0 + 0.125 - 0.625 + 0\right)}_{\sum_f FC^f_{x_1}} = 0.0$$

Table 5.2 collects feature contributions for all 10 records in the example dataset. These results can be interpreted as follows:

- for instances x_1, x_3, the contribution of f2 is positive, i.e., the value of this feature increases the probability of being in class 1 by 0.125. However, the large negative contribution of the feature f3 implies that the value of this feature for instances x_1 and x_3 was decisive in assigning the class 0 by the forest.
- for instances x_6, x_7, x_9, the decision is based only on the feature f3.
- for instances x_2, x_4, x_5, the contribution of both features leads the forest decision towards class 0.
- for instances x_8, x_{10}, $\hat{Y}$ is 0.5. This corresponds to the case where one of the trees points to class 0 and the other to class 1. In practical applications, such situations are resolved through a random selection of the class. Since $\hat{Y}^r = 0.5$, the lack of decision of the forest has a clear interpretation in terms of feature contributions: the amount of evidence in favour of one class is counterbalanced by the evidence pointing towards the other.

5.4 Feature Contributions for General Classifiers

When $K > 2$, the set Δ_K cannot be described by a one-dimensional value as above. We, therefore, generalize the quantities introduced in the previous section to a multi-dimensional case. Y^n_{mean} in a node n is an element of Δ_K, whose k-th coordinate, $k = 1, 2, \ldots, K$, is defined as

$$Y^n_{mean,k} = \frac{|\{i \in TS(n) : \ i \in C_k\}|}{|TS(n)|}, \tag{5.4}$$

Table 5.2: Feature contributions for the Random Forest model from Figure 5.1.

		Sepal		Petal		
	$\hat{Y}$	Length (f1)	Width (f2)	Length (f3)	Width (f4)	Prediction
x_1	0.0	0	0.125	-0.625	0	**0**
x_2	0.0	0	-0.125	-0.375	0	**0**
x_3	0.0	0	0.125	-0.625	0	**0**
x_4	0.0	0	-0.125	-0.375	0	**0**
x_5	0.0	0	-0.125	-0.375	0	**0**
x_6	1.0	0	0	0.5	0	**1**
x_7	1.0	0	0	0.5	0	**1**
x_8	0.5	0	0.125	-0.125	0	**?**
x_9	1.0	0	0	0.5	0	**1**
x_{10}	0.5	0	0	0	0	**?**

where $TS(n)$ is the set of training instances in the node n and $|\cdot|$ denotes the number of elements of a set. Hence, if an instance is selected randomly from a local training set in a node n, the probability that this instance is in class C_k is given by the k-th coordinate of the vector Y^n_{mean}. Local increment LI^c_f is analogously generalized to a multidimensional case:

$$LI^c_f = \begin{cases} Y^c_{mean} - Y^p_{mean}, & \text{if the split in the parent is performed over the feature } f, \\ \underbrace{(0, \ldots, 0)}_{K \text{ times}}, & \text{otherwise,} \end{cases}$$

where the difference is computed coordinate-wise. Similarly, $FC^f_{i,t}$ and FC^f_i are extended to vector-valued quantities. Notice that if the condition **(U)** is satisfied, Equation (5.3) holds with Y^r being a coordinate-wise average of vectors Y^n_{mean} over all root nodes n in the forest.

Take an instance i and let C_k be the class to which the forest assigns this instance. Our aim is to understand which variables/features drove the forest to make that prediction. We argue that the crucial fact is the one which explains the value of the k-th coordinate of $\hat{Y}_i$. Hence, we want to study the k-th coordinate of FC^f_i for all features f.

Algorithm 8 FC(RF,i)

1: $k \leftarrow forest\_predict(RF, i)$
2: $FC \leftarrow vector(features)$
3: **for** each tree T in forest F **do**
4: $parent \leftarrow root(T)$
5: **while** $parent$! = TERMINAL **do**
6: $f \leftarrow SplitFeature(parent)$
7: **if** $i[f] <= SplitValue(parent)$ **then**
8: $child \leftarrow leftChild(parent)$
9: **else**
10: $child \leftarrow rightChild(parent)$
11: **end if**
12: $FC[f] \leftarrow FC[f] + Y^{child}_{mean,k} - Y^{parent}_{mean,k}$
13: $parent \leftarrow child$
14: **end while**
15: **end for**
16: $FC \leftarrow FC$ / nTrees(F)
17: **return** FC

Pseudo-code to calculate feature contributions (FC) for a particular instance towards the class predicted by the Random Forest is presented in Algorithm 8. Its inputs consist of a Random Forest model RF and an instance i which is represented as a vector of feature values. In line 1, $k \in \{1, 2, \ldots, K\}$ is assigned the index of a class predicted by the Random Forest RF for the instance i. The following line creates a vector of real numbers indexed by features and initialized to 0. Then for each tree in the forest RF the instance i is run down the tree and feature contributions are calculated. The quantity $SplitFeature(parent)$ identifies a feature f on which the split is performed in the node $parent$. If the value $i(f)$ of that feature f for the instance i is lower or equal to the threshold $SplitValue(parent)$, the route continues to the left child of the node $parent$. Otherwise, it goes to the right child (each node in the tree has either two children or is a terminal node). A position corresponding to the feature f in the vector FC is updated according to the change of value of $Y_{mean,k}$, i.e., the k-th coordinate of Y_{mean}, between the parent and the child.

Algorithm 9 provides a sketch of the preprocessing step to compute Y^n_{mean} for all nodes n in the forest. The parameter D denotes the set of instances used for training of the forest RF. In line 2, TS is assigned the set used for growing tree T. This set

Algorithm 9 $Y_{mean}(RF, D)$

1: **for** each tree T in forest F **do**
2: $TS \leftarrow$ training set for tree T
3: use DFS algorithm to compute training sets in all other nodes n of tree T and compute the vector Y^n_{mean} according to formula (5.4).
4: **end for**

is further split in nodes according to values of splitting variables. We propose to use DFS (depth first search [17]) to traverse the tree and compute the vector Y^n_{mean} once a training set for a node n is determined. There is no need to store a training set for a node n once Y^n_{mean} has been calculated.

The above algorithms are implemented as a package randomForest Feature Contributions (rfFC) in R and published at R-Forge [82]. The detailed documentation is included in Appendix A.

5.5 Analysis of Feature Contributions

Feature contributions provide the means to understand mechanisms that lead the model towards particular predictions. This is important in chemical or biological applications where the additional knowledge of the forest's decision-making process can inform the development of new chemical compounds or explain their interactions with living organisms. Feature contributions may also be useful for assessing the reliability of model predictions for unseen instances. They provide complementary information to a forest's voting results. This section proposes three techniques for finding patterns in the way a Random Forest uses available features and linking these patterns with the forest's predictions.

5.5.1 Median Analysis

The median of a sequence of numbers is a value such that the number of elements bigger than it and the number of elements smaller than it is the same. When the number of elements in the sequence is odd, this is the central element of the sequence. Otherwise, it is common to take the midpoint between the two most central elements. In statistics, the median is an estimator of the expectation which is less affected by outliers than

the sample mean. We will use this property of the median to find a "standard level" of feature contributions for representatives of a particular class. This standard level will facilitate an understanding of which features are instrumental for the classification. It can also be used to judge the reliability of forest's prediction for an unseen instance.

For a given Random Forest model, we select those instances from the training dataset that are classified correctly. We calculate the medians of contributions of every feature separately for each class. The medians computed for one class are combined into a vector which is interpreted as providing the aforementioned "standard level" for this class. If most of the instances from the training dataset belonging to a particular class are close to the corresponding vector of medians, we may treat this vector justifiably as a standard level. When a prediction is requested for a new instance, we query the Random Forest model for the fraction of trees voting for each class and calculate feature contributions leading to its final prediction. If a high fraction of trees votes for a given class and the feature contributions are close to the standard level for this class, we may reasonably rely on the prediction. Otherwise we may doubt the Random Forest model prediction.

It may, however, happen that many instances from the training dataset correctly predicted to belong to a particular class are distant from the corresponding vector of medians. This might suggest that there is more than one standard level, i.e., there are multiple mechanisms relating features to correct classes. The next subsection presents more advanced methods capable of finding a number of standard levels – distinct patterns followed by the Random Forest model in its prediction process.

5.5.2 Cluster Analysis

Clustering is an approach for grouping elements/objects according to their similarity [37]. This allows us to discover patterns that are characteristic for a particular group. As discussed above, feature contributions in one class may have more than one "standard level". When this is discovered, clustering techniques can be employed to find if there is a small number of distinct standard levels, i.e., feature contributions of the instances in the training dataset group around a few points with only a relatively few instances being far away from them. These few instances are then treated as unusual representatives of a given class. We shall refer to clusters of instances around these

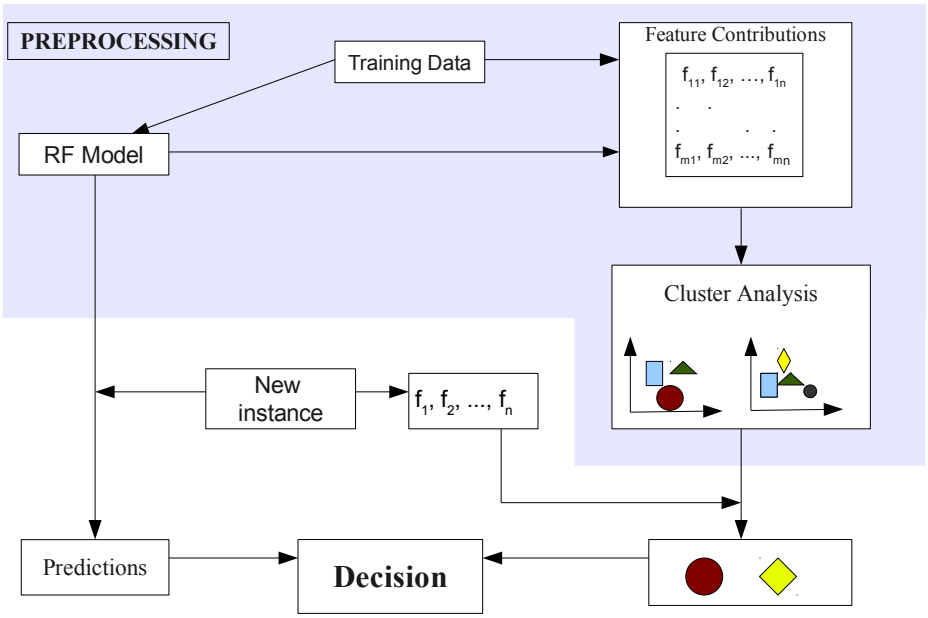


Figure 5.2: The workflow for assessing the reliability of the prediction made by a Random Forest (RF) model.

standard levels as ”core clusters”.

The analysis of core clusters can be of particular importance for applications. For example, in the classification of chemical compounds, the split into clusters may point to groups of compounds with different mechanisms of activity. We should note that the similarity of feature contributions does not imply that particular features are similar. We examined several examples and noticed that clustering based upon the feature values did not yield useful results whereas the same method applied to feature contributions was able to determine a small number of core clusters.

Figure 5.2 demonstrates the process of analysis of model reliability for a new instance using cluster analysis. In a preprocessing phase, feature contributions for instances in the training dataset are obtained. The optimal number of clusters for each class can be estimated by using one of the following methods: the Akaike information criterion (AIC), the Bayesian information criterion (BIC) or the Elbow method

[37, 90]. We noticed that these methods should not be rigidly adhered to: their underlying assumption is that the data is clustered and we only have to determine the number of these clusters. As we argued above, we expect feature contributions for various instances to be grouped into a small number of clusters and we accept a reasonable number of outliers interpreted as unusual instances for a given class. Clustering algorithms tend to push those outliers into clusters, hence increasing the number of clusters unnecessarily. We recommend, therefore, to treat the calculated optimal number of clusters as the maximum value and consecutively decrease it looking at the structure and performance of the resulting clusters: for each cluster we assess the average fraction of trees voting for the predicted class across the instances belonging to this cluster as well as the average distance from the centre of the cluster. Relatively large clusters with the former value close to 1 and the latter value small form the group of core clusters.

To assess the reliability of the model prediction for a new instance, we recommend looking at two measures: the fraction of trees voting for the predicted class as well as the cluster to which the instance is assigned based on its feature contributions. If the cluster is one of the core clusters and the distance from its centre is relatively small, the instance is a typical representative of its predicted class. This together with high decisiveness of the forest suggests that the model's prediction should be trusted. Otherwise, we should allow for an increased chance of misclassification.

5.5.3 Log-likelihood Analysis

Feature contributions for a given instance form a vector in a multi-dimensional Euclidean space. Using a popular k-mean clustering method, for each class we divide vectors corresponding to feature contributions of instances in the training dataset into groups minimizing the Euclidean distance from the centre in each group. Figure 5.3 shows a box-plot of feature contributions for instances in a core cluster in a hypothetical Random Forest model. Notice that some features are stable within a cluster – the height of the box is small. Others (F1 and F4) display higher variability. One would therefore expect that the same divergence of contributions for features F3 and F4 from their mean value should be treated differently. It is more significant for the feature F3 than for the feature F4. This is unfortunately not taken into account when the Eu-

clidean distance is considered. Here, we propose an alternative method for assessing the distance from the cluster centre which takes into account the variation of feature contributions within a cluster. Our method has probabilistic roots and we shall present it first from a statistical point of view and provide other interpretations afterwards.

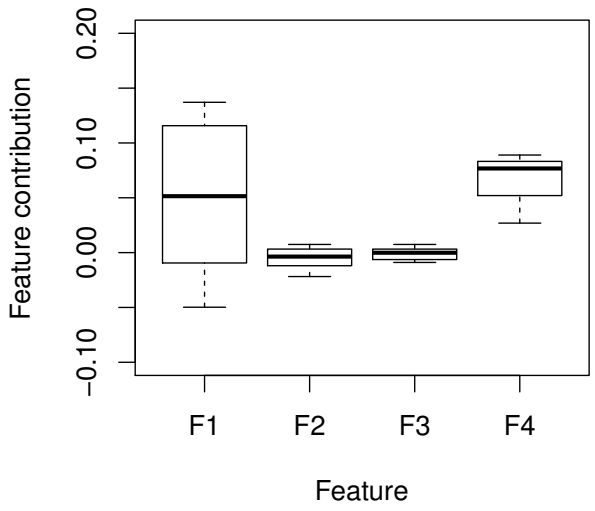


Figure 5.3: The box-plot for feature contributions within a core cluster for a hypothetical Random Forest model.

We assume that feature contributions for instances within a cluster share the same base values (μ_f) - the centre of the cluster. We attribute all discrepancies between this base value and the actual feature contributions to a random perturbation. These perturbations are assumed to be normally distributed with the mean 0 and the variance σ_f^2, where f denotes the feature. The variance of the perturbation for each feature is selected separately – we use the sample variance computed from feature contributions of instances of the training dataset belonging to this cluster. Although it is clear that perturbations for different features exhibit some dependence, it is impossible to assess it given the number of instances in a cluster and the large number of features typically in use. A covariance matrix of feature contributions has $F(F+1)/2$ distinct entries, where F is the number of features. This value is usually larger than the size of a cluster making it impossible to retrieve useful information about the dependence structure of

feature contributions. Application of more advanced methods, such as principal component analysis, is left for future research. Therefore, we resort to a common solution: we assume that the dependence between perturbations is small enough to justify treating them as independent. Summarising, our statistical model for the distribution of feature contributions within a cluster is as follows: feature contributions for instances within a cluster are composed of a base value and a random perturbation which is normally distributed and independent between features.

Take an instance i with feature contributions FC_i^f. The log-likelihood of being in a cluster with the centre (μ_f) and variances of perturbations (σ_f^2) is given by

$$LL_i = \sum_f \Big(-\frac{(FC_i^f - \mu_f)^2}{2\sigma_f^2} - \frac{1}{2}\log(2\pi\sigma_f^2) \Big). \tag{5.5}$$

The higher the log-likelihood the bigger the chance of feature contributions of the instance i to belong to the cluster. Notice that the above sum takes into account the observations we made at the beginning of this subsection. Indeed, as the second term in the sum above is independent of the considered instance, the log-likelihood is equivalent to

$$\sum_f \Big(-\frac{(FC_i^f - \mu_f)^2}{2\sigma_f^2} \Big),$$

which is the negative of the squared weighted Euclidean distance between FC_i^f and μ_f. The weights are inversely proportional to the variability of a given feature contribution in the training instances in the cluster. In our toy example of Figure 5.3, this corresponds to penalizing more for discrepancies for features F2 and F3, and significantly less for discrepancies for features F1 and F4.

In the following section, we analyse relations between the log-likelihood and classification for a UCI Breast Cancer Wisconsin Dataset.

5.6 Experimental Results

In this section, we demonstrate how the techniques from the previous section can be applied to improve understanding of a Random Forest model. We consider one example of a binary classifier using the UCI Breast Cancer Wisconsin Dataset [115] (BCW

Dataset) and one example of a general classifier for the UCI Iris Dataset [116]. We complement our studies with a robustness analysis.

5.6.1 Breast Cancer Wisconsin Dataset

The UCI Breast Cancer Wisconsin Dataset contains characteristics of cell nuclei for 569 breast tissue samples; 357 are diagnosed as benign and 212 as malignant. The characteristics were captured from a digitized image of a fine needle aspirate (FNA) of a breast mass. There are 30 features, three (the mean, the standard error and the average of the three largest values) for each of the following 10 characteristics: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. For brevity, we numbered these features from $F1$ to $F30$ according to their order in the data file.

To reduce correlation between features and facilitate model interpretation, the min-max (minimal-redundancy-maximal-relevance) method was applied and the following features were removed from the dataset: 1, 3, 8, 10, 11, 12, 13, 15, 19, 20, 21, 24, 26. A Random Forest model was generated on 2/3 randomly selected instances using 500 trees. The other 1/3 of instances formed the testing dataset. The validation showed that the model accuracy was 0.9682 (only 6 instances out of 189 were classified incorrectly); similar accuracy was achieved when the model was generated using all the features.

We applied our feature contribution algorithm to the above Random Forest binary classifier. To align notation with the rest of the chapter, we denote the class "malignant" by 1 and the class "benign" by 0. Aggregate results for the feature contributions for all training instances and both classes are presented in Figure 5.4. Light-grey bars show medians of contributions for instances of class 0, whereas black bars show medians of contributions for instances of class 1 (malignant). Notice that there are only a few significant features in the graph: F4 – the mean of the cell area, F7 – the mean of the cell concavity, F14 – the standard deviation of the cell area, F23 – the average of three largest measurements of the cell perimeter and F28 – the average of three largest measurements of concave points. This selection of significant features is perfectly in agreement with the results of the permutation based variable importance (the left panel of Figure 5.5) and the Gini importance (the right panel of Figure 5.5). Interpreting the

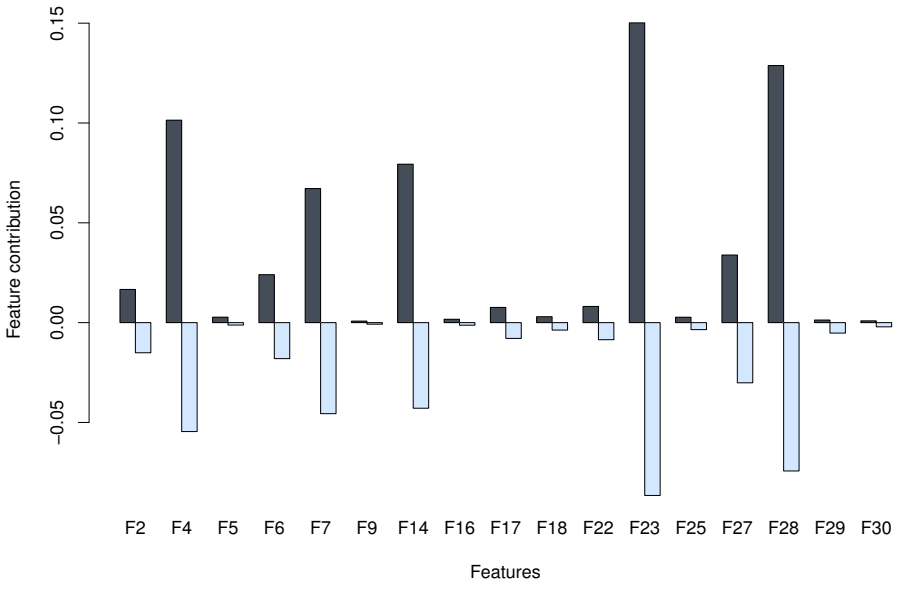


Figure 5.4: Medians of feature contributions for each class for the BCW Dataset.

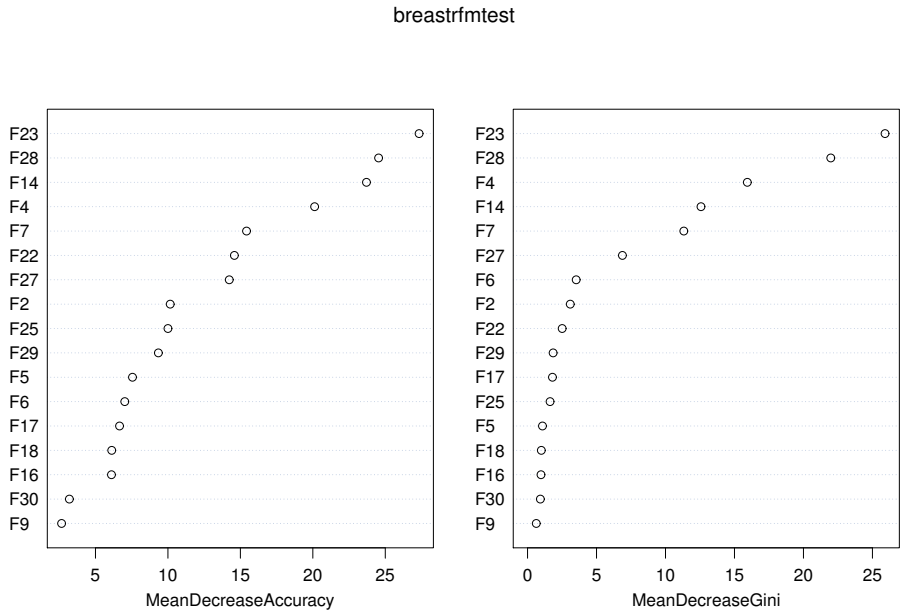


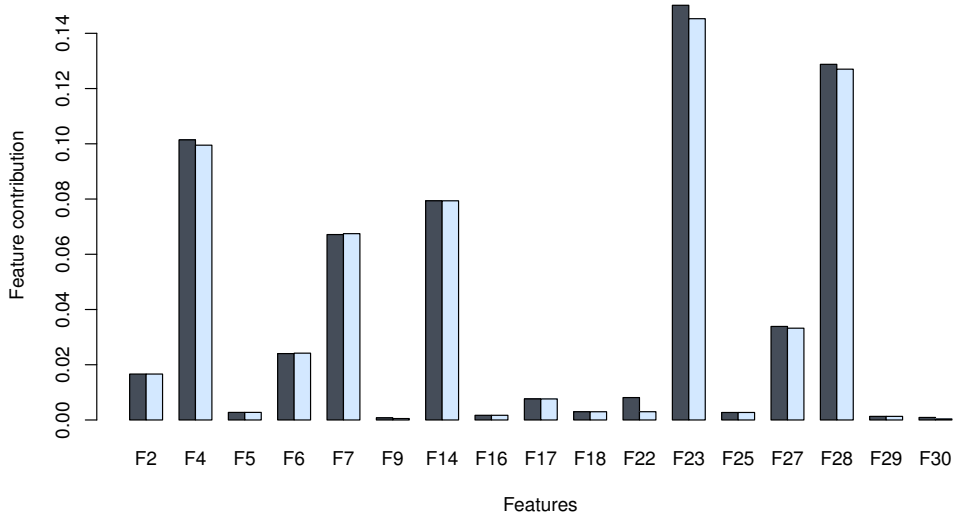
Figure 5.5: The left panel shows permutation based variable importance and the right panel displays Gini importance for a RF binary classification model developed for the BCW Dataset.

Table 5.3: Percentage of trees that vote for each class in RF model for a selection of instances from the BCW Dataset.

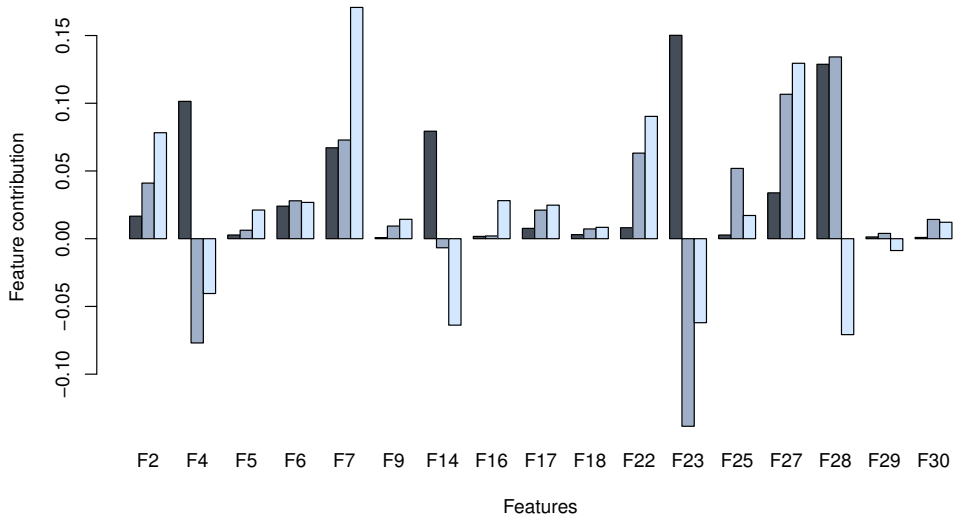
Instance Id	benign (class 0)	malignant (class 1)
3	0	1
194	0.298	0.702
537	0.234	0.766

size of bars as the level of importance of a feature, our results are in line with those provided by the Gini index. However, the main advantage of the approach presented in this chapter lies in the fact that one can study the reasons for the forest's decision for a *particular instance*.

Comparison of feature contributions for a particular instance with medians of feature contributions for all instances of one class provides valuable information about the forest's prediction. Take an instance predicted to be in class 1. In a typical case when the large majority of trees votes for class 1 the feature contributions for that instance are very close to the median values (see Figure 5.6a). This happens for around 80% of all instances from the testing dataset predicted to be in class 1. However, when the decision is less unanimous, the analysis of feature contributions may reveal interesting information. As an example, we have chosen instances 194 and 537 (see Table 5.3) which were classified correctly as malignant (class 1) by a majority of trees but with a significant number of trees expressing an opposite view. Figure 5.6b presents feature contributions for these two instances (grey and light grey bars) against the median values for class 1 (black bars). The largest differences can be seen for the contributions of very significant features F23, F4 and F14: it is highly negative for the two instances under consideration compared to a large positive value commonly found in instances of class 1. Recall that a negative value contributes towards the classification in class 0. There are also three new significant attributes (F2, F22 and F27) that contribute towards the correct classification as well as unusual contributions for features F7 and F28. These newly significant features are judged as only moderately important by both of the variable importance methods in Figure 5.5. It is, therefore, surprising to note that the contribution of these three new features was instrumental in correctly classifying instances 194 and 537 as malignant. This highlights the fact that features which may not generally be important for the model may, nonetheless, be important for classify-



(a)



(b)

Figure 5.6: Comparison of the medians of feature contributions (toward class 1) over all instances of class 1 (black bars) with a) feature contributions for instance number 3 (light-grey bars) b) feature contributions for instances number 194 (grey bars) and 537 (light-grey bars) from the BCW Dataset. The fractions of trees voting for class 0 and 1 for these three instances are collected in Table 5.3.

ing specific instances. The approach presented in this chapter is able to identify such features, whilst the standard variable importance measures for Random Forest cannot.

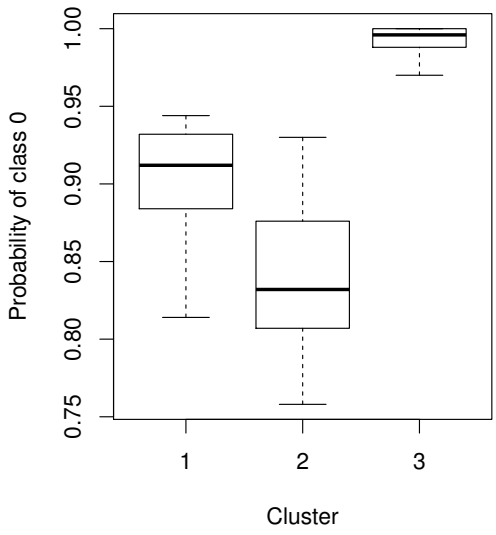
5.6.2 Cluster Analysis and Log-likelihood

The training dataset previously derived for the BCW Dataset was partitioned according to the true class labels. A clustering algorithm implemented in the R package kmeans was run separately for each class. This resulted in the determination of three clusters for class 0 and three clusters for class 1. The structure and size of clusters is presented in Table 5.4. Each class has one large cluster: cluster 3 for class 0 and cluster 2 for class 1. Both have a bigger concentration of points around the cluster centre (small average distance) than the remaining clusters. This suggests that there is exactly one core cluster corresponding to a class. This explains the success of the analysis based on the median as the vectors of medians are close to the centres of unique core clusters.

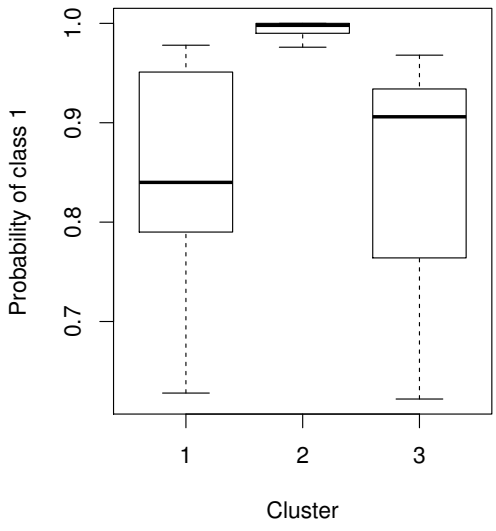
Figure 5.7 lends support to our interpretation of core clusters. The upper panel shows the box-plot of the fraction of trees voting for class 0 among training instances belonging to each of the three clusters. A value close to one represents predictions for which the forest is nearly unanimous. This is the case for cluster 3. Two other clusters comprise around 10% of the training instances for which the Random Forest model happened to be less decisive. A similar pattern can be observed in the case of class 1, see the bottom panel of the same figure. The unanimity of the forest is observed for the most numerous cluster 2 with other clusters showing lower decisiveness. The reason for this becomes clear once one looks at the variability of feature contributions within each cluster, see Figure 5.8. The upper and lower ends of the box correspond to the 75% and 25% quantiles, whereas the whiskers show the full range of the data.

Table 5.4: The structure of clusters for BCW Dataset. For each cluster, the size (the number of training instances) is reported in the left column and the average Euclidean distance from the cluster centre among the training dataset instances belonging to this cluster is displayed in the right column.

	Cluster 1		Cluster 2		Cluster 3	
	size	avg. distance	size	avg. distance	size	avg. distance
class 0	12	0.220	16	0.262	213	0.068
class 1	20	0.241	109	0.111	10	0.336

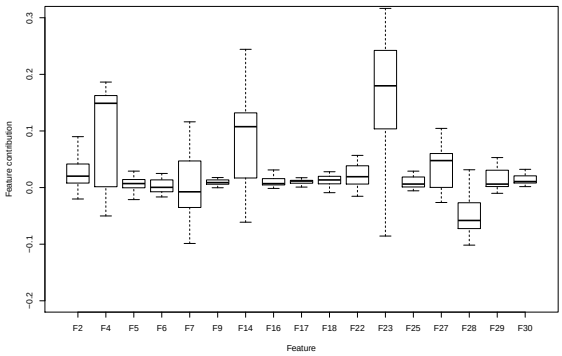


(a) Class 0

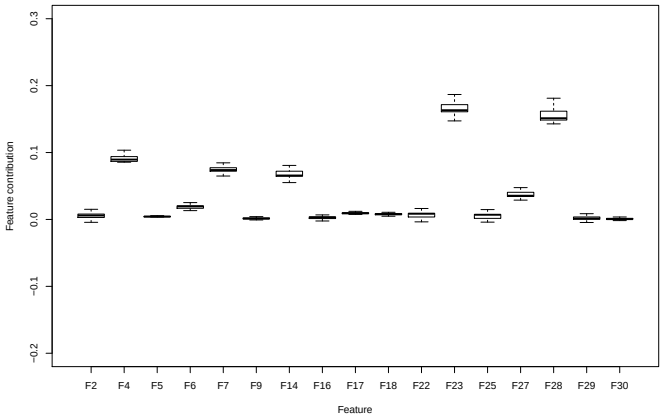


(b) Class 1

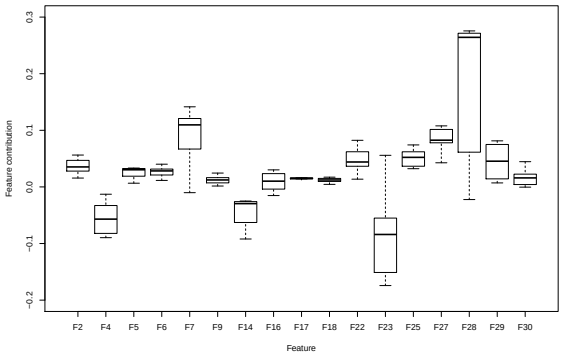
Figure 5.7: Fraction of forest trees voting for the correct class in each cluster for training part of the BCW Dataset.



(a) Cluster 1



(b) Cluster 2



(c) Cluster 3

Figure 5.8: Boxplot of feature contributions (towards class 1) for training instances in each of three clusters obtained for class 1.

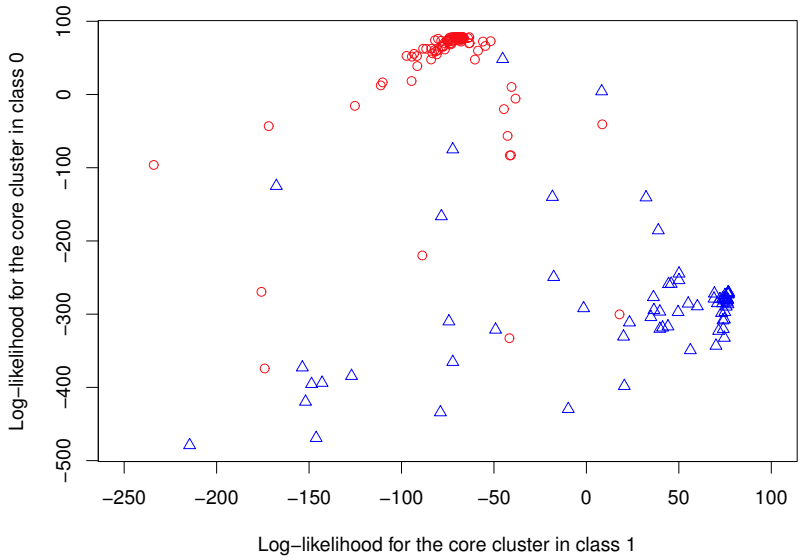


Figure 5.9: Log-likelihoods for belonging to the core cluster in class 0 (vertical axis) and class 1 (horizontal axis) for the testing dataset in BCW. Circles correspond to instances of class 0 while triangles denote instances of class 1.

Cluster 2 enjoys a minor variability of all the contributions which supports our earlier claims regarding the similarity of instances (in terms of their feature contributions) in the core class. One can see much higher variability in two remaining clusters showing that the forest used different features as evidence to classify instances in each of these clusters. Although in cluster 2 all contributions were positive, in clusters 1 and 3 there are features with negative contributions. Recall that a negative value of a feature contribution provides evidence against being in the corresponding class, here class 1 (malignant).

Based on the observation that clusters correspond to a particular decision-making route for the Random Forest model, we introduced the log-likelihood as a way to assess the distance of a given instance from the cluster centre, or, in a probabilistic interpretation, to compute the likelihood that the instance belongs to the given cluster. The likelihood is obtained by applying the exponential function to the log-likelihood. It should however be clarified that one cannot compare the likelihood for the core cluster in class 0 with the likelihood for the core cluster in class 1. The likelihood can only be

used for comparisons within one cluster: having two instances we can say which one is more likely to belong to a given cluster. By comparing it to a typical likelihood for training instances in a given cluster we can further draw conclusions about how well an instance fits that cluster. Figure 5.9 presents the log-likelihoods for the two core clusters (one for each class) for instances from the testing dataset. Shapes are used to mark instances belonging to each class: circles for class 0 and triangles for class 1. Notice that likelihoods provide a very good split between classes: instances belonging to class 0 have a high log-likelihood for the core cluster of class 0 and rather low log-likelihood for the core cluster of class 1. And vice-versa for instances of class 1.

5.6.3 Iris Dataset

In this section we use the UCI Iris Dataset [116] to demonstrate interpretability of feature contributions for multi class classification models. We generated a Random Forest model on 100 randomly selected instances. The remaining 50 instances were used to assess the accuracy of the model: 47 out of 50 instances were correctly classified. Then we applied our approach for determining the feature contributions for the generated model. Figure 5.10 presents medians of feature contributions for each of the three classes. In contrast to the binary classification case, the medians are positive for all classes. A positive feature contribution for a given class means that the value of this feature directs the forest towards assigning this class. A negative value points towards the other classes.

Feature contributions provide valuable information about the reliability of Random Forest predictions for a particular instance. It is commonly assumed that the more trees voting for a particular class, the higher the chance that the forest decision is correct. We argue that the analysis of feature contributions offers a more refined picture. As an example, take two instances: 120 and 150. The first one was classified in class Versicolour (88% of trees voted for this class). The second one was assigned class Virginica with 86% of trees voting for this class. We are, therefore, tempted to trust both of these predictions to the same extent. Table 5.5 collects feature contributions for these instances towards their predicted classes. Recall that the highest contribution to the decision is commonly attributed to features 3 (Petal.Length) and 4 (Petal Width), see Figure 5.10. These features also make the highest contributions to the predicted

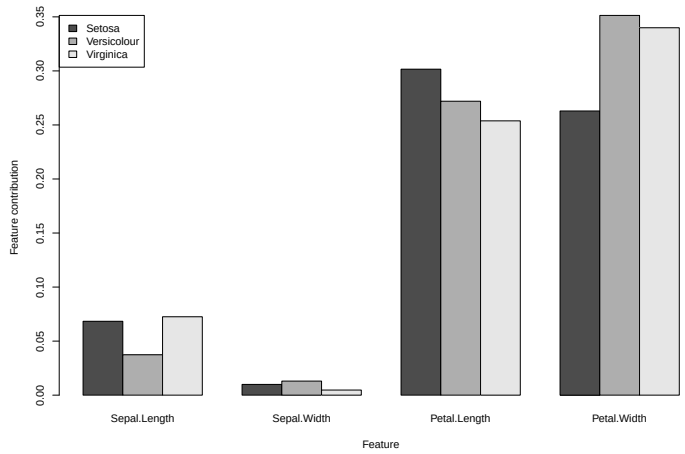


Figure 5.10: Medians of feature contributions for each class for the UCI Iris Dataset.

Table 5.5: Feature contributions towards predicted classes for selected instances from the UCI Iris Dataset.

Instance	Sepal		Petal	
	Length	Width	Length	Width
120	0.059	0.014	0.053	0.448
150	-0.097	0.035	0.259	0.339

class for instance 150. The indecisiveness of the forest may stem from an unusual value for the feature 1 (Sepal.Length) which points towards a different class. In contrast, the instance 120 shows standard (low) contributions of the first two features and unusual contributions of the last two features: very low for feature 3 and high for feature 4. Recall that features 3 and 4 tend to contribute most to the forest's decision (see Figure 5.10) with values between 0.25 and 0.35. The low value for feature 3 is non-standard for its predicted class, which increases the chance of it being wrongly classified. Indeed, both instances belong to class Virginica while the forest classified the instance 120 wrongly as class Versicolour and the instance 150 correctly as class Virginica.

The cluster analysis of feature contributions for the UCI Iris Dataset revealed that it is sufficient to consider only two clusters for each class. Cluster sizes are 4 and 38 for class Setosa, 2 and 25 for class Versicolour and 3 and 28 for class Virginica. Core

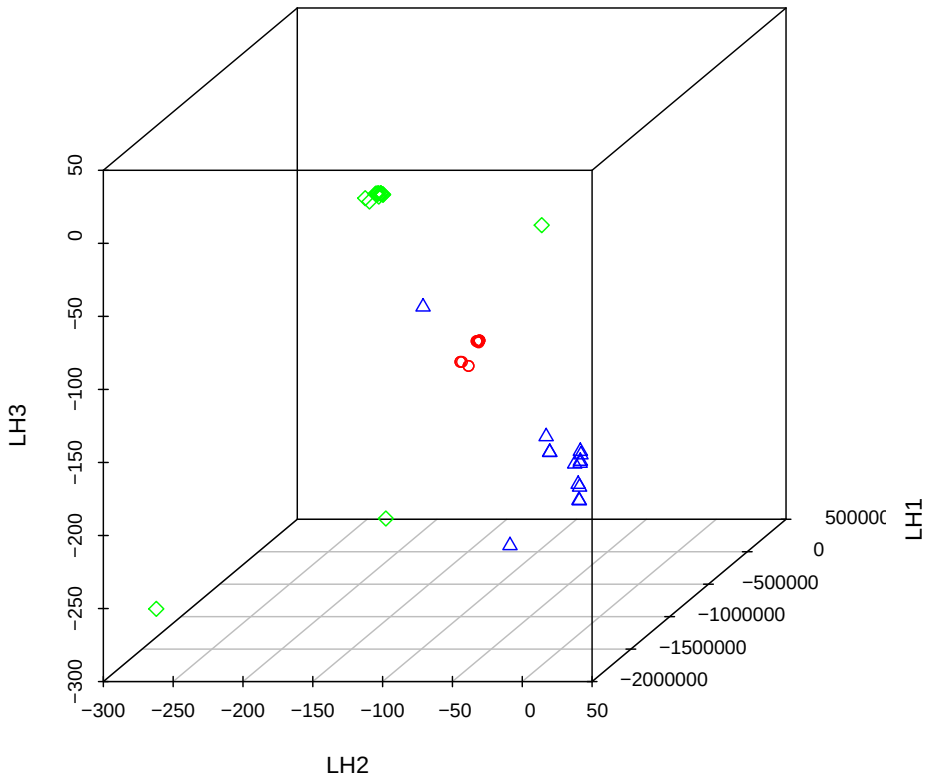


Figure 5.11: Log-likelihoods for all instances in UCI Iris Dataset towards core clusters for each class. Circles represent the Setosa class, triangles represent Versicolour and diamonds represent the Virginica class. Points corresponding to the same class tend to group together and there are only four instances that are far from their cores.

clusters were straightforward to determine: for each class, the largest of the two clusters was selected as the core cluster. Figure 5.11 displays an analysis of log-likelihoods for all instances in the dataset. For every instance, we computed feature contributions towards each class and calculated log-likelihoods of being in the core clusters of the respective classes. On the graph, each point represents one instance. The coordinate LH1 is the log-likelihood for the core cluster of class Setosa, the coordinate LH2 is the log-likelihood for the core cluster of class Versicolour and the coordinate LH3 is

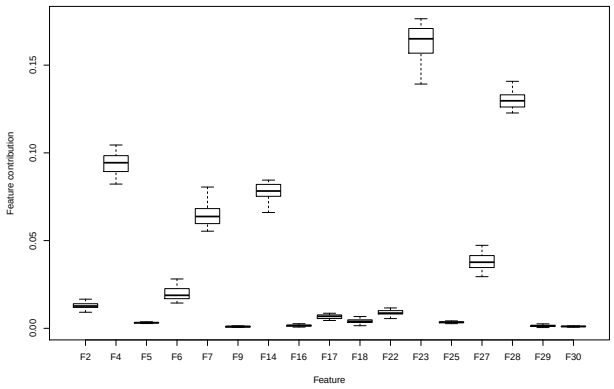
the log-likelihood for the core cluster of class Virginica. Shapes of points show the true classification: class Setosa is represented by circles, Versicolour by triangles and Virginica by diamonds. Notice that points corresponding to instances of the same class tend to group together. This can be interpreted as the existence of coherent patterns in the reasoning of the Random Forest model.

5.6.4 Robustness Analysis

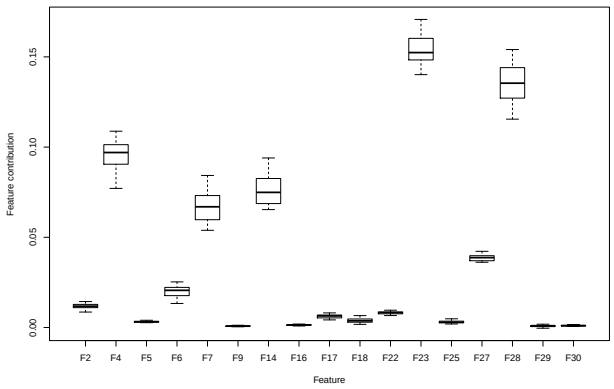
For the validity of the study of feature contributions, it is crucial that the results are not artefacts of one particular realization of a Random Forest model but that they convey actual information held by the data. We therefore propose a method for robustness analysis of feature contributions. We will use the UCI Breast Cancer Wisconsin Dataset studied in Subsection 5.6.1 as an example.

We removed instance number 3 from the original dataset to allow us to perform tests with an unseen instance. We generated 100 Random Forest models with 500 trees with each model built using an independent randomly generated training set with $379 \approx 2/3 \cdot 568$ instances. The rest of the dataset for each model was used for its validation. The average model accuracy was 0.963. For each generated model, we collected medians of feature contributions separately for training and testing datasets and each class. The variation of these quantities over models for class 1 and the training dataset are presented using a box plot in Figure 5.12a. The top of the box is the 75% quantile, the bottom is the 25% quantile, while the bold line in the middle is the median (recalling that this is the median of the median feature contributions across multiple models). Whiskers show the extent of minimal and maximal values for each feature contribution. Notice that the variation between simulations is moderate and conclusions drawn for one realization of the Random Forest model in Subsection 5.6.1 would hold for each of the generated 100 Random Forest models.

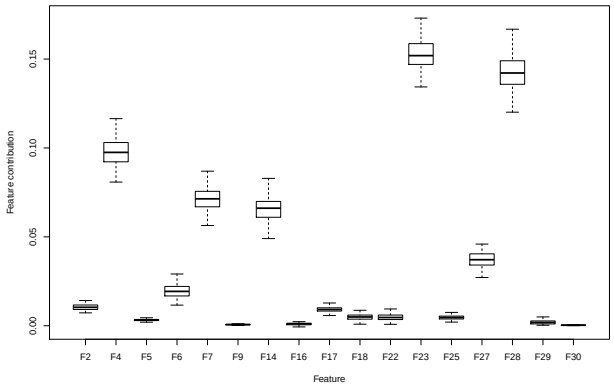
A testing dataset contains those instances that do not take part in the model generation. One can, therefore, expect more errors in the classification of the forest, which, in effect, should imply lower stability of the calculated feature contributions. Indeed, the box plot presented in Figure 5.12b shows a slight tendency towards increased variability of the feature contributions when compared to Figure 5.12a. However, these results are qualitatively on a par with those obtained on the training datasets. We can,



(a) Medians of feature contributions for training datasets



(b) Medians of feature contributions for testing datasets



(c) Feature contributions for an unseen instance

Figure 5.12: Feature contributions towards class 1 for 100 Random Forest models for the BCW dataset.

therefore, conclude that feature contributions computed for a new (unseen) instance provide reliable information. We further tested this hypothesis by computing feature contributions for instance number 3 that did not take part in the generation of models. The statistics for feature contributions for this instance over 100 Random Forest models are shown in Figure 5.12c. Similar results were obtained for other instances.

5.7 Summary

Feature contributions provide a novel approach towards the Random Forest model interpretation. They measure the influence of variables/features on the prediction outcome and provide explanations as to why a model makes a particular decision. Although there are approaches for analysing machine learning models, the analysis of the model decision for a particular instance is still a difficult task.

The focus of this chapter was the well known Random Forest model. It ensembles a number of decision trees and the implementation of the kernel methods for model interpretation can become even more complex task in this case. This is why, we focus on extracting information from the model structure rather than providing complex methods to calculate feature contributions for a single instance. The main contribution in this part was the extension of the feature contribution method of [67] to Random Forest classification models and the design of three techniques (median, cluster analysis and log-likelihood) for finding patterns in the Random Forest's use of available features. The additional contribution was the implementation of a package for the R statistical programming language. This package has been submitted to R-Forge [82] under the name `rfFC`.

The proposed methodology was validated using UCI benchmark datasets. Experimental results showed the robustness of the proposed methodology. We also demonstrated how feature contributions can be applied to understand the dependence between instance characteristics and their predicted classification and to assess the reliability of the prediction. The relation between feature contributions and standard variable importance measures was also investigated. Currently there is ongoing research to validate this approach in predictive toxicology to interpret Random Forest models for Ames mutagenicity.

Chapter 6

Conclusions

The reuse of existing information can support decision making processes in various application domains. This has become one of the most significant challenges due to the large amount of available data and models used for data analysis. The reuse of models is the main interest of this research. This chapter summarises the work presented in the thesis highlighting main contributions, discussing novel open problems and making recommendations for future work.

6.1 Research Contributions

In the domain of life sciences the reuse of information has become crucial due to the current aims of reduction in the number of animal tests. This encourages various R&D institutions to search for alternative methods that can be used in the product (such as drugs, cosmetic, agriculture and domestic products) development processes. These methods are mostly based on the reuse of existing available information. Such information represents not only data but also models that can provide new insights or discover patterns in existing data. Although data curation, quality and integration as well as model development processes have been widely studied over the last ten years, management and efficient reuse of models has been left to users. In these days, models can be seen as important information assets, thus there is a need to provide a framework for their storage, management and efficient reuse.

Making models in predictive toxicology reliable and reusable sources of informa-

tion has became a major motivation for this research. This thesis proposed general methodologies, frameworks and algorithms, and presented solutions in the area of QSAR modelling. To define needs and aims for model storage and management, an extensive review of the toxicity data and model development processes have been presented in Chapter 2. Various toxicity frameworks have been reviewed and we found the following limitations for the model reuse:

- Existing systems allow generation and reuse of models only within their framework. To make use of existing models, users are required to register with the system and to submit data that they use for predicting a given endpoint. This discourages modellers from using such predictive toxicology systems because the data in use is often confidential.

- Decision on model reuse is left to the user. A potential user is required to make a comparison of model applicability domains and their predictivity for a given endpoint in order to decide if the model can make reliable predictions for a new chemical compound.

- Although all model representations follow the OECD principles, model exchange between various toxicity platforms is not possible. The OECD principles are incorporated within the model representation format designed for a particular system. A transfer of the model from one platform to another requires parsing of meta-information about a model.

- Models are not continuously validated. Often models are updated when model creators become aware of newly available data that can be used to validate or improve existing models; otherwise models are left without any updates. This is not a major problem for models derived from the chemical structure and those that have a large applicability domain. But for local models, it is crucial to extend the boundary of the chemical space where the model gives reliable predictions and provide limitations and conditions where the prediction can be unreliable.

The review of toxicity systems was partially published in [30] in respect to the principles of data governance, and partially in [83] according to model storage, provenance and management practices. The conclusion from these reviews was that existing toxi-

city frameworks strongly support model development processes and model storage but do not support model validation and further reuse.

In Chapter 3 a novel concept of *model governance* was proposed. The idea is to treat models as valuable information assets rather than business intelligence tools. Decision domains of the model governance were built based on the principles of data governance. The novelty of this research was defining model governance processes which aimed to:

- ensure that a model is properly used,
- validate and maintain model effectiveness,
- help understanding model weaknesses (i.e. where the model can be safely applied, how reliable it is, etc.).

In Section 3.3 three processes: model evaluation, control and validation were defined and discussed in terms of the rules and actions taken in each process. The Information Management System for Data and Model Governance was proposed in Section 3.4. The framework includes human roles in the data and model organisational level and defines responsibilities. An important part of the decision domains for model governance is the definition of model meta-data. In Section 3.6 the six rules for Minimum Information About QSAR Model Representation were proposed. These rules are designed to comply with the defined model governance processes and include information about the model development process as well as information about model performance. In contrast to other model representations, storage of this information allows the analysis and comparison of models. Providing users with information about model provenance can increase the trust in further model reuse. The analysis of model performance supports the assessment of model reliability. Based on the proposed six rules, the markup language called MIAQMR-ML was designed. The proposition of the model governance framework together with the model representation format was published in [83]. A proof of the proposed concept was the design of database and implementation of Syngenta in-house Data and Model Governance Framework presented in Section 3.7.

To decide if a model can be safely applied to new data and used to support a user decision, a novel model identification framework was proposed in Chapter 4. In this research, model identification represents the selection of a model from a group of models

coming from various sources. In Section 4.2 a partitioning model that splits a search space into groups was proposed. The partition is done by grouping similar instances maximizing the similarity of the elements within the group and minimizing the error of the model assigned to this group. The assumption of this approach was that groups are disjoint and only one model is assigned to a group. The construction of the partitioning model is a difficult task as this is a bi-criteria optimization problem and the solution has to be a trade-off between the similarity of group elements and the model performance for these elements.

In the first attempt, the problem of constructing *a partitioning model* was reduced to a one-criteria problem, see the Double Min-Score (DMS) algorithm in Section 4.3. This is a classification problem that uses two pre-defined rules: select the nearest neighbour and identify its most predictive model. The algorithm and results were published in [122].

In the second attempt, to solve the original bi-criteria problem of the partitioning model construction, the concept of *Pareto neighbourhood* was used in Section 4.4. Three new lemmas for Pareto point properties were proposed in Section 4.4.1. Based on these lemmas an algorithm that searches for Pareto points in a given search space was designed. The Pareto points for a query instance define its Pareto neighbourhood. This Pareto neighbourhood is used in the model identification process, see Section 4.4.2. There are two new algorithms proposed: the Centroid Pareto Model Identification (CPMI) and Average Pareto Model Identification (APMI). The first method identifies a model that is associated with the Pareto point for which the Euclidean distance to the neighbourhood centroid is minimal. The second method averages model errors for the instance represented by Pareto points and then the model with the smallest average error is selected. Usage of the Pareto points for model identification is a novel concept. The proposed methodology, algorithms and results for QSAR model identification were published in [84].

In Section 4.5 experimental results were discussed. In this thesis, three use cases were considered. The first one is model identification for local models with limited applicability domains. In this case, two models were used from the JRC QSAR database for the Tetratox dataset. In the second case, global models available as tools for calculating LogP were compared with the in-house Syngenta model. Finally, as the third case, models were developed and collected during a competition that aimed to calcu-

late chemical compound persistence in soil. The role of this project in the competition was the preparation of the datasets and further validation of the collected models. The experimental results demonstrated the advantage of the methods proposed in this thesis, and indicated that automated model identification is a promising research direction with many practical applications.

Another contribution of this thesis was to provide tools for model interpretation. In many life science domains, and particularly in predictive toxicology, mechanistic interpretation of model predictions is necessary to increase trust and to inform decision-making. Linear models are the easiest to be interpreted thanks to the availability of model parameters and their statistical significance. Machine learning approaches and non-linear models do not hold such transparency. Information is often hidden within the model structure. Interpretation involves measuring the influence of variables/features on the prediction outcome and providing explanations as to why a model makes a particular decision.

In Chapter 5, a model interpretation method for random forest models was proposed. This method generalises the *feature contribution* approach proposed in [67]. The original procedure for the computation of feature contributions applies to random forest models predicting numerical observed values. This chapter extended it to random forest models with categorical predictions, i.e., where the observed value determines one from a finite set of classes. The methodology was published in [81].

The second contribution in this chapter was the analysis of feature contributions. This is a novel concept showing how feature contributions can be used in order to analyse why a model makes a particular decision and whether this decision is reliable. In Section 5.5, three techniques for discovering class-specific feature contributions called "patterns" in the decision-making process of random forest models are proposed: the analysis of median feature contributions, of clusters and log-likelihoods. These methods were validated using benchmark datasets. The experimental results, in Section 5.6, showed the robustness of the proposed methodology. We also demonstrated how feature contributions can be applied to understand the dependence between instance characteristics and their predicted classification and to assess the reliability of the prediction. The relationship between feature contributions and standard variable importance measures was also investigated. The methods proposed in this section were publish in [85]. Currently we continue this research to validate these proposed methodolo-

gies in predictive toxicology to interpret random forest models for Ames mutagenicity prediction.

The last contribution for this thesis was the implementation of the R package called random forest Feature Contributions (`rfFC`). This package is currently available in R-Forge [82] and is prepared for submission to CRAN [19]. It includes feature contribution extraction as well as the calculation of changes in prediction for the varying data values. The package was implemented in collaboration with Dr. Richard Marchese Robinson, a former research associate in Product Safety department in Syngenta.

The methodologies and algorithms proposed in this thesis are not domain specific. Model governance processes and model identification were discussed with applications in predictive toxicology. They can be easily adapted and implemented in other domains. Entire project consists of three main research directions that are interlinked. This research opens new problems in model reuse. A list of possible future research projects is discussed in the following section.

6.2 Future Work

The work in this thesis consists of three research topics: model reuse, model identification and model interpretation. They are linked to each other creating a foundation for the development of a complex framework for model integration and reuse. This opens a new domain of research where models are treated as information objects and further model processing is required. In this section detailed information about future research plans is provided.

Model Governance processes presented in Chapter 3 can be understood as quality control procedures for using, improving, monitoring and maintaining models. They define actions that should be taken within each process but do not propose or implement any performance metrics that evaluate these processes. Similarly to the concept of data governance, "maturity models" [108] should be proposed in the context of model management.

The methodology for model identification presented in Chapter 4 is a starting point toward model integration and aggregation. This research can be extended by considering the following problems:

1. The methods proposed in this chapter were applied to regression models. For classification models the error of model performance can not be calculated using Formula 4.1. In this case, the alternative method can be defined as follows:

$$e_{i,j} = \begin{cases} -1, & \text{is FN,} \\ 0, & \text{is TP,} \\ 1, & \text{is FP,} \end{cases} \tag{6.1}$$

   where TP stands for True Positive (when the output was classified correctly), FP – for False Positive (error is a type I), and FN – for False Negative for i instance and j model.

2. In this research binary data (chemical fingerprints) and the Tanimoto measure for calculating data similarity were considered. The assumption was that sometimes for new instances it can be difficult to obtain descriptors that were used to build models. An interesting question is how good the proposed methods could be if we combine various similarity measures for mixed data types (combination fingerprints with qualitative descriptors). This is possible due to the general definition of the proposed framework. If we look at Definition 4.1, chemical space is defined by all possible available descriptors. For other domains, the chemical space is understood by the search space with the elements described according to the domain specification. There is a question how to deal with a sparse search space.

3. Proposed partitioning methods are based on pre-defined classification rules. The question is whether it is possible to provide machine learning approaches that are able to cluster the search space described by the partitioning model (see Definition 4.3), and how good they can be in terms of model identification.

4. In this research we assumed that clusters are disjoint subsets of search space. What if we consider that these subsets can overlap. This means that we can find groups of elements that can have more than one optimal model identified. In this case, we can consider a construction of the consensus model and implementation of ensemble methods for prediction. The consensus should be calculated from models M_i for which data $x \in D_i$.

5. Implementation of the above propositions, transfer us from model interpretation to model aggregation. This is a much more general description of model reuse and includes: a partition of the search space, identification of a model or a family model for a new unseen instance and a consensus model to predict a specific activity.

6. Since this methodology can be implemented in any application domain, the study of other similarity measures used to defined neighbourhood should also be investigated.

In many domains the interpretation of the model becomes one of the crucial elements of model validation. Knowledge of how models make predictions increases the trust in the model reliability. In machine learning there are many methods that allow us to generate a good predictive models. Examples of methods include: Decision Trees, Support Vector Machine (SVM), Random Forest and Artificial Neural Network. They are often used to analyse large scale data where standard methods have problems with the size of data. In Chapter 5, a method for interpreting random forest model prediction was proposed. This research can be extended to:

1. Implementation of feature contributions method to a single decision tree. The difficulty here will be to propose methods that calculate local increments of feature contributions for trees that do not follow CART.

2. Calculation of the feature contributions for artificial neural network models. An interesting question is if it is possible to extract this information from the structure of the model. If yes, how then the weights on nodes in all layers should be analysed.

3. Further analysis of feature contributions. For example, in regression random forest models feature contributions represent a change in the forest prediction from the model that uses the average as a predictor. For classification models, the contributions represent the change of probability that an instance belongs to a given class. This is important knowledge for domains such as chemistry or biology where the importance of a particular substructure/atom or gene can be investigated. This approach is straightforward for binary data where the pres-

ence or absence of a particular feature influences model prediction. But how to analyse and interpret features with constant/categorical values?

4. Implementation of visualisation methods for feature contributions. It is clear that for large scale data it is not possible to analyse contributions manually. This research proposed three methods that allow the interpretation and basic visualisation but the current `rfFC` package version does not include them. It is planned to add these methods to the package.

There is still a lot of work required to provide a flexible, transparent, interoperable, efficient, widely acceptable framework for model integration and aggregation. Some of the future work suggested above are subjects of ongoing research. Currently, there is ongoing research to build consensus models to aggregate a subset of identified models for new instances. There is also ongoing collaboration with Dr. Richard Marchese Robinson from Liverpool John Moores University to use feature contribution approach in cheminformatics applications.

References

[1] S. Akhondi, J. Kors, and S. Muresan, "Consistency of systematic chemical identifiers within and between small-molecule databases," *Journal of Cheminformatics*, vol. 4, no. 1, p. 35, 2012.

[2] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Muller, "How to explain individual classification decisions," *Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

[3] K. V. Balakin and S. Ekins, *Pharmaceutical Data Mining: Approaches and Applications for Drug Discovery*, ser. Wiley Series on Technologies for the Pharmaceutical Industry. Wiley, 2009.

[4] R. Benigni, C. Bossa, T. Netzeva, A. Rodomonte, and I. Tsakovska, "Mechanistic (Q)SAR of aromatic amines: new models for discriminating between homocyclic mutagens and nonmutagens, and validation of models for carcinogens." *Environmental and Molecular Mutagenesis*, vol. 48, no. 9, pp. 754–771, 2007.

[5] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron, "Minimum information about a microarray experiment (MIAME) – toward standards for microarray data," *Nature Genetics*, vol. 29, no. 4, pp. 365–371, 2001.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] L. Breiman and A. Cutler. (2008) Random forests. [Online]. Available: http://www.stat.berkeley.edu/~breiman/RandomForests/ [Accessed: 20 January 2014].

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, ser. Statistics/Probability Series. Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984.

[9] F. Buchwald, T. Girschick, E. Frank, and S. Kramer, "Fast conditional density estimation for quantitative structure-activity relationships," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press, 2010, pp. 1268–1273.

[10] R. Burns. Supervisory insights. [Online]. Available: http://www.fdic.gov/regulations/examinations/supervisory/insights/siwin05/article01_model_governance.html [Accessed: 20 January 2013].

[11] D. Cao, Y. Liang, Q. Xu, H. Li, and X. Chen, "A new strategy of outlier detection for QSAR/QSPR," *Journal of Computational Chemistry*, vol. 31, no. 3, pp. 592–602, 2010.

[12] L. Carlsson, E. A. Helgee, and S. Boyer, "Interpretation of nonlinear QSAR models applied to ames mutagenicity data," *Journal of Chemical Information and Modeling*, vol. 49, no. 11, pp. 2551–2558, 2009.

[13] J. Cartmell, S. Enoch, D. Krstajic, and D. Leahy, "Automated QSPR through competitive workflow," *Journal of Computer-Aided Molecular Design*, vol. 19, no. 11, pp. 821–833, 2005.

[14] Chemical Abstracts Service. CAS. [Online]. Available: http://www.cas.org/content/chemical-substances [Accessed: 20 January 2014].

[15] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *Journal of Machine Learning Research*, vol. 10, pp. 747–776, 2009.

[16] N. R. C. Committee on Models in the Regulatory Decision Process, *Models in Environmental Regulatory Decision Making*. The National Academies Press, 2007.

[17] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[18] G. Cramer, A. Ford, and L. Hall, "Estimation of toxic hazard - a decision tree approach," *Food and Cosmetic Toxicology*, vol. 16, pp. 255–276, 1978.

[19] CRAN. The Comprehensive R Archive Network. [Online]. Available: http://cran.r-project.org/ [Accessed: 18 January 2014].

[20] Data Governance Institute. Data governance. [Online]. Available: http://www.datagovernance.com/adg_data_governance_basics.html [Accessed: 15 January 2014].

[21] Data Mining Group. Predictive Model Markup Language (PMML). [Online]. Available: http://www.dmg.org/v4-0-1/GeneralStructure.html [Accessed: 15 January 2014].

[22] Economist Intelligence Unit. (2008) The future of enterprise information governance. [Online]. Available: http://www.emc.com/collateral/analyst-reports/economist-intell-unit-info-governence.pdf [Accessed: 20 May 2013].

[23] M. Ehrgott, *Multicriteria Optimization*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[24] M. Ehrgott, "Vilfredo Pareto and multi-objective optimization," *Documenta Mathematica*, vol. Optimization Stories, pp. 447–453, 2012.

[25] S. Ekins and A. J. Williams, "Precompetitive preclinical ADME/Tox data: set it free on the web to facilitate computational model building and assist drug development," *Lab on a Chip*, vol. 10, no. 1, pp. 13–22, 2010.

[26] C. Ellison, M. Cronin, J. Madden, and T. Schultz, "Definition of the structural domain of the baseline non-polar narcosis model for Tetrahymena pyriformis," *SAR and QSAR in Environmental Research*, vol. 19, no. 7-8, pp. 751–783, 2008.

[27] S. Enoch, M. C. Cronin, T. Shultz, and J. Madden, "An evaluation of global QSAR models for the prediction of the toxicity of phenols to Tetrahymena pyriformis," *Chemosphere*, vol. 71, no. 7, pp. 1225–1232, 2008.

[28] EPA. Estimation Program Interface (EPI) Suite. [Online]. Available: http://www.epa.gov/oppt/exposure/pubs/episuite.htm [Accessed: 20 January 2013].

[29] D. Fourches, E. Muratov, and A. Tropsha, "Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research," *Journal of Chemical Information and Modeling*, vol. 50, no. 7, pp. 1189–1204, 2010.

[30] X. Fu, A. Wojak, D. Neagu, M. Ridley, and K. Travis, "Data governance in predictive toxicology: A review," *Journal of Cheminformatics*, vol. 3, no. 1, p. 24, 2011.

[31] F. Garzotto, L. Mainetti, and P. Paolini, "Information reuse in hypermedia applications," in *Proceedings of the the Seventh ACM Conference on Hypertext*, ser. HYPERTEXT '96. New York, NY, USA: ACM, 1996, pp. 93–104.

[32] J. Gasteiger, Ed., *Handbook of Chemoinformatics: From Data to Knowledge*. John Wiley and Sons Inc, 2003.

[33] A. Golbraikh and A. Tropsha, "Beware of q^2," *Journal of Molecular Graphics and Modeling*, vol. 20, no. 4, pp. 269–276, 2002.

[34] P. Gramatica, "Principles of QSAR models validation: internal and external," *QSAR and Combinatorial Science*, vol. 26, no. 5, pp. 694–7012, 2007.

[35] P. Gramatica, "On the development and validation of QSAR models," in *Computational Toxicology*, ser. Methods in Molecular Biology, B. Reisfeld and A. N. Mayeno, Eds. Humana Press, 2013, vol. 930, pp. 499–526.

[36] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[37] D. J. Hand, P. Smyth, and H. Mannila, *Principles of data mining*. Cambridge, MA, USA: MIT Press, 2001.

[38] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, and K.-R. Muller, "Visual interpretation of kernel-based prediction models," *Molecular Informatics*, vol. 30, no. 9, pp. 817–826, 2011.

[39] B. Hardy, N. Douglas, C. Helma, M. Rautenberg, N. Jeliazkova, V. Jeliazkov, I. Nikolova, R. Benigni, O. Tcheremenskaia, S. Kramer, T. Girschick, F. Buchwald, J. Wicker, A. Karwath, M. Gutlein, A. Maunz, H. Sarimveis, G. Melagraki, A. Afantitis, P. Sopasakis, D. Gallagher, V. Poroikov, D. Filimonov, A. Zakharov, A. Lagunin, T. Gloriozova, S. Novikov, N. Skvortsova, D. Druzhilovsky, S. Chawla, I. Ghosh, S. Ray, H. Patel, and S. Escher, "Collaborative development of predictive toxicology applications," *Journal of Cheminformatics*, vol. 2, no. 1, p. 7, 2010.

[40] J. A. Hartigan and M. A. Wong, "A K-Means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[41] C. Helma, Ed., *Predictive Toxicology*. Taylor & Francis Group, 2005.

[42] IdeaConnection. [Online]. Available: http://www.ideaconnection.com/ [Accessed: 20 October 2013].

[43] Ideaconsult Ltd. [Online]. Available: http://ideaconsult.net/ [Accessed: 20 January 2014].

[44] Ideaconsult Ltd. Ambit. [Online]. Available: http://ambit.sourceforge.net [Accessed: 20 January 2014].

[45] INCHEMICOTOX. In Chemico Test Results. [Online]. Available: http://www.inchemicotox.org/results/ [Accessed: 13 January 2014].

[46] INKSPOT. [Online]. Available: http://www.inkspotscience.com/ [Accessed: 22 January 2014].

[47] ISDA. Microarray Data Analysis. [Online]. Available: http://isda.ncsa.uiuc.edu/Microarrays/ [Accessed: 22 January 2014].

[48] Istituto di Ricerche Farmacologiche Mario Negri. CAESAR. [Online]. Available: http://www.caesar-project.eu/ [Accessed: 22 January 2014].

[49] S. Izrailev and D. K. Agrafiotis, "A method for quantifying and visualizing the diversity of QSAR models," *Journal of Molecular Graphics and Modelling*, vol. 22, no. 4, pp. 275–284, 2004.

[50] P. Jackson, *Introduction to Expert Systems*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[51] M. Jahnson and G. Maggiora, *Concept of Application of Molecular Similarity*. John Willey & Sons, 1990.

[52] R. C. James, S. M. Roberts, and P. L. Williams, *General Principles of Toxicology*. John Wiley & Sons, Inc., 2003, pp. 1–34.

[53] J. S. Jaworska, N. Nikolova-Jelizkova, and T. Aldenerg, " QSAR applicability domain estimation by projection of the training set in descriptor space: A review." *ATLA. Alternatives to laboratory animals*, vol. 33, no. 5, pp. 445–459, 2005.

[54] Joint Research Centre (IHCP). JRC QSAR Model Database. [Online]. Available: http://qsardb.jrc.ec.europa.eu/qmrf/ [Accessed: 10 November 2012].

[55] Joint Research Centre (IHCP). QSAR Model Reporting Formats (QMRF). [Online]. Available: http://ihcp.jrc.ec.europa.eu/our_labs/predictive_toxicology/qsar_tools/QRF [Accessed: 10 January 2014].

[56] Joint Research Centre (IHCP). Toxmatch. [Online]. Available: http://ihcp.jrc.ec.europa.eu/our_labs/predictive_toxicology/qsar_tools/toxmatch [Accessed: 10 January 2014].

[57] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[58] R. N. Jorissen and M. K. Gilson, "Virtual screening of molecular databases using a support vector machine," *Journal of Chemical Information and Modeling*, vol. 45, no. 3, pp. 549–561, 2005.

[59] R. Judson, "Public databases supporting computational toxicology," *Journal of Toxicology and Environmental Health, Part B*, vol. 13, no. 2, pp. 218–231, 2010.

[60] G. W. Kauffman and P. C. Jurs, "QSAR and k-nearest neighbor classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors," *Journal of Chemical Information and Computer Sciences*, vol. 41, no. 6, pp. 1553–1560, 2001.

[61] R. J. Kavlock. (2004) A framework for computational toxicology research. [Online]. Available: http://www.epa.gov/comptox/download_files/basic_information/comptoxframework06_02_04.pdf [Accessed: 22 January 2014].

[62] V. Khatri and C. V. Brown, "Designing data governance," *Communications of the ACM*, vol. 53, no. 1, pp. 148–152, 2010.

[63] H. J. Klimisch, M. Andreae, and U. Tillmann, "A systematic approach for evaluating the quality of experimental toxicological and ecotoxicological data," *Regulatory Toxicology and Pharmacology*, vol. 25, no. 1, pp. 1–5, 1997.

[64] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[65] M. Kooper, R. Maes, and E. R. Lindgreen, "On the governance of information: Introducing a new concept of governance to support the management of information," *International Journal of Information Management*, vol. 31, no. 3, pp. 195–200, 2011.

[66] L. Kuncheva, *Combining pattern classifiers: Methods and Algorithms*. Wiley, 2004.

[67] V. E. Kuz'min, P. G. Polishchuk, A. G. Artemenko, and S. A. Andronati, "Interpretation of QSAR models based on random forest methods," *Molecular Informatics*, vol. 30, no. 6-7, pp. 593–603, 2011.

[68] M. Lenzerini, "Data integration: a theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS '02. New York, NY, USA: ACM, 2002, pp. 233–246.

[69] Lhasa Limited. [Online]. Available: http://www.lhasalimited.org/ [Accessed: 22 January 2014].

[70] Lhasa Limited. Derek. [Online]. Available: http://www.lhasalimited.org/products/derek-nexus.htm [Accessed: 22 January 2014].

[71] R. Lowe, R. C. Glen, and J. B. O. Mitchell, "Predicting phospholipidosis using machine learning," *Molecular Pharmaceutics*, vol. 7, no. 5, pp. 1708–1714, 2010.

[72] G. M. Maggiora, "On outliers and activity cliffs: Why QSAR often disappoints," *Journal of Chemical Information and Modeling*, vol. 46, no. 4, pp. 1535–1535, 2006.

[73] M. Makhtar, D. Neagu, and M. Ridley, "Predictive model representation and comparison: Towards data and predictive models governance," in *Computational Intelligence (UKCI), 2010 UK Workshop on*, 2010, pp. 1–6.

[74] A. McNaught, "The IUPAC international chemical identifier : InChl - A new standard for molecular informatics," *Chemistry International*, vol. 28, no. 6, pp. 12–15, 2006.

[75] T. I. Netzeva, A. Worth, T. Aldenberg, R. Benigni, M. T. D. Cronin, P. Gramatica, J. S. Jaworska, S. Kahn, G. Klopman, C. A. Marchant, G. Myatt, N. Nikolova-Jeliazkova, G. Y. Patlewicz, R. Perkins, D. Roberts, T. Schultz, D. W. Stanton, J. van de Sandt, W. Tong, G. Veith, and C. Yang, "Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. the report and recommendations of ECVAM workshop 52." *ATLA. Alternatives to laboratory animals*, vol. 33, no. 2, pp. 155–73, 2005.

[76] N. L. Novere, A. Finney, M. Hucka, U. S. Bhalla, F. Campagne, J. Collado-Vides, E. J. Crampin, M. Halstead, E. Klipp, P. Mendes, P. Nielsen, H. Sauro,

B. Shapiro, J. L. Snoep, H. D. Spence, and B. L. Wanner, "Minimum information requested in the annotation of biochemical models (MIRIAM)," *Nature Biotechnology*, vol. 23, pp. 1509–1515, 2005.

[77] OECD. OECD principles for the validation, for regulatory purposes, QSAR models. [Online]. Available: http://www.oecd.org/dataoecd/33/37/37849783.pdf [Accessed: 22 January 2014].

[78] Online Chemical Modelling Environment. OCHEM. [Online]. Available: http://ochem.eu/ [Accessed: 22 January 2014].

[79] OpenQSAR. [Online]. Available: http://www.openqsar.com/index.jsp [Accessed: 10 November 2012].

[80] OpenTox. [Online]. Available: http://www.opentox.org [Accessed: 22 January 2014].

[81] A. Palczewska, J. Palczewski, R. Robinson, and D. Neagu, "Interpreting random forest models using a feature contribution method," in *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, 2013, pp. 112–119.

[82] A. Palczewska. `rfFC`. [Online]. Available: https://r-forge.r-project.org/R/?group_id=1725 [Accessed: 22 January 2014].

[83] A. Palczewska, X. Fu, P. Trundle, L. Yang, D. Neagu, M. Ridley, and K. Travis, "Towards model governance in predictive toxicology," *International Journal of Information Management*, vol. 33, no. 3, pp. 567–582, 2013.

[84] A. Palczewska, D. Neagu, and M. Ridley, "Using pareto points for model identification in predictive toxicology," *Journal of Cheminformatics*, vol. 5, no. 1, p. 16, 2013.

[85] A. Palczewska, J. Palczewski, R. Marchese Robinson, and D. Neagu, "Interpreting random forest classification models using a feature contribution method," in *Integration of Reusable Systems*, ser. Advances in Intelligent Systems and Computing, T. Bouabana-Tebibel and S. H. Rubin, Eds. Springer International Publishing, 2014, vol. 263, pp. 193–218.

[86] G. Patlewicz, N. Jeliazkova, A. Gallegos Saliner, and A. P. Worth, "Toxmatch-a new software tool to aid in the development and evaluation of chemically similar groups," *SAR and QSAR in Environmental Research*, vol. 19, no. 3, pp. 397–412, 2008.

[87] A. Prasad, L. Iverson, and A. Liaw, "Newer classification and regression tree techniques: Bagging and random forests for ecological prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.

[88] PubChem. [Online]. Available: http://pubchem.ncbi.nlm.nih.gov/ [Accessed: 22 January 2014].

[89] QSAR DATA BANK. [Online]. Available: http://qsardb.org/ [Accessed: 22 January 2014].

[90] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2012.

[91] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[92] rcdk. Interface to the CDK Libraries. [Online]. Available: http://cran.r-project.org/web/packages/rcdk/index.html [Accessed: 22 January 2014].

[93] Registration, Evaluation, Authorisation & Restriction of CHemicals . REACH ANNEX XI. [Online]. Available: http://www.reachonline.eu/REACH/EN/REACH_EN/articleXI.html [Accessed: 22 January 2014].

[94] Registration, Evaluation, Authorisation & Restriction of CHemicals. REACH. [Online]. Available: http://www.hse.gov.uk/reach/ [Accessed: 22 January 2014].

[95] L. Rosenbaum, G. Hinselmann, A. Jahn, and A. Zell, "Interpreting linear support vector machine models with heat map molecule coloring," *Journal of Cheminformatics*, vol. 3, no. 1, p. 11, 2011.

[96] A. Rusinko, M. W. Farmen, C. G. Lambert, P. L. Brown, and S. S. Young, "Analysis of a large structure/biological activity data set using recursive partitioning,"

Journal of Chemical Information and Computer Sciences, vol. 39, no. 6, pp. 1017–1026, 1999.

[97] L. Sachs, *Applied statistics: A handbook of techniques*. Springer-Verlag (New York N.Y.), 1982.

[98] K. Schneider, M. Schwarz, I. Burkholder, A. Kopp-Schneider, L. Edler, A. Kinsner-Ovaskainen, T. Hartung, and S. Hoffmann, "ToxRTool, a new tool to assess the reliability of toxicological data," *Toxicology Letters*, vol. 189, no. 2, pp. 138–144, 2009.

[99] T. W. Schultz, "Tetratox: Tetrahymena pyriformis population growth impairment endpointa surrogate for fish lethality," *Toxicology Methods*, vol. 7, no. 21, pp. 289–309, 1997.

[100] E. Serna M., "Maturity model of knowledge management in the interpretativist perspective," *International Journal of Information Management*, vol. 32, no. 4, pp. 365–371, 2012.

[101] G. Shuurmann, R. Ebert, J. Chen, B. Wang, and R. Kuhne, "External validation and prediction employing the predictive squared correlation coefficient - test set activity mean vs training set activity mean." *Journal of Chemical Information and Modeling*, vol. 48, pp. 2140–2145, 2008.

[102] O. Spjuth, E. Willighagen, R. Guha, M. Eklund, and J. Wikberg, "Towards interoperable and reproducible QSAR analyses: Exchange of datasets," *Journal of Cheminformatics*, vol. 2, no. 1, p. 5, 2010.

[103] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen, "The chemistry development kit (CDK): An open-source java library for chemo- and bioinformatics." *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 2, pp. 493–500, 2003.

[104] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[105] I. Sushko, S. Novotarskyi, R. Karner, A. Pandey, M. Rupp, W. Teetz, S. Brandmaier, A. Abdelaziz, V. Prokopenko, V. Tanchuk, R. Todeschini, A. Varnek, G. Marcou, P. Ertl, V. Potemkin, M. Grishina, J. Gasteiger, C. Schwab, I. Baskin, V. Palyulin, E. Radchenko, W. Welsh, V. Kholodovych, D. Chekmarev, A. Cherkasov, J. Aires-de Sousa, Q.-Y. Zhang, A. Bender, F. Nigsch, L. Patiny, A. Williams, V. Tkachenko, and I. Tetko, "Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information," *Journal of Computer Aided Molecular Design*, vol. 25, no. 6, pp. 533–554, 2011.

[106] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[107] E. Sweden. Data Governance – Managing Information As An Enterprise Asset Part I – An Introduction. [Online]. Available: http://www.nascio.org/publications/documents/NASCIO-DataGovernance-Part1.pdf [Accessed: 22 January 2014].

[108] E. Sweden. Data Governance Part II: Maturity Models A Path to Progress. [Online]. Available: http://www.nascio.org/publications/documents/NASCIO-DataGovernancePTII.pdf [Accessed: 22 January 2014].

[109] Talete. Dragon 6. [Online]. Available: http://www.talete.mi.it/products/dragon_description.htm [Accessed: 22 January 2014].

[110] R. V. Tappeta and J. E. Renaud, "Interactive multiobjective optimization procedure," *AIAA Journal*, vol. 37, no. 7, pp. 881–889, 1999.

[111] R. Todeschini, V. Consonni, and M. Pavan, "A distance measure between models: a tool for similarity/diversity analysis of model populations," *Chemometrics and Intelligent Laboratory Systems*, vol. 70, no. 1, pp. 55–61, 2004.

[112] R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors*. Wiley-VCH Verlag GmbH, 2008.

[113] A. Tropsha, "Best Practices for QSAR Model Development, Validation, and Exploitation," *Molecular Informatics*, vol. 29, no. 6-7, pp. 476–488, 2010.

[114] A. Tropsha, P. Gramatica, and V. K. Gombar, "The importance of being earnest: Validation is the absolute essential for successful application and interpretation of QSPR models." *QSAR and Combinatorial Science*, vol. 22, no. 1, pp. 69–77, 2003.

[115] UCI Machine Learning Repository. Breast Cancer Wisconsin Diagnostic Dataset. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29 [Accessed: 22 January 2014].

[116] UCI Machine Learning Repository. Iris Dataset. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Iris [Accessed: 22 January 2014].

[117] H. Verhaar, C. Leeuwen, and J. Hermens, "Classifying environmental pollutants," *Chemosphere*, vol. 25, no. 4, pp. 471–491, 1992.

[118] J. Walker, I. Gerner, E. Hulzebos, and K. Schlegel, "The skin irritation corrosion rules estimation tool (SICRET)." *QSAR and Combinatorial Science*, vol. 24, no. 3, pp. 378–384, 2005.

[119] M. Waters and J. Fostel, "Toxicogenomics and systems toxicology: aims and prospects," *Nature Review Genetics*, vol. 5, no. 12, pp. 936–948, 2004.

[120] P. Weill and J. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Boston, MA, USA: Harvard Business School Press, 2004.

[121] P. Willet, J. M. Berdnard, and G. M. Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 983–996, 1998.

[122] A. Wojak, D. Neagu, and M. Ridley, "Double min-score (DMS) algorithm for automated model selection in predictive toxicology," in *United Kingdom Workshop in Computational Intelligence (UKCI 2011)*, 2011, pp. 150–156.

[123] S. Wold, M. Sjstrm, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, 2001.

[124] D. J. Wood, D. Buttar, J. G. Cumming, A. M. Davis, U. Norinder, and S. L. Rodgers, "Automated QSAR with a hierarchy of global and local models," *Molecular Informatics*, vol. 30, no. 11-12, pp. 960–972, 2011.

[125] Y. Xue, H. Li, C. Y. Ung, C. W. Yap, and Y. Z. Chen, "Classification of a diverse set of tetrahymena pyriformis toxicity chemical compounds from molecular descriptors by statistical learning methods," *Chemical Research in Toxicology*, vol. 19, no. 8, pp. 1030–1039, 2006.

[126] L. Yang, D. Neagu, M. T. Cronin, M. Hewitt, S. Enoch, J. C. Madden, and K. Przybylak, "Towards a fuzzy expert system on toxicological data quality assessment," *Molecular Informatics*, vol. 32, no. 1, pp. 65–78, 2013.

[127] X.-Q. Zeng, G.-Z. Li, J. Yang, and M. Yang, "A novel metric for redundant gene elimination based on discriminative contribution," in *Bioinformatics Research and Applications*, ser. Lecture Notes in Computer Science, I. Mndoiu, R. Sunderraman, and A. Zelikovsky, Eds. Springer Berlin Heidelberg, 2008, vol. 4983, pp. 256–267.

[128] H. Zhu, A. Tropsha, D. Fourches, A. Varnek, E. Papa, P. Gramatical, T. Öberg, P. Dao, A. Cherkasov, and I. Tetko, "Combinatorial QSAR modeling of chemical toxicants tested against tetrahymena pyriformis." *Journal of Chemical Information and Modeling*, vol. 48, no. 4, pp. 766–784, 2008.

Appendix A

rfFC-package

Package 'rfFC'

October 11, 2013

Type Package

Title randomForest Feature Contributions

Version 1.0

Date 2013-01-24

Author Anna Palczewska <annawojak@gmail.com>, Richard Marchese Robinson <rmarcheserobinson@gmail.com>

Maintainer Anna Palczewska <annawojak@gmail.com>

Depends R (>= 2.13), randomForest

Description Random Forest Feature Contribution

License GPL (version 2 or later)

LazyLoad yes

R topics documented:

ames *Hansen Ames Mutagenicity Dataset*

Description

This binary classification dataset corresponds to the benchmark dataset for Ames mutagenicity modelling presented by Hansen *et al.*

Usage

```
data(ames)
```

Format

ames is a data frame with 6512 cases (rows) and 170 variables (columns). This data set consists of two types of Activity: (1) positive, (0) negative chemical compounds. All chemical structures are encoded using binary attributes: a bit vector calculated based upon the MACCS key fingerprint. Dataset structure: CAS_NO, Activity, Canonical_Smiles, X1-X166 - 166 binary descriptors, Type. The CAS_NO column presents the CAS number (i.e. the instance ID). The 'Training' and 'Test' labels in the Type column denote the first of the five splits presented by Hansen *et al.*

References

K. Hansen, S. Mika, T. Schroeter, A. Sutter, A. ter Laak, T. Steger-Hartmann, N. Heinrich, K.-R. Mueller (2009),Benchmark Data Set for in Silico Prediction of Ames Mutagenicity. *Journal of Chemical Information and Modeling*, 49, 2077-2081.

checkForestUnanimity *Check the unanimity of all trees in the Random Forest model*

Description

This method checks the **unanimity** of all individual trees in the forest for classification models: this denotes the condition that, for any given leaf (i.e. terminal) node of the tree, all instances in the training set assigned to that node should belong to a single class. If this holds for a single tree, the tree is considered **unanimous**. Only if this condition -i.e. that all trees are **unanimous** - holds will the predictions obtained (for "class 1") for a binary classification model using predict(...,type="prob") and predictBC(...) be the same.

Usage

```
checkForestUnanimity(object, dataT)
```

Arguments

object	an object of the class randomForest
dataT	a data frame with columns containing the attributes (descriptors) for all instances (rows) in the training set of the randomForest object

Value

A list with the following components:

dec	TRUE if all trees in the forest are **unanimous**, otherwise FALSE
tcCount	a list providing the number of training set instances in each class for each terminal node in all trees. Where the number 0 is presented for all classes, the corresponding node is not a terminal node.
tuStatus	a vector, with one element per tree, denoting whether or not that tree was **unanimous** (TRUE) or not (FALSE)

Author(s)

Anna Palczewska <annawojak@gmail.com>

See Also

randomForest

Examples

```
#Iris dataset
library(randomForest)
data(iris)
rF_Model <- randomForest(x=iris[,-5],y=as.factor(as.character(iris[,5])),
                              ntree=10,importance=TRUE, keep.inbag=TRUE,replace=FALSE)

#Check unanimity
itest<-checkForestUnanimity(rF_Model, iris[,-5])



## Not run:
# Ames dataset
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
                               ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
itest<-checkForestUnanimity(rF_Model, ames_train[,-1])

## End(Not run)
```

`featureContributions` *Feature Contributions for a Random Forest Model*

Description

This method calculates feature contributions for a given dataset and an existing Random Forest model `randomForest`. The feature contributions are computed separately for each instance/record in dataset and provide detailed information about relationships between variables and the predicted value. This method was implemented based upon the approach of Kuz'min *et al.* for regression models and extended to classification models. For a binary classification model the method returns the feature contributions towards class "one". For a multi-class model, the feature contributions are calculated towards the class predicted by the `randomForest` model for a given instance.

The method does not work for unsupervised models. The `randomForest` model must have a stored in-bag matrix that keeps track of which samples were used to build trees in the forest and sampling without replacement must be used to generate a model.

Hence, all Random Forest models analyzed by this method must be generated as follows:

```
model <- randomForest(...,keep.inbag=TRUE,replace=FALSE)
```

The reason for this current limitation is because, in the code of the `randomForest` implementation of Random Forest provided by Liaw and Wiener, the `inbag` matrix does not record how many times a sample was used to build a particular tree (if sampling with replacement).

Usage

```
featureContributions(object, lInc, dataT, mClass=NULL)
```

Arguments

`object`	an object of the class `randomForest`
`lInc`	local increments of feature contributions calculated for this object using `getLocalIncrements`
`dataT`	a data frame containing the variables in the model (columns) for all instances (rows) for which feature contributions are desired
`mClass`	a name of the class to which feature contributions is calculated. The class name must to match to the one class name from the randomForest object variable `y`. By default, the value of this parameter is set to `NULL`. In this case the feature contributions are calculated to the predicted class returned by the `predict` method from the `randomForest` package. This option is avaliable only for the multi-classification problems.

Value

A list with the following components:

contrib — n x m matrix of feature contributions, where n is the number of records (i.e. instances) and m is the number of features/variables (i.e. attributes/descriptors) of the dataset dataT.
For regression, or multi-class classification, the feature contributions represent the signed contributions towards the predicted value or a given class defined by the argument mClass.
For binary classification, by default, the classes are internally treated as numeric 1 and 0 and the feature contributions represent the signed contributions towards "class 1". If the class labels in the training set are presented as "1" and "0", then the corresponding classes will be internally treated as 1 and 0 respectively; otherwise, this mapping will be performed arbitrarily with the class of the first instance in the training set treated internally as "class 1", or to the class provided as a parameter in getLocalIncrements.

Author(s)

Anna Palczewska <annawojak@gmail.com> and
Richard Marchese Robinson <rmarcheserobinson@gmail.com>

References

V.E. Kuz'min et al. (2011), Interpretation of QSAR Models Based on Random Forest Methods, *Molecular Informatics*, 30, 593-603.
A. Palczewska et al. (2013), Interpreting random forest models using a feature contribution method, *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration IEEE IRI 2013*, August 14-16, 2013, San Francisco, California, USA, 112-119.

See Also

randomForest, getLocalIncrements

Examples

```
#Multi-class Classification
library(randomForest)
data(iris)
rF <- randomForest(x=iris[,-5],y=as.factor(as.character(iris[,5])),
    ntree=25,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
#Get Local feature incremets
li<-getLocalIncrements(rF, iris[,-5])
#Calculate feature contributions
fc<-featureContributions(rF, li, iris[,-5])

## Not run:
#Binary classification
library(randomForest)
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
          ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
```

```
li <- getLocalIncrements(rF_Model,ames_train[,-1])
fc<-featureContributions(rF_Model, li, ames_train[,-1])

## End(Not run)
```

`getChanges` *Get change in prediction for an updated feature.*

Description

This method calculates the changes in predictions, for a pre-determined Random Forest model and set of instances, resulting from updating the value(s) of a specified (vector of) feature(s). The method works with regression and classification models. In case of binary classification the predictions are calculated by `predictBC()` or `predict.randomForest` and represent the probabilities of being in a given class. If the model was obtained directly via the `randomForest()` function, the type of predictions calculated correspond to `predict.randomForest()`. However, if the model is a binary classification model and was obtained via post-processing the original model from `randomForest()`, using `prepareForPredictBC()`, the type of predictions calculated correspond to `predictBC()`.

Usage

```
getChanges(features, dataT, object, value=NULL, type=NULL, mcls=NULL)
```

Arguments

`features`	a vector of the feature numbers/names to be updated
`dataT`	a data frame containing the variables in the model for all instances for which changes in predictions are desired
`object`	an object of the class `randomForest`
`value`	a vector of new feature values for the features provided in `features` N.B. If this is set to NULL, and the specified features are binary, the prediction changes reported are those associated with the only possible change in value for these features: from 1 to 0 or vice-versa.
`type`	the type of the predictions considered for classification models, by default it is set to `type="prob"` but can be set to `type="votes"`.
`mcls`	main class that be set to "1" for binary classification. If `NULL`, the class name from the first record in `dataT` will be set as "1"

Value

A matrix `n x m` of prediction changes, `n` is the number of instances in `dataT` and `m` is the number of updated features.

Author(s)

Anna Palczewska <annawojak@gmail.com> and
Richard Marchese Robinson <rmarcheserobinson@gmail.com>

See Also

randomForest

Examples

```
library(randomForest)
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
ames_train<-ames_train[1:100,]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
  ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
gc <- getChanges(c(1,166), ames_train, rF_Model)
change<-getChanges(c(1), ames_train[1, ], rF_Model, value = c(0.49))
```

---

getLocalIncrements *Get Local Increments of Feature Contributions for a Random Forest Model*

Description

This method calculates local increments of feature contributions from an existing randomForest model. This method was implemented based upon the approach of Kuz'min et al. for regression models and extended to classification models. The method does not work for unsupervised models. The randomForest model must have a stored in-bag matrix that keeps track of which samples were used to build trees in the forest and sampling without replacement must be used to generate a model. Hence, all Random Forest models analyzed by getLocalIncrements() and, subsequently, featureContributions(), must be generated as follows: model <- randomForest(...,keep.inbag=TRUE,replace=FALSE) The reason for this current limitation is because, in the code of the randomForest implementation of Random Forest provided by Liaw and Wiener, the inbag matrix does not record how many times a sample was used to build a particular tree (if sampling with replacement). The method returns local increments for all nodes in each tree for regression and binary classification models. In case of multi-classification problems the method returns the local increments calculated for all classes for every tree node in the forest.

Usage

```
getLocalIncrements(object,  dataT, binAsReg=TRUE, mcls=NULL)
```

Arguments

object	an object of the class randomForest
dataT	a data frame containing the variables in the model for all instances for which feature contributions are desired
binAsReg	this option is only relevant for binary classification. If TRUE (default), the binary classification model is treated like a regression model,for the purpose of calculating feature contributions, with the class labels treated as numeric values of 1 or 0. If FALSE, only the local increments in favour of the predicted class (for the forest as a whole) are calculated - as per the treatment of multi-class classifiers.
mcls	main class that be set to "1" for binary classification. If NULL, the class name from the first record in dataT will be set as "1", otherwhise the provided class will be map to "1".

Value

A list with the following components:

type	the type of the method used for calculating local increments of feature contributions
forest	If a multi-class classification model, or a binary classification model analyzed using the binAsReg=FALSE option, has been analyzed, this is a list that contains: a vector lIncrements of local increments for all classes and each node of each tree, and a k x ntree matrix rmv of the mean proportion of instances in each class in the root nodes, where k is the number of classes and ntree is the number of trees in the forest. If a regression model,or a binary classification model analyzed using the binAsReg=TRUE option, has been analyzed, this is this is a list that contains: a vector lIncrements of local increments for all classes and each node of each tree, and another vector, of length ntree, rmv of the mean activity (with the two classes treated as numeric values of 1 or 0 in the case of binary classification) of instances in the root nodes.

Author(s)

Anna Palczewska <annawojak@gmail.com> and
Richard Marchese Robinson <rmarcheserobinson@gmail.com>

References

V.E. Kuz'min et al. (2011). Interpretation of QSAR Models Based on Random Forest Methods, *Molecular Informatics*, 30, 593-603.
A. Palczewska et al. (2013), Interpreting random forest models using a feature contribution method, *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration IEEE IRI 2013*, August 14-16, 2013, San Francisco, California, USA, 112-119.

See Also

randomForest

Examples

```
 ## Not run:
#Binary classification
library(randomForest)
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
        ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
li <- getLocalIncrements(rF_Model,ames_train[,-1])

## End(Not run)
```

`predictBC` *Makes predictions for a binary classification Random Forest model by averaging "probabilities".*

Description

This method makes predictions for a binary classification Random Forest model by computing the arithmetic mean of the "probability" generated by each tree, across all trees in the forest, that the instance being predicted will belong to the "selected" class. For a single tree, the probability is calculated as the proportion of local training set instances assigned to the terminal node in question which belong to the "selected" class. The class of the first instance in the complete training dataset is chosen as the "selected" class. This function will only work when applied to a `randomForest` object modified by `prepareForPredictBC`.

Usage

```
predictBC(object, dataT)
```

Arguments

`object`	an object of class `randomForest`
`dataT`	a data frame containing the variables in the model for the instances for which predictions are desired

Value

A vector of predictions for instances from the `dataT` dataset. The predicted values represent the estimated probability that the instance is in the "selected" class (the class of the the first instance in `dataT`).

Author(s)

Anna Palczewska <annawojak@gmail.com>

See Also

randomForest, prepareForPredictBC

Examples

```
## Not run:
library(randomForest)
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
   ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)

new_Model<-prepareForPredictBC(rF_Model, ames_train[,-1])
predicted<-predictBC(new_Model, ames_train[,-1])

## End(Not run)
```

`prepareForPredictBC` *Convert node predictions into probabilities for binary classification models.*

Description

This method can only be aplied for a binary classification model. Its primary purpose is to process a `randomForest` object as required for `predictBC()`. This method converts node predictions in the `randomForest` object. The current class label in terminal nodes is replaced by the probability of belonging to a "selected" class - where the probability is calculated as the proportion of local training set instances assigned to the terminal node in question which belong to the "selected" class. The class of the first instance in the complete training dataset is chosen as the "selected" class.

Usage

```
prepareForPredictBC(object,  dataT, mcls=NULL)
```

Arguments

`object`	an object of the class `randomForest`
`dataT`	a data frame containing the variables in the model for all instances in the training set
`mcls`	main class that be set to "1" for binary classification. If `NULL`, the class name from the first record in `dataT` will be set as "1"

Value

an object of class `randomForest` with a new `type="binary"`.

Author(s)

Anna Palczewska <annawojak@gmail.com>

See Also

randomForest

Examples

```
## Not run:
library(randomForest)
data(ames)
ames_train<-ames[ames$Type=="Train",-c(1,3, ncol(ames))]
rF_Model <- randomForest(x=ames_train[,-1],y=as.factor(as.character(ames_train[,1])),
ntree=500,importance=TRUE, keep.inbag=TRUE,replace=FALSE)
new_Model<-prepareForPredictBC(rF_Model, ames_train[,-1])

## End(Not run)
```

Index