Open Source Workflow Engine for Cheminformatics: From Data Curation to Data Analysis

I n a u g u r a l - D i s s e r t a t i o n

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Thomas Kuhn

aus Herten

Köln, 2009

Berichterstatter: Prof. Dr. D. Schomburg
Prof. Dr. T. Wiehe

Tag der mündlichen Prüfung: 12.02.2009

Abstract

The recent release of large open access chemistry databases into the public domain generates a demand for flexible tools to process them so as to discover new knowledge. To support *Open Drug Discovery* and *Open Notebook Science* on top of these data resources, is it desirable for the processing tools to be Open Source and available to everyone.

The aim of this project was the development of an Open Source workflow engine to solve crucial cheminformatics problems. As a consequence, the *CDK-Taverna* project developed in the course of this thesis builds a cheminformatics workflow solution through the combination of different Open Source projects such as Taverna (workflow engine), the *Chemistry Development Kit* (CDK, cheminformatics library) and *Pgchem::Tigress* (chemistry database cartridge). The work on this project includes the implementation of over 160 different workers, which focus on cheminformatics tasks. The application of the developed methods to real world problems was the final objective of the project.

The validation of Open Source software libraries and of chemical data derived from different databases is mandatory to all cheminformatics workflows. Methods to detect the atom types of chemical structures were used to validate the atom typing of the *Chemistry Development Kit* and to identify curation problems while processing different public databases, including the EBI drug databases *ChEBI* and *ChEMBL* as well as the natural products *Chapman & Hall Chemical Database*. The CDK atom typing shows a lack on atom types of heavier atoms but fits the need of databases containing organic substances including natural products.

To support combinatorial chemistry an implementation of a reaction enumeration workflow was realized. It is based on generic reactions with lists of reactants and allows the generation of chemical libraries up to O(1000) molecules.

Supervised machine learning techniques (perceptron-type artificial neural networks and support vector machines) were used as a proof of concept for quantitative modelling of adhesive polymer kinetics with the *Mathematica GNWI.CIP* package. This opens the

perspective of an integration of high-level “experimental mathematics” into the *CDK-Taverna* based scientific pipelining.

A chemical diversity analysis based on two different public and one proprietary databases including over 200,000 molecules was a large-scale application of the methods developed. For the chemical diversity analysis different molecular properties are calculated using the *Chemistry Development Kit*. The analysis of these properties was performed with *Adaptive-Resonance-Theory* (ART 2-A algorithm) for an automatic unsupervised classification of open categorical problems. The result shows a similar coverage of the chemical space of the two databases containing natural products (one public, one proprietary) whereas the *ChEBI* database covers a distinctly different chemical space. As a consequence these comparisons reveal interesting white-spots in the proprietary database. The combination of these results with pharmacological annotations of the molecules leads to further research and modelling activities.

Zusammenfassung

In jüngerer Zeit führt die Veröffentlichung von lizenzfreien Open-Access Chemiedatenbanken zu einer erhöhten Nachfrage für flexible Tools zur Verarbeitung der Daten und zur Gewinnung neuen Wissens. Zur Unterstützung der *Open Drug Discovery* und der *Open Notebook Science* ist es erstrebenswert, dass zusätzlich zu den Daten, auch die Anwendungen zur Bearbeitung der Daten, Open-Source und somit für jedermann verfügbar sind.

Ziel dieser Arbeit war die Entwicklung einer Open-Source Workflow Lösung zur Bearbeitung von Chemoinformatik-Problemen. Das *CDK-Taverna-Projekt* erstellt eine Chemoinformatik-Workflow-Lösung durch die Kombination verschiedener Open-Source-Projekte wie z.B. *Taverna* (Workflowumgebung), das *Chemistry Development Kit* (CDK, Chemoinformatik-Bibliothek) oder *Pgchem::Tigress* (chemische Datenbankerweiterung). Während der Arbeit an diesem Projekt wurden mehr als 160 verschiedene Prozessoren zur Bearbeitung von Chemoinformtik-Problemen implementiert. Neben der Implementierung von verschiedenen Prozessoren stand die Anwendung der entwickelten Methoden auf reale Probleme im Zentrum dieser Arbeit.

Die Validierung von Softwarebibliotheken sowie von Daten verschiedener Datenbanken ist obligatorisch für jede Chemoinformatik-Workflow-Lösung. Die Validierung der Methoden zur Identifizierung von Atomtypen des *Chemistry Development Kits* erfolgte während der Verarbeitung von Datensätzen verschiedener Datenbanken zur Erkennung von Kurierungsproblemen. Folgende Datenbanken wurden eingesetzt: die pharmakologischen EBI Datenbanken *ChEBI* und *ChEMBL* und die Naturstoff *Chapman & Hall Chemical Datenbank*. Die Validierung zeigte, dass es dem CDK an Atomtypen schwerer Atome mangelt, sich jedoch sehr gut für Naturstoffe sowie organischer Moleküle eignet.

Zur Unterstützung der kombinatorischen Chemie wurde eine Implementierung eines Reaktionsenumerator-Workflows umgesetzt. Dieser basiert auf generischen Reaktionen sowie Eduktlisten und ermöglicht die Erstellung von chemischen Bibliotheken mit bis zu O(1000) Molekülen.

Methoden des nicht überwachten Maschinen Lernens (*Perceptron-Type Artificial Neural Network* und *Support Vector Machines*) wurden als Proof-of-Concept für die quantitative Modellierung der Kinetik von Klebstoffpolymer mittels der *Mathematica GNWI.CIP* Erweiterung eingesetzt. Diese Perspektive ermöglicht die Integration von „experimenteller Mathematik“ in die auf *CDK-Taverna* basierende wissenschaftliche Workflow-Lösung.

Eine chemische Diversitätsanalyse basierend auf den Daten zweier öffentlicher und einer proprietären Datenbank mit zusammen mehr als 200000 Molekülen stellte eine weitere Anwendung der entwickelten Methoden dar, die innerhalb dieser Arbeit erledigt wurden. Für die chemische Diversitätsanalyse wurden verschiedene molekulare Eigenschaften unter Nutzung des *Chemistry Development Kits* berechnet. Die Analyse der Eigenschaften erfolgte mittels einer Implementierung eines Algorithmus der *Adaptiven Resonanztheorie* (ART 2-A) zur automatischen nicht überwachten Klassifizierung von offen kategorischen Problemen. Die Analyse zeigte eine ähnliche Abdeckung des chemischen Raums der beiden Naturstoff-Datenbanken. Einzig die *ChEBI* Datenbank deckte einen anderen chemischen Raum ab. Die Diversitätsanalyse beinhaltete auch die Suche nach White-Spots einzelner Datenbanken. Als Ergebnis dieser Vergleiche wurden interessante White-Spots innerhalb der proprietären Datenbank entdeckt. Die Kombination aus diesen Ergebnissen mit der pharmakologischen Annotation einzelner Moleküle führt zu weiteren Forschungs- und Modellierungaktivitäten.

Danksagung

Ich möchte mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben, besonders auch denen, die ich hier nicht erwähne.

Mein Dank gilt Prof. Dr. Dietmar Schomburg für die Übernahme der Funktion des Erstprüfers, sowie Prof. Dr. Thomas Wiehe für die Übernahme des Zweitgutachtens.

Prof. Dr. Axel Klein und PD Dr. Karsten Niefind danke ich für die Übernahme des Vorsitzes bzw. Beisitzes bei meiner Disputation.

Mein besonderer Dank gilt PD Dr. Christoph Steinbeck für das Ermöglichen dieser Arbeit mit diesem interessanten Thema, sowie die Betreuung der Arbeit.

Prof. Dr. Achim Zielesny möchte ich besonders danken für die fachliche und seelische Unterstützung, die Diskussionen sowie die Hilfe während der ganzen Arbeit. Sein Engagement half sehr bei der Erstellung und Fertigstellung dieser Arbeit.

Stefan Neumann und dem gesamten GNWI Team danke ich für Ihre geistige und materielle Unterstützung. Die tiefgreifenden Diskussionen und Anregungen die Sie eingebracht haben, trugen zum erfolgreichen Gelingen bei. Auch die aufbauenden und fachfremden Gespräche haben meine Leistungen und meinen Willen gestärkt.

Dem ehemaligen CUBIC Team möchte ich für die freundliche Aufnahme, die regen Diskussionen sowie für die Unterstützung bei meiner Arbeit danken.

InterMed Discovery, insbesondere Dr. Matthias Gehling danke ich für die Überlassung der Daten sowie das entgegengebrachte Vertrauen.

Last but not least, möchte ich meiner Frau Maren sowie meiner Familie für die stetige Unterstützung in guten aber auch in schlechten Zeiten danken. Ohne euch wäre eine solche Arbeit nicht möglich gewesen. Danke.

List of Abbreviations

AN	Artificial Neuron
ANN	Artificial Neural Network
API	Application Programming Interface
ART	Adaptive Resonance Theory
CDK	Chemistry Development Kit
ChEBI	Chemical Entry of Biological Interest
CML	Chemical Markup Language
CRM	Customer Relationship Management
EBI	European Bioinformatics Institute
GiST	Generalized Search Tree
HTS	High Throughput Screening
InChI	IUPAC International Chemical Identifier
IUPAC	International Union of Pure and Applied Chemistry
MCSS	Maximum Common Substructure Searches
MMV	Medicine for Malaria Venture
NaN	Not a Number
NMR	Nuclear Magnetic Resonance
NN	Neural Networks
OASIS	Organization for the Advancement of Structured Information Standards
OSI	Open Source Initiative
PPPs	Public-private partnerships
QSAR	Quantitative Structure-Activity Relationships
QSPR	Quantitative Structure-Property Relationships
R&D	Research and Development
RCP	Rich Client Platform
SAR	Set of All Rings
Scufl	Simple Conceptual Unified Flow Language
SDK	Software Development Kit
SMILES	Simplified Molecular Line Entry Specification
SOA	Service Oriented Architecture
SOMs	Self-Organizing Feature Map
SPI	Service Provider Interfaces
SQL	Structure Query Language
SSSR	Smallest Set of Smallest Rings
tsv	Tabular Separated Values

Index of Table

List of Figures

Table of Contents

1 Introduction

The recent release of large open Chemistry databases into the public domain such as *PubChem*(1), *ZINC*(2), *ChEBI*(3) and *ChemDB*(4) calls for flexible, open toolkits to process them. These databases and tools will, for the first time, create opportunities for academia and third-world countries to perform state-of-the-art open drug discovery and transnational research - endeavours so far a sole domain of the pharmaceutical industry. In order to facilitate this process, the tools for processing these datasets have to be free and open. Within *Open Drug Discovery* and *Open Notebook Science* (5) scientists can participate in the research and get unrestricted access to what has been learned (6).

The delayed development of public cheminformatics tool kits, compared with bioinformatics, was caused by a dearth of publicly available chemistry data. The Cheminformatics tool kits handle collections of small molecules with at most a few dozen atoms which are usable for combinatorial chemistry as building blocks (7), in polymer chemistry as monomers, as molecular probes for analysing biological systems in chemical genomics and system biology and of course in drug discovery for the screening, design and discovery of new compounds. This wide range of applications justifies the fundamental requirement of public chemiformatics tool kits.

Figure 1.1 shows two possible starting points for the design of new drugs. One starting point is a protein with a known structure and perhaps a corresponding ligand. With public molecule databases available, an in silico docking experiment with millions of molecules could bring up new lead structures. With sufficient computer power, a docking of all known small molecules against all proteins from the *PDB* (8) is thinkable. On the other hand starting from the ligand, a search for similar molecules on the databases could detect known similar compounds. These compounds could yield to a better biocompatible ligand and thus to a new lead structure.

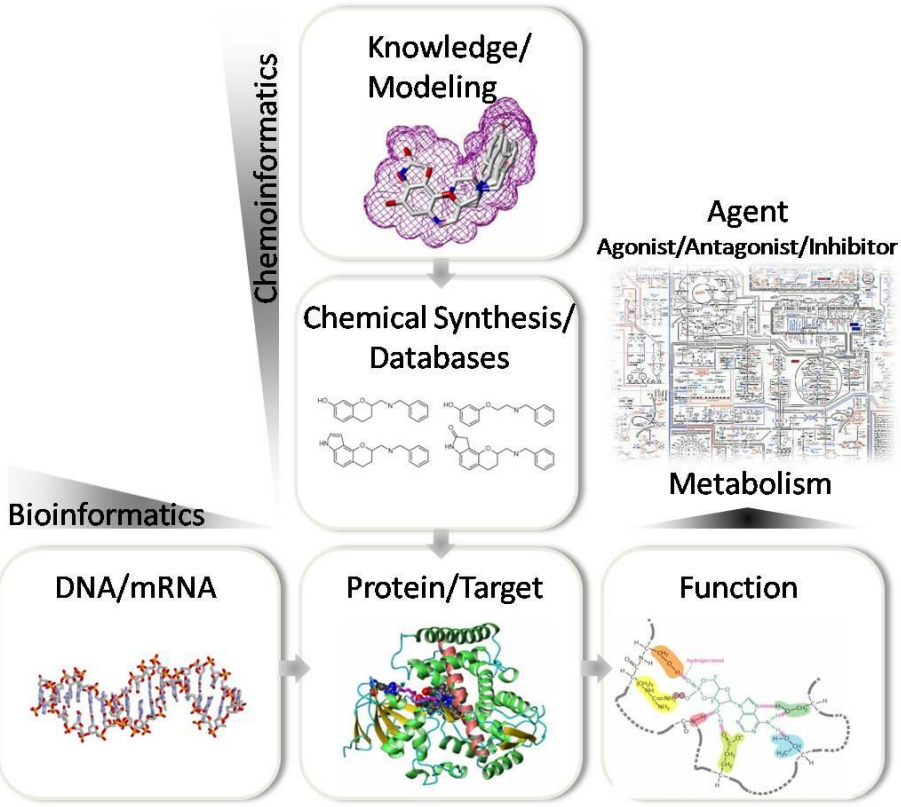


Figure 1.1: Methods of detecting pharmacological targets and the disciplines involved

The availability of large chemical datasets has generated a demand for publicly available tools to handle and process these data. There are tools needed to create subsets using substructure searches or chemical similarity. At this point different definitions of similarity, such as the *Tanimoto* coefficient or the *Distance Moment* (9) are usable. These subsets create the necessity for flexible local storage systems, which can be a single file or folder with molecular structures stored in a specific format or a database. Such a database should preferably provide chemical functionality such as substructure search or the ability to count the number of atoms in each molecule. To support this functionality a database needs a cheminformatics cartridge.

Another well-known problem with chemical information systems is the format used for the representation of stored molecules. Nowadays there are many available and differing formats for storing molecular information which makes an error-prone conversion between formats nearly always mandatory. Some newer and open formats are the *Chemical Markup Language* (CML) (10) or the *IUPAC International Chemical*

Identifier (InChI). Both are attempting to replace existing proprietary formats in the long run.

The level of the chemical information stored in a molecule description is another problem. Many datasets contain molecules with disallowed atom configurations such as five-bonded nitrogen (see Figure 1.2). These types of disallowed atom configurations can lead to the calculation of incorrect molecular properties. A scientist handling large a number of molecules from different databases have to check his molecules to avoid this kind of problems. After the identification of the problematic molecules, follows the exclusion of the molecules as the first and easiest possibility or the replacement of the disallowed atom types by allowed ones in a curation process.

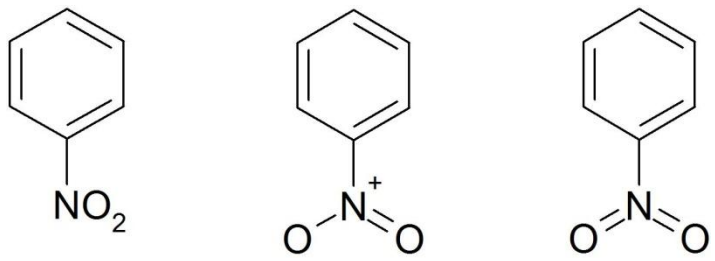


Figure 1.2: The first two representations of Nitrobenzene are chemically correct; whereas Nitrobenzene with five bonded nitrogen is unreasonable.

The availability of a very large number of chemical datasets has lead to an increased demand for tools to process these data automatically or semi-automatically. These tools should be flexible, extensible and usable for scientists with little or no specific programming knowledge.

1.1 The Workflow Paradigm

The term “Workflow” can be used to describe different things in different contexts. The abstraction of real work segregated into the divisions of work, work sub-divisions and with specific types of ordering describes a workflow. In 1996 the *Workflow Management Coalition* defined a workflow as: “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” (11)

Leyman and Roller (12) describes four different types for workflows in business. Most of them have direct counterparts in science and engineering. A **collaborative workflow** is commonly met with a large project which involves many individuals and which has a high business value for a company: This includes the production, documentation and release of major products. A scientific counterpart would be the management of data produced and distributed due to the undertaking of a large experiment. The second type of workflow they describe is "**ad hoc**". This less formal activity can be a notification process, which informs about a changed business practice or policy. Notification driven workflows are common in science. A good example is an agent process that looks for new publications with specific keywords. The third workflow type is **administrative** and refers to enterprise activities such as database management and maintenance scheduling. These frequently used workflows do not belong to the core business of a company. The last type of workflow referred to is **production workflow**. This workflow type contains processes directly concerned with the core business of a company, such as the setting up of loans and their processing by a bank. These administrative and the productive workflows have a counterpart in science. The managing of data originating from a measuring instrument could be seen as an example of an administrative workflow. Alternatively, the daily data analysis of new measured properties would belong to the production workflow category in science. (13)

The daily work of a chemist involves number of different workflows. For example, an organic chemist who synthesises a new compound and analyses the success of the reaction involved with an NMR spectrometer usually uses an electronic lab journal for the documentation of the experiment. The same chemist also uses a range of other tools to handle his daily work. He uses specific tools to calculate chemical properties or tools for the statistical analysis of his results. All these steps are stages in a specific workflow and therefore occur in a specific sequence.

The advantage of a workflow environment is the creation of workflows for the scientist in a Lego^{TM} like manner. He can combine minimum sized different units or individual workers to get a workflow that satisfy current requirements. The workflow, once created, is a source of suitable documentation and becomes the basis for future work. A major advantage of a workflow system is the reduction of error possibilities due to user interactions - such as the incorrect copying of data between different tools or the

extraction of wrong or inappropriate content from specific results. A once-created workflow is executable in the same environment for multiple iterations. The metaphor drawn here is that “the whole is greater than the sum of the parts”.

1.2 Open Source - Open Tools

Open Source is an overall shared development methodology (14) that offers a practical mode of access to software itself, and all background information regarding it, including source code. According to the *Open Source Initiative* (OSI) Open Source does not just mean access to the source code. The terms of distribution of Open Source software must fulfil different criteria: (15)

- Free redistribution
- Program has to include the source code
- Modification and derived works must be allowed
- Integrity of the author’s source code
- No discrimination against persons or groups
- No discrimination against field or endeavour
- Distribution of the license
- The license must not be specific to a product
- License must not restrict other software
- License must be technology-neutral

Historically all software was open before the commercialization of software started in the 1980s apart from in the main frame world. The distribution of software systems at this time happened directly with the hardware and was often freely exchange in user forums. As a reaction to the commercialization of Software, *Richard Stallman* founded the *Free Software Foundation* 1985 to support the free software movement.

One of the co-founders of OSI *Eric S. Raymond* summarized firstly in 1997, his views on the advantages of Open Source software development in his famous essay “The Cathedral and the Bazaar” with its central thesis “given enough eyeballs, all bugs are shallow”. (16)

Raymond compared two different software development models, the “Cathedral” and the “Bazaar” with each other. The “Cathedral” model often used for proprietary software development (but also used in Open Source projects) is a synonym for centralized development. Here, development between releases is restricted to an exclusive group of developers. Long development cycles and a vertical management are the characteristics for this development model. Probably the most famous example for the “Bazaar” model is the development of Linux. There, the development and the distribution occurs over the Internet and is visible to the public. A collaborative development approach with short development cycles and a democratic management are the foundations of this model.

The fundamentals of science are similar to the idea behind Open Source. Both propagate the improving of knowledge through producing, sharing and validating information. The ongoing Open Access discussion, for example, leads to larger amount of freely accessible scientific publications and data becoming available. Over the last few years, the foundation of a number of non-profit organizations for supporting Open Source scientific software has lead to a better interoperability between different projects. Examples include the *Blue Obelisk Movement* (6) and the *OpenScience Project* (17).

A sign for the increased importance of the Open Source movement is the first advertisement for a W2 professorship in the field of Open Source software in Germany by the University of Erlangen-Nürnberg (18).

1.3 Open Drug Discovery

In the recent years, the low number of novel therapeutics approved by the industrialised nation’s medical institutions or boards such as the US FDA has caused great concern about the productivity levels and apparently declining innovation levels of the pharmaceutical industry. (19) Reasons cited for this ineffectiveness are the hierarchical organisational structures of vertical integrated companies with their rigid and inflexible research units. (20) In addition, there is the danger of innovation stagnating through the rising transaction costs of the bringing together of current research results.

The expense of pharmaceutical research and the low number of novel therapeutics has lead to average costs of over US$ 1 billion (21)(22) for the development of a successful new drug, which has to include the cost of failed drug developments. This immense investment forces the pharmaceutical companies to focus their research and development (R&D) on diseases prevalent in the comparatively wealthy industrial nations.

The idea of adopting the Open Source concept for pharmaceutical R&D has occurred while biology is becoming more and more an information based science. The first contact with Open Source within the drug development was in the field of bioinformatics with the use of Open Source tools and databases. The differences between the software and concepts in biology and those in chemistry have limited its application on drug development up to now.

In the last couple of years, organizations called public-private partnerships (PPPs) have adapted the Open Source model for drug development. The PPPs created new, low-cost business models through a combination of Open Source and outsourcing. Thus, these organizations can tackle challenges that the blockbuster business model of large pharmaceutical companies cannot address and which might be a way to address market niches. (23)

Existing public-private partnerships such as the *Medicines for Malaria Venture* (MVV) (24) address many of the practical questions involved in Open Source drug research. These types of business operate as virtual pharmaceutical company, which allow everyone to contribute to their projects. An expert committee of the MVV, for example, reviews every proposal and assigns the fundable proposals to a company project leader. Different facilities such as universities, pharmaceutical companies, biotechnology companies or research institutes are assigned to do the research and development for the projects. The MVV pays for the research from private or public contributions. During every development step such as the target validation, the identification and optimization of lead structures or the preclinical and the clinical development, the expert committees decide on the ongoing stages of each project on the basis of the available project data. Figure 1.3 shows the portfolio of the MMV.

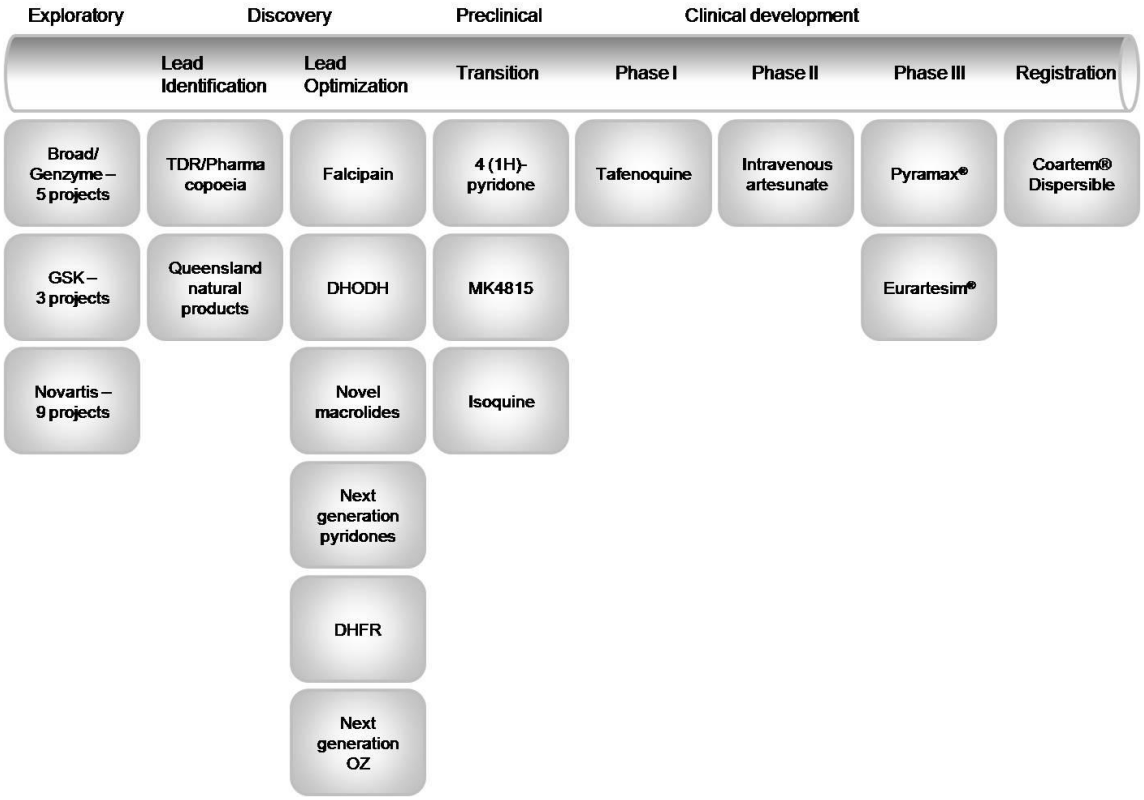


Figure 1.3: Portfolio of the Medicines for Malaria Venture (24)

Other projects also adopted the Open Source principle for drug development such as the *International Haplotype-Mapping Project* (HapMap) (25) and the *Tropical Disease Initiative* (26). The goal of the HapMap project is to develop a haplotype map of the human genome, the HapMap, which will describe the common patterns of human DNA sequence variation. The Tropical Disease Initiative provides tools for Open Source drug discovery. These tools allow scientists to work together on the development of drugs against diseases such as Malaria or Tuberculosis.

Beside the public-private partnerships, many organizations provide tools or databases, which are essential for an open drug research. These contributions are the base on which the open research is possible and this base is becoming larger and larger. Recently (2008) the *European Bioinformatics Institute* obtained the funding to provide open access to large-scale drug research data. (27) This will help in the discovery and development of new medicines through the public availability of data about drugs and small molecules.

1.4 Service Oriented Architecture

In the workflow paradigm as described in chapter 1.1 the *Service Oriented Architecture* (SOA) plays a fundamental role. A workflow environment allows the creation of workflows though the combination of different services. Thus, a workflow is a container, which describes the composition of different services and contains information about the interaction between these services.

The *Service Oriented Architecture* (SOA) is an architectural variant of computer systems that supports service orientation. Service orientation defines a way of thinking in terms of services, service-based development and the outcomes of a service. (28) There are many different definitions for SOA. The Organization for the *Advancement of Structured Information Standards* (OASIS) defines SOA as the following: “A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.” (29) Another abstract definition from *Hao He* (30) “SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners.”

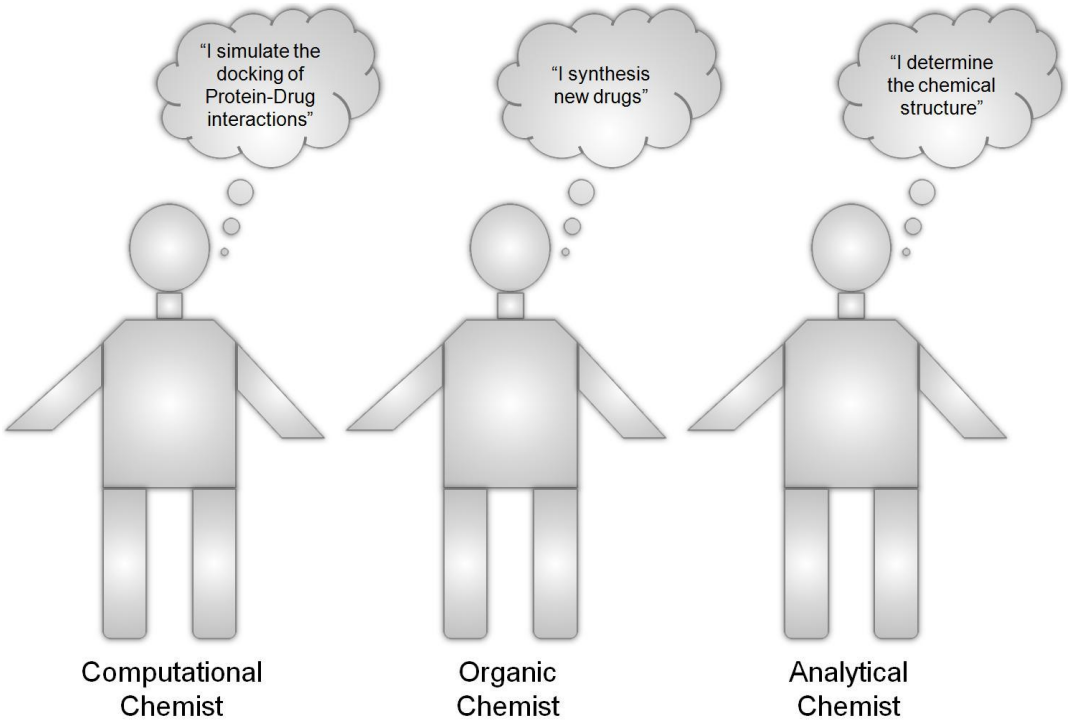


Figure 1.4: Different individuals provide different distinct services.

As long as civilisation has existed, services have played a fundamental role in the everyday world around us. Historically a service provides the fulfilment of a distinct task in support of others. Figure 1.4 shows different individuals, who provide different distinct services. A group of services combined into one service, which is a typical scenario defining a company or organization, requires the recognition of some fundamental characteristics defining each (sub)service (see Figure 1.5). Important characteristics of services are availability, reliability and the ability to communicate using the same language. (31) (32)

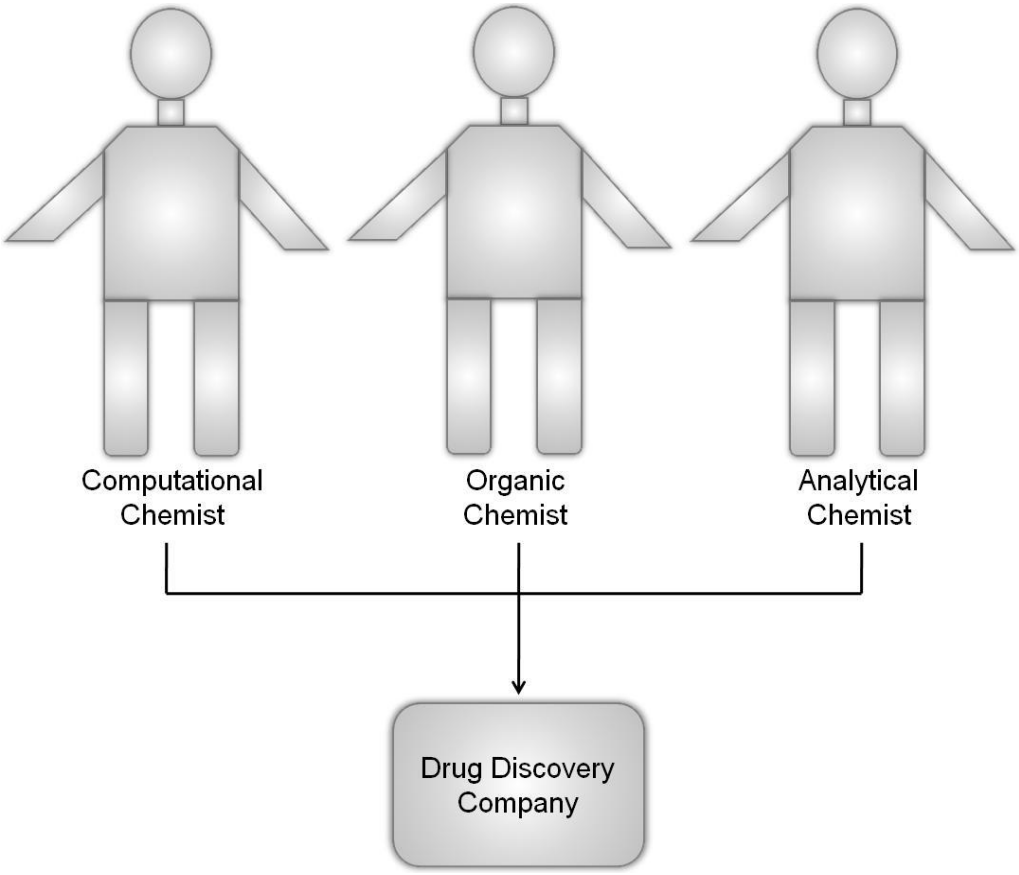


Figure 1.5: A company coordinates the services of employees to carry out its business

The SOA design paradigm follows a software engineering theory known as the "separation of concerns" (33). This theory states that the decomposing of large problems into smaller sets of problems enables a more effective solution of the problem to be found. For the design of a SOA, services have to fulfil some fundamental design requirements. (32)

- Standardized Service Contract
- Service Loose Coupling
- Service Abstraction
- Service Reusability
- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability

2 Aim of this work

The aim of this project is to build of a free and open workflow engine for cheminformatics and drug discovery after getting an overview of available solutions. These fields, like any other science, encompass sets of typical workflows. Areas calling for such workflow support include

- Chemical data filtering, transformation and migration workflows
- Chemical information retrieval related workflows (structures, reactions, object relational data etc.)
- Data analysis workflows (statistics, clustering, machine learning / computational intelligence, QSAR / QSPR / pharmacophore oriented workflows)

Parts of this project are the implementation of algorithms and applications helping scientists to create workflows for handling the described tasks. The work uses different Open Source tools to build a cheminformatics workflow solution, which allows scientists to create, run and share workflows to fit their research demands.

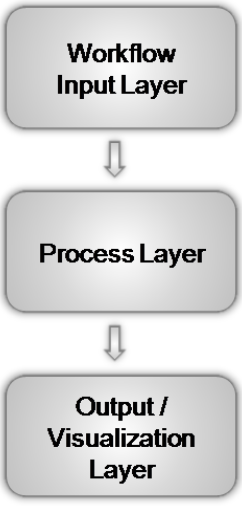


Figure 2.1: The three typical layers of each workflow

Workflows typically consist of three layers (see Figure 2.1) containing one or many services. Figure 2.2 shows the schema of a complex workflow.

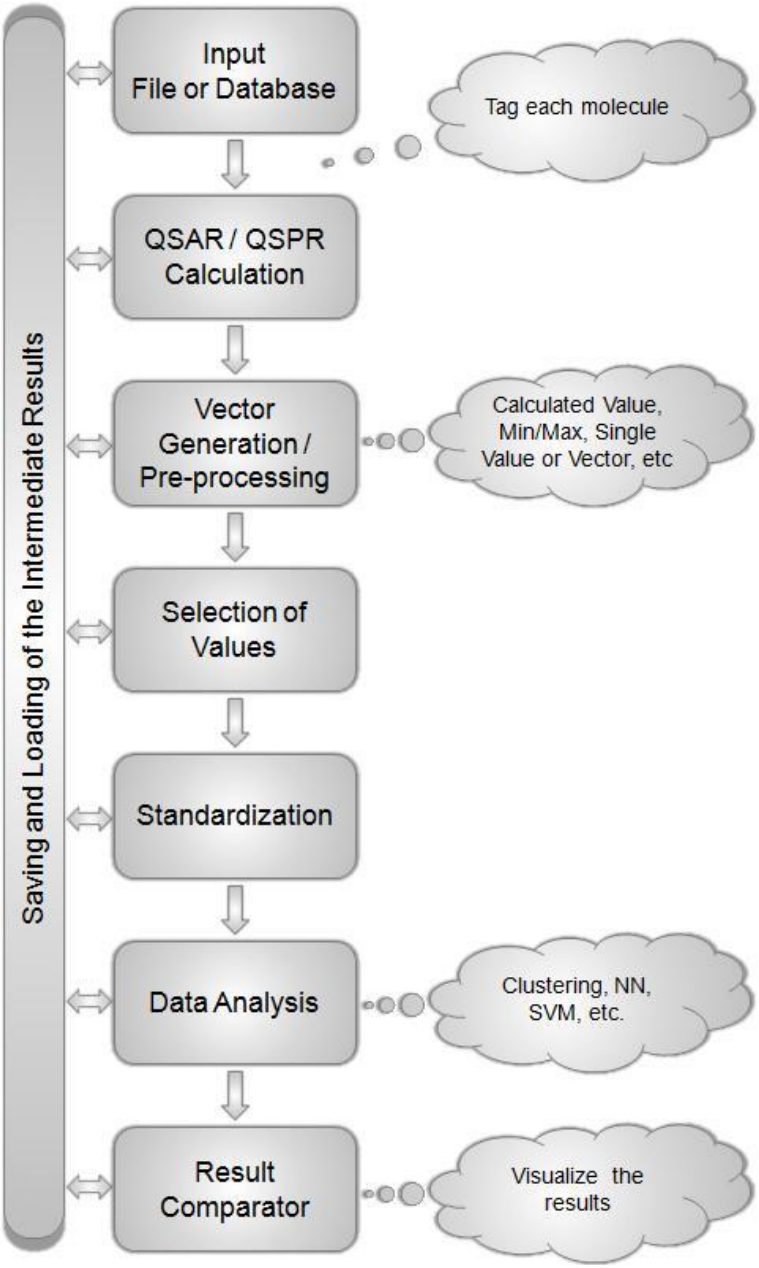


Figure 2.2: A data analysis workflow with steps for loading molecules from a file or a database, generation of data for the analysis and the comparison of different analysis results.

Through a cooperation with InterMed Discovery, a natural product lead-discovery company, a validation of the implemented methods, algorithms and applications on real world data was possible and an important part of this work.

The outcome of this project is released under the terms of an Open Source license. During the whole time of this work, all implemented methods, algorithms and workflows are publically available on the project website and on Sourceforge (34), a

public source code repository. Besides the opinion that the development of software as Open Source provides numerous advantages, especially within scientific environments, the support of *Open Drug Discovery* and *Open Notebook Science* is a contribution to help create opportunities for academia and third-world countries to perform state-of-the-art research.

3 State of Technology

This chapter will shortly introduce some of the available other workflow solutions for cheminformatics problems. The descriptions make no judgments about the described software packages. More information about the different tool kits can be found in a recent publication of *Wendy Warr* (35) or on the websites of each tool.

3.1 Pipeline Pilot

The *Pipeline Pilot* is a commercial closed source solution from *SciTegic*, a wholly owned subsidiary of *Accelrys* (36) which uses a data pipelining approach for its tool kit. Within its client-server platform the user constructs workflows by graphically combining components for data retrieval, filtering, analysis and reporting. The *Pipeline Pilot* provides three different user interface clients:

- *Professional Client* - For building new own components
- *Lite Client* - For building workflows
- *Web Portal Client* - To run standardized analysis

A number of different collections of components are available to the user. The collections cover

- The cheminformatics with the chemistry and *ADMET* component collection
- The bioinformatics with gene expression and sequence analysis component collections
- The statistics and data modeling with modeling, plate analytics, R-statistics and decision trees collections
- The image analysis collections
- The life science modeling and simulations with the *Catalyst* and *CHARMm* component collections
- The text-mining and analytics collections
- The material modeling and simulation collection
- The report and dashboard creation collection

SciTegic provides different tools for the integration of external applications and databases into *Pipeline Pilot*. Beside the integration tools, three *Software Development Kits* (SDKs) are usable to control the *Pipeline Pilot* execution from external applications.

3.2 InforSense

The *InforSense* platform (37), a commercial closed source solution, is based on a *Service Oriented Architecture* methodology and provides scalable services to build, deploy, and embed analytical applications into web portals or *Customer Relationship Management* (CRM) systems. Four set of components build the basis for the platform:

- *Analytical Workflow* - Allows the developing of analytical applications by visually constructing analytical workflows. Over 700 different modules exist including modules for data mining, statistics or domain specific modules for cheminformatics, bioinformatics, genomics and clinical research.
- *Analytical portal* - Enables the packaging of analytical workflows as reusable, interactive web applications.
- *In-database execution* – Provides methods for embedding of analytical workflows within databases. The execution of workflows within the database has the advantages of scalability, security and the data accuracy of an in-database execution.
- *Grid* – Supports a scaling up and scaling out of the infrastructure.

ChemSense the cheminformatics module for the analytical workflow allows the combination of tools from multiple cheminformatics vendors, including *ChemAxon* (38), *Daylight* (39), *MDL* (40), *Molecular Networks* (41) and *Tripos* (42). Besides the available extensions, an SDK allows the implementation of custom applications.

3.3 KNIME

KNIME (43) is a modular data exploration platform developed by *Michael Bertholds* group at the University of Konstanz, Germany. It enables the user to visually create data flows, execute analysis steps and interactively investigate the results through different

views on the data. *KNIME* provides, with its standard release, several hundred different nodes including nodes for data processing, modelling, analysis and mining. It includes different interactive views to visualize the data such as scatter plots or parallel coordinates. In addition, the joining, manipulating, partitioning and transforming of different database sources are possible with *SQL*, *Oracle* and *DB2* databases.

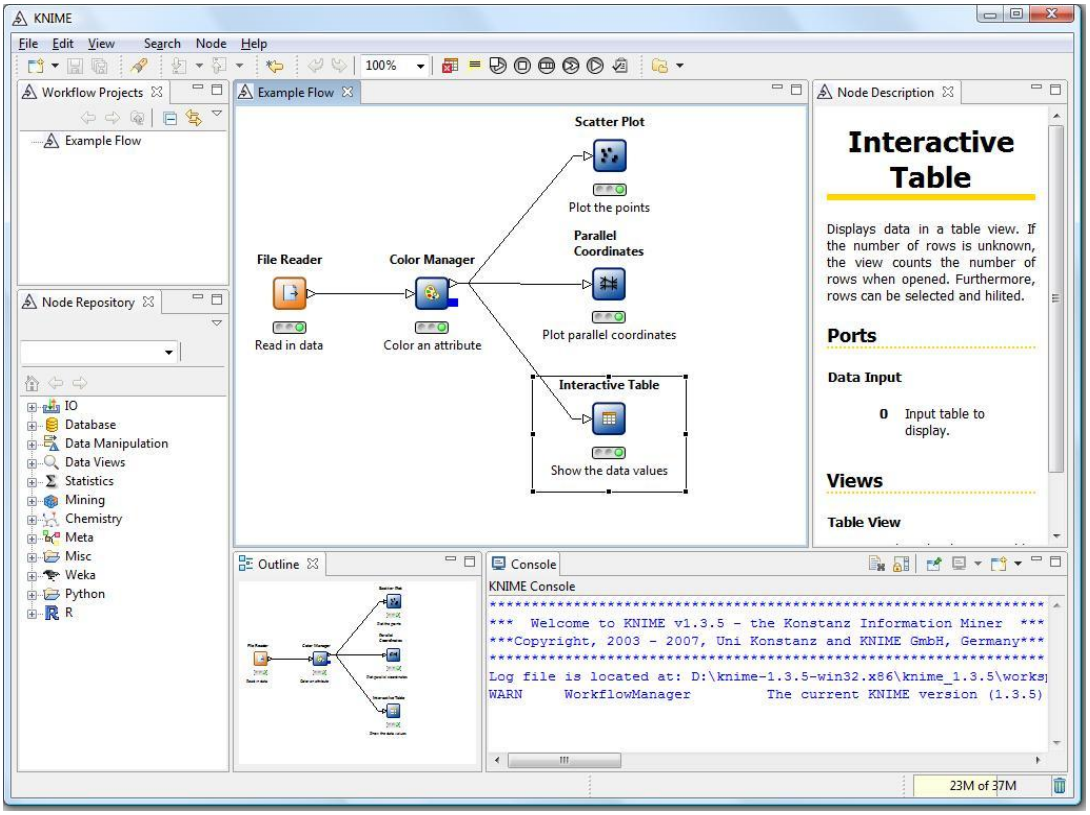


Figure 3.1: This screenshot of *KNIME* shows its typical multi window layout.

The Open Source platform *Eclipse* (44) provides the base for *KNIME* and allows with its extensible architecture and the modular *Application Programming Interface* (API), for a creation of custom nodes. Various for-profit organisations such as *Tripos* (42), *Schrödinger* (45), *ChemAxon* (38) and others have implemented sets of nodes and provide business support options for *KNIME*.

KNIME is available under the terms of the Aladdin free public license. This Open Source dual licensing model allows free use for non-profit as well as for profit-organizations but permits the distribution of *KNIME* to third parties only in exchange for payment.

3.4 Kepler

Kepler (46) is a user friendly, Open Source application for analyzing, modeling and sharing scientific data and analytical processes. The *Kepler* project is a collaborative development from the University of California at Davis, Santa Barbara and San Diego, which released its 1.0.0 version in May 2008. *Kepler's* aim is not only the facilitation of the execution of specific analyses, but it tries to support the users in the sharing and reusing of data, workflows and components. This allows the community to develop solutions to address common problems.

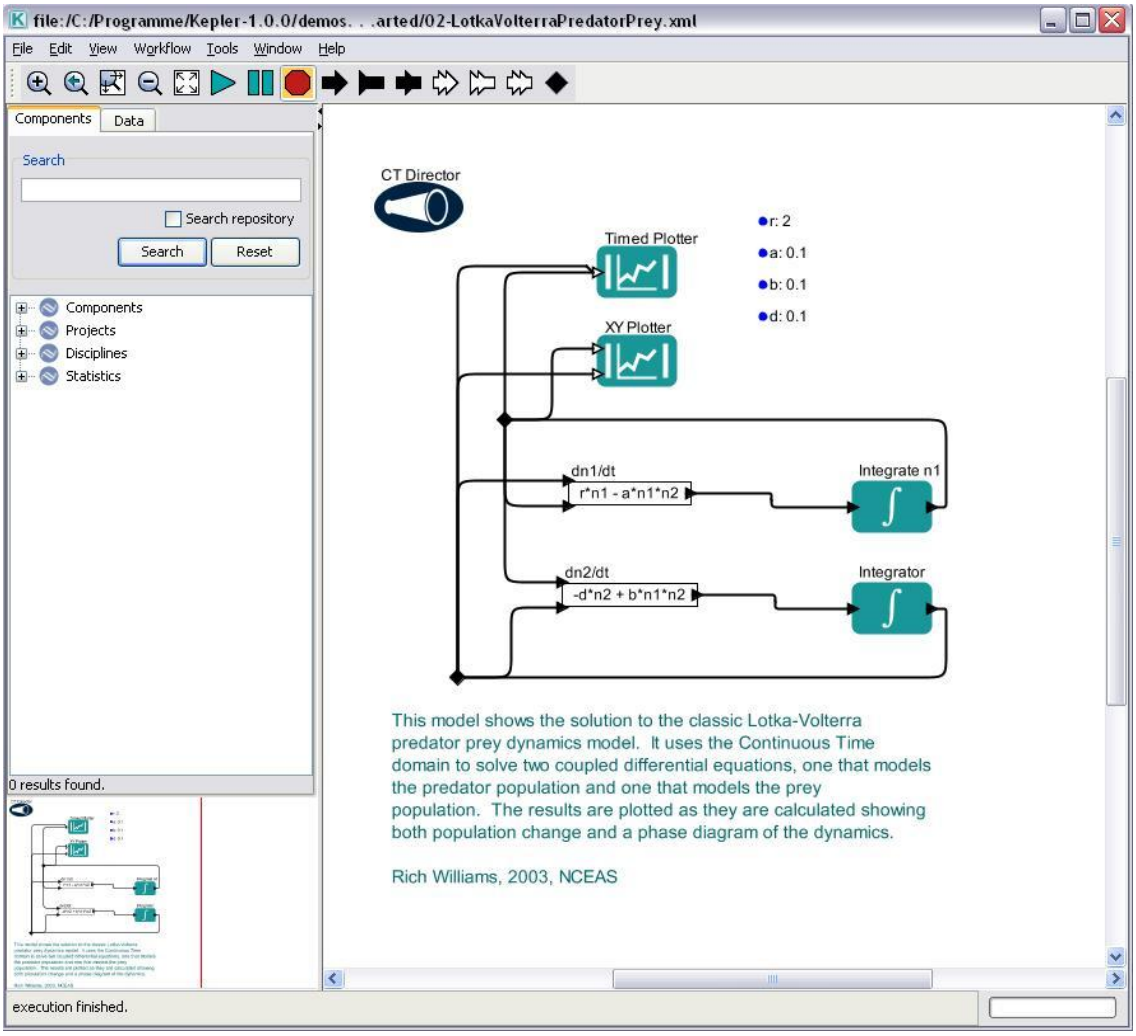


Figure 3.2: This screenshot of *Kepler* shows one of the example workflows.

The 1.0.0 release of *Kepler* already contains a library of over 350 processing components. After the customizing and connecting of different components, the created workflow is runnable on a desktop environment. Although *Kepler* has only recently been released, it is already successfully used to study the effect of climate change on species distribution (13), to simulate supernova explosions, to identify transcription factors and to perform statistical analysis.

Kepler is released and freely available under the term of the Open Source BSD License.

4 Software, Libraries and Methods

This chapter contains detailed descriptions of the software tools, libraries and methods used in the course of this work.

4.1 Taverna

The *Taverna* (47) workbench is a free Open Source software tool for the designing and executing of workflows. *Tom Oinn* at the *European Bioinformatics Institute* (EBI) leads the development of *Taverna* with the *myGrid* development team from the University of Manchester. *Taverna* releases are available under the terms of the Open Source GNU Lesser General Public License.

Taverna was originally developed for the composition and enactment of bioinformatics workflows (48). Nowadays it offers the possibility of being used in various other disciplines such as astrology, social science and cheminformatics. Originally, the *Taverna* workbench enabled scientists to orchestrate web services and existing bioinformatics applications in a graphical manner. (49)

4.1.1 Taverna's Architecture

Taverna's modular and extensible architecture enables the integration of different kind of services into the workflow execution environment. An overview of *Taverna's* architecture is shown in Figure 4.1. The *graphical user interface* (GUI) of *Taverna*, the workbench, permits browsing through the available services, the creation and running of workflows and the handling of metadata (see chapter 4.1.2).

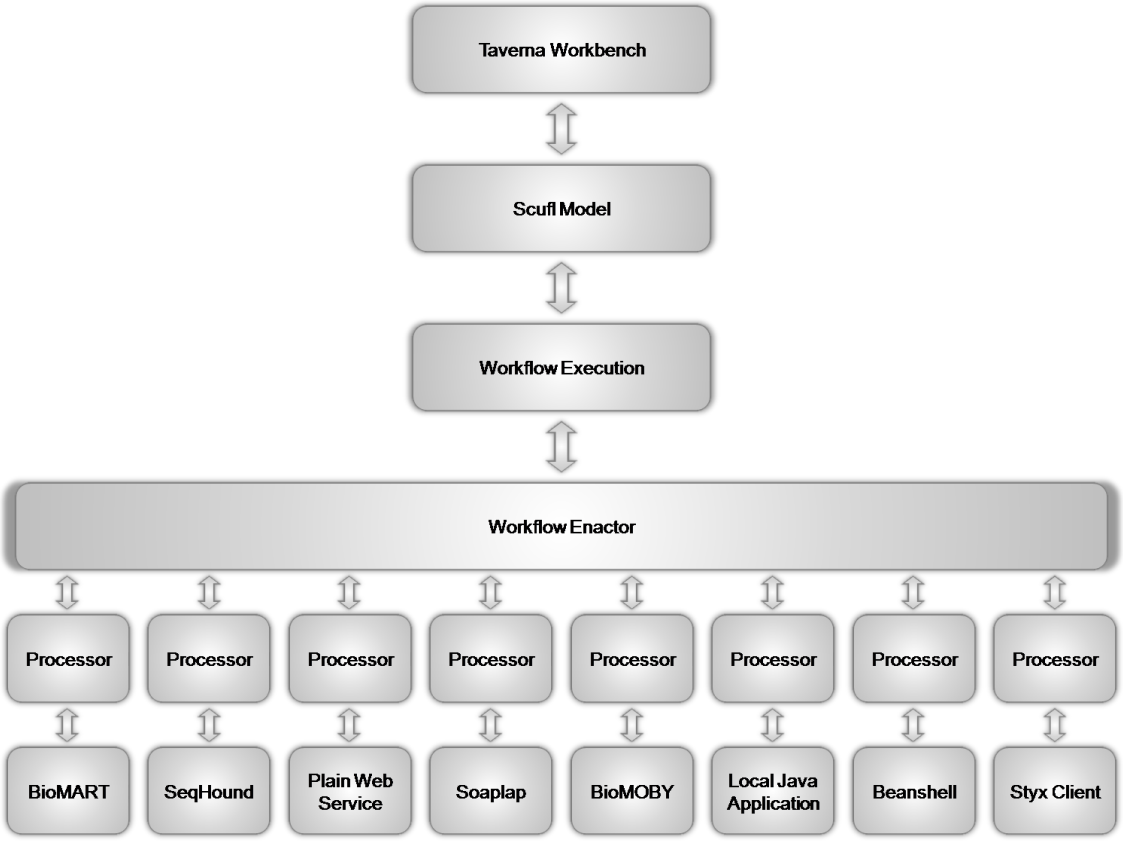


Figure 4.1: The modular architecture of *Taverna* allows the integration of different kind of tools. The *Taverna* Workbench is used as graphical user interface. The Scufl model describes a workflow in an XML-based conceptual language. The workflow execution layer uses this language to load the required processor type through the workflow enactor.

A workflow in *Taverna* represents a graph of processors. Each processor transforms a set of data inputs into a set of data outputs. The *Simple Conceptual Unified Flow Language* (Scufl) represents these workflows. Scufl is an XML-based conceptual language. Each atomic task represents one processing step of the workflow. In the Scufl language, a workflow contains three main entities:

- Processors
- Data links
- Coordination constraints

A processor is an object, externally characterized by a set of input ports, a set of output ports and a status, which can be *initializing*, *waiting*, *running*, *complete*, *failed* or *aborted*. Currently there are many different implementations for the processor available (see Figure 4.1). Each input or output port is associated with three types of metadata, a

MIME type, a semantic type based on the *myGrid* bioinformatics ontology (50) and a free textual description.

The data links indicate the flow of the data from a data source to a data sink. A processor output or a workflow input, are possible data sources. As data sink a processor input port or a workflow output is thinkable. Multiple links from one data source will distribute the same values to the different data sinks.

A coordination constraint links two processors and controls their execution. This allows the control of the order of execution of two processors with no direct data dependency between them.

The workflow execution layer uses as workflow description the Scufl language and handles the execution of the workflow. This layer controls the flow assumptions made by the user, manages the implicit iterations over collections and implements the fault recovery strategies on behalf of the user. Chapter 4.1.3 shows more details on the iteration strategy and fault tolerance of *Taverna*.

The aim of the workflow enactor layer is the interaction with and invoking of concrete services. The concept of this layer allows (through the integration of plug-ins) the extension of processor types.

The framework of *Taverna* provides three levels of extensibility: (51)

- The first level provides a plug-in framework for adding new GUI panels which allows an integration of GUI elements for the user interaction to control or configure extensions incorporated into *Taverna*. This extension is available at the workbench layer.
- The second level allows the integration of new processor types. It enables the enactment engine to recognize and invoke the new processor types and extends the workflow execution layer with new possibilities.
- The last level provides a loose integration of external components via an event-observer interface and is also available at the workflow execution layer. It enables the extension, which reacts on the events generated by the workflow enactor during the change of the state of the workflow.

4.1.2 The Taverna Workbench

The aim of the *Taverna* workbench is to provide a GUI where the user can access different service, compose a workflow, run the workflow and visualize the result of the workflow. For this, the workbench provides different views. The view for the composing of workflows is shown in Figure 4.2.

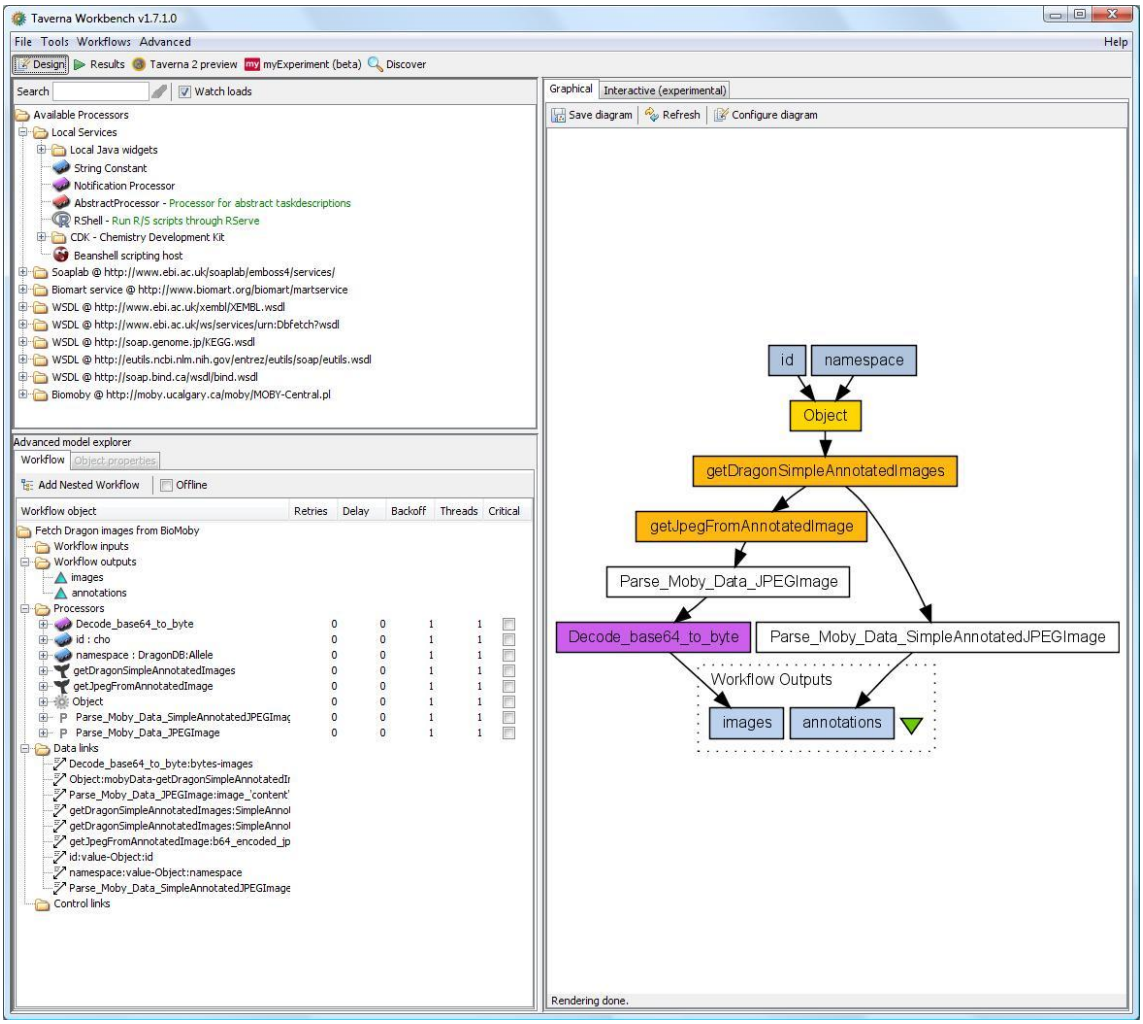


Figure 4.2: The *Taverna* workbench contains different views. Shown here is the view used for the designing of workflows. In the upper left corner, all available services are selectable. In the lower left corner all processors, links between processors and control links of the currently loaded workflow are visible. The right side of the screenshot shows the workflow diagram. This workflow fetches images and annotations of snapdragons from a public database.

Besides the view for the composing of workflows, the result view shows the state of the different processors during the execution of a workflow on one of its tabs (see Figure 4.3). This view allows also the inspection of the intermediate results of every processor.

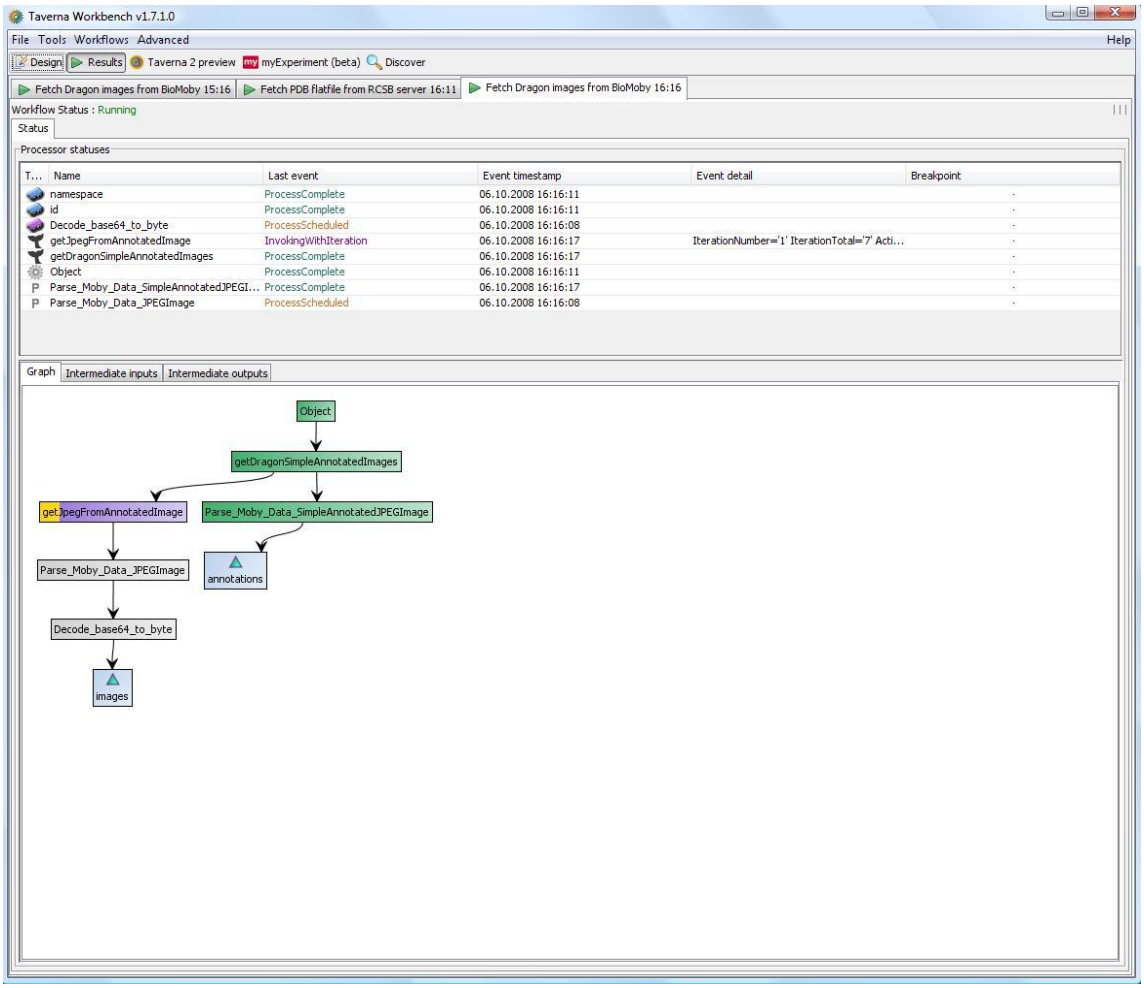


Figure 4.3: The result view shows the state of the different processors during the execution of a workflow. On the upper side it shows the list of processors from this workflow with their states. A graphical display of the state of the workflow is at the bottom of this screenshot.

Figure 4.4 shows another tab of the result view with the results of a workflow execution. In this case, the workflow results are images and annotations of snapdragons.

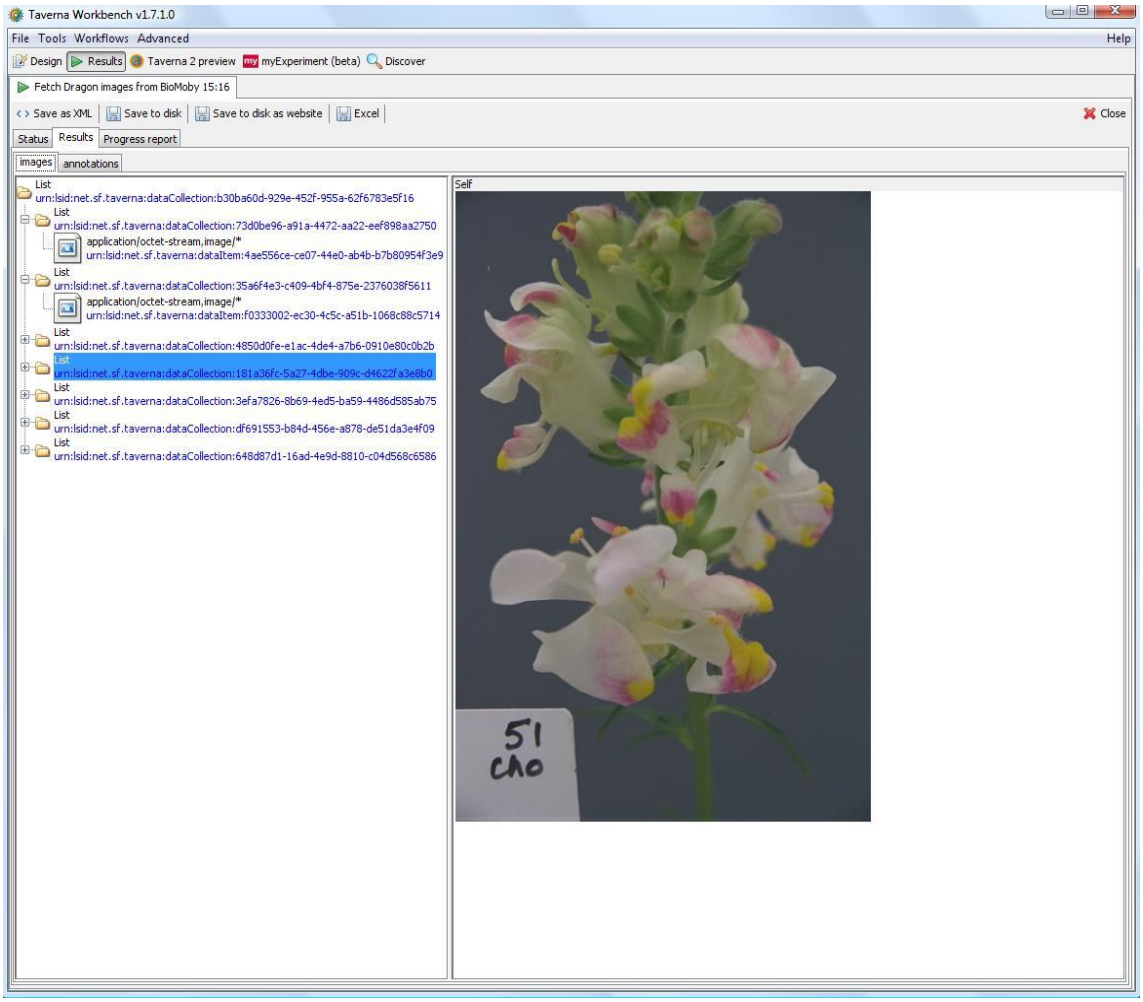


Figure 4.4: Result view showing the outcome of the workflow in Figure 4.2.

4.1.3 Iteration Strategy and Fault Tolerance

In *Taverna*, workflows are a series of inter-related components linked together by data. There the output of one component forms the input to the next. The process flows of workflows often require constructs such as the ‘if … else’ or ‘while…’ of traditional imperative languages. To support these constructs additional invocation semantics have been defined. (52)

- Iterative Semantics
- Recursive Semantics
- Fault Tolerance

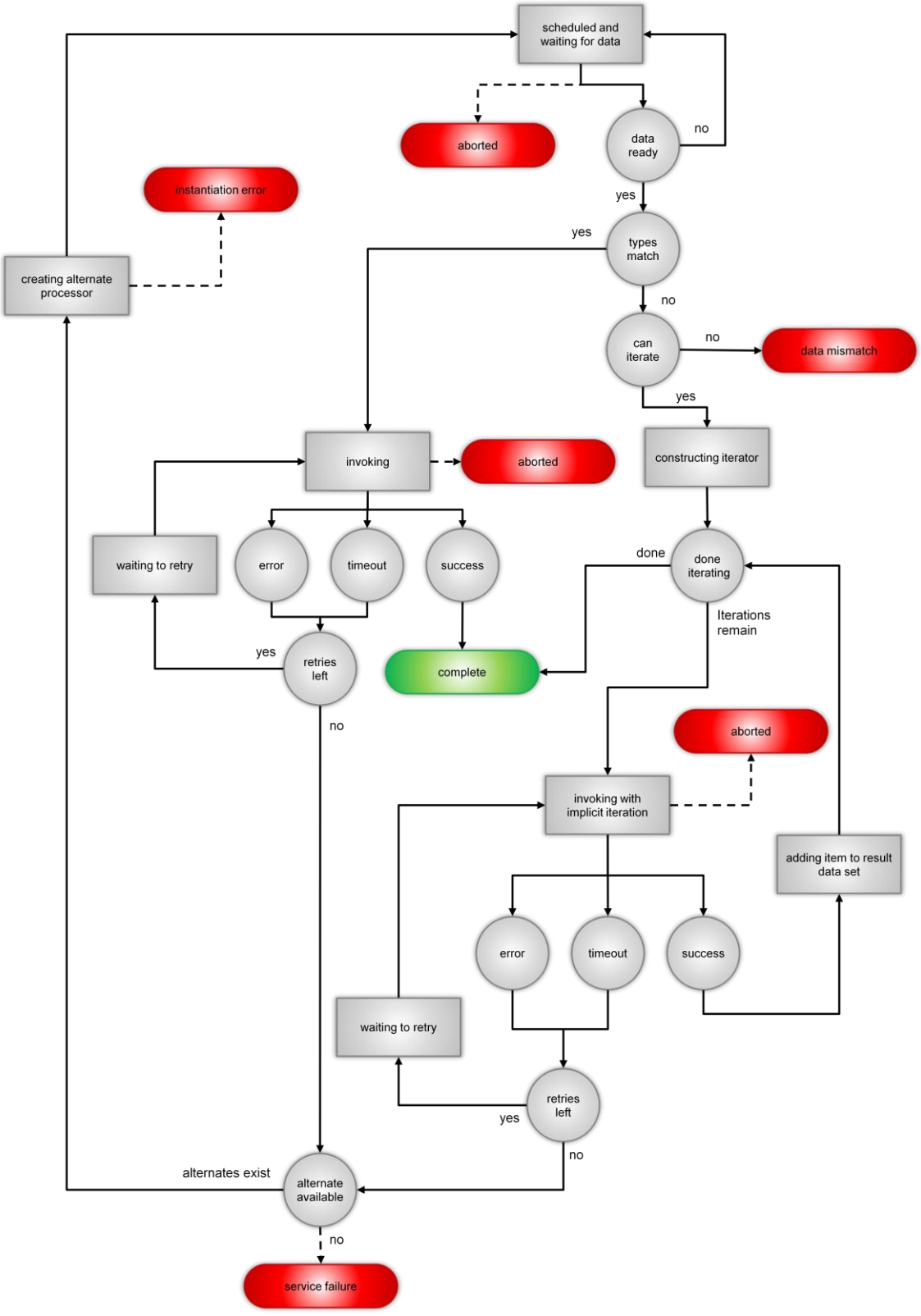


Figure 4.5: Schema of *Taverna's* state machine governing the processes of fault tolerance and iteration (52)

The iterative semantics introduces a mechanism for the introspection of input data at runtime comparing the data type expected with the currently supplied data type from the upstream process. With the help of this process, the workflow engine is able to split the incoming data sets into a queue of data items, which the operation has declared it is capable of computing. The *Taverna* workbench provides a GUI element to edit the iteration strategy if the user needs an explicit override of the split.

The recursive semantics support the repeated invocation of a single operation from the use in recursive functions or constructs like ‘call service until the result is true’.

In *Taverna* workflows, every processor has its individual fault tolerance setting. The user can specify the number of retries for a processor proclaiming a failure event, the time between the retries, or add an alternative processor to substitute the failing one. The failure settings allow the specification of so-called critical processors. If a critical processor fails, the whole workflow aborts. Otherwise, the workflow will continue to run without the invocation of the downstream processes. Figure 4.5 shows a detailed schema of *Taverna’s* state machine.

4.1.4 Nested Workflows

The *Taverna* environment features so-called nested workflows where whole workflows can be used as one processor. Nested workflows have (like any other processor) their individual fault tolerance settings.

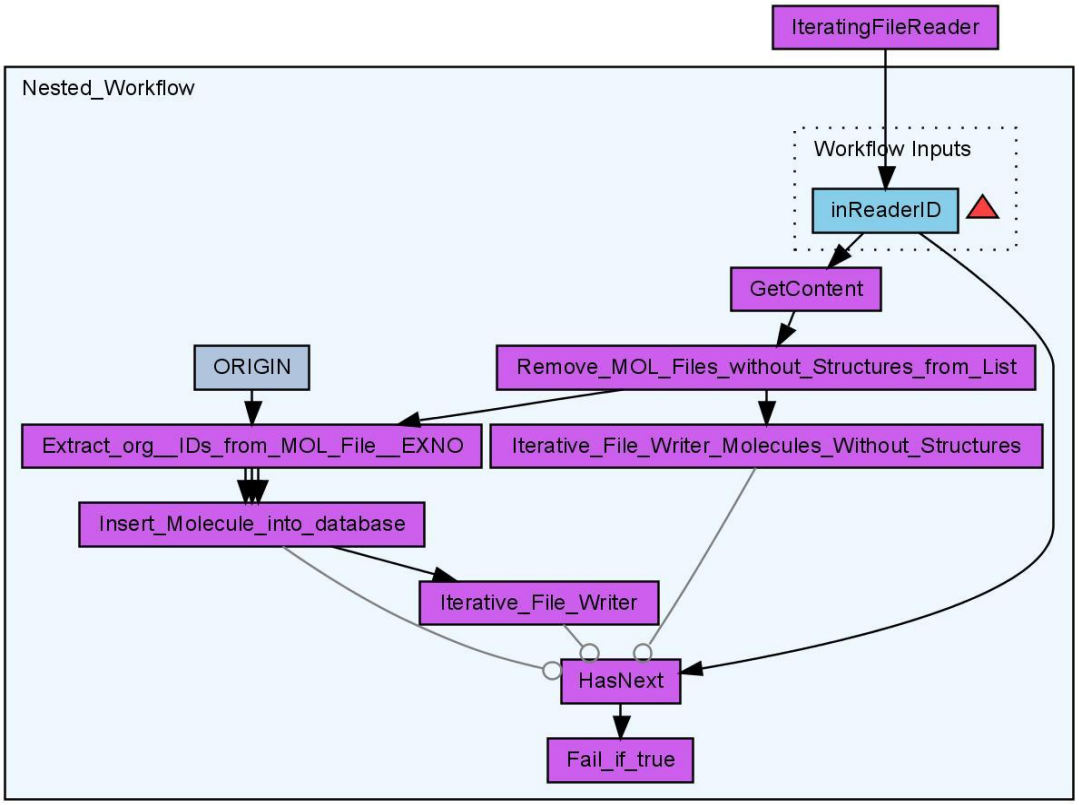


Figure 4.6: Used an iterative file reader and a nested workflow to insert molecules from a file into a database (53)

4.1.5 MyExperiment.org

A successful workflow environment needs an ecosystem of tools for supporting the complete scientific lifecycle (54). This lifecycle includes:

- designing and running of workflows
- providing service management
- discovering and publishing of services and workflows
- provenance logging

myExperiment(55) is a virtual research environment for the social sharing of workflows. On this platform workflow users can gossip about and exchange workflows, regardless of the workflow system. The reuse of workflows is effective at multiple levels (56):

- As part of the daily work a scientist reuses workflows with different parameters and data or may modify the workflow.

- The sharing of workflows with other scientists doing similar work allows a broadcasting of the workflow designer's practice(s).
- The reuse of workflows, components of workflows and workflow patterns allows the supporting of science outside its initial applications.

myExperiment provides a web based virtual research environment for the reuse of workflows. The design of *myExperiment* is inspired by the *Web 2.0* (57) community. Beside the website, *myExperiment* provides a plug-in for *Taverna*, which allows the searching, viewing and downloading of workflow directly within the *Taverna* workbench.

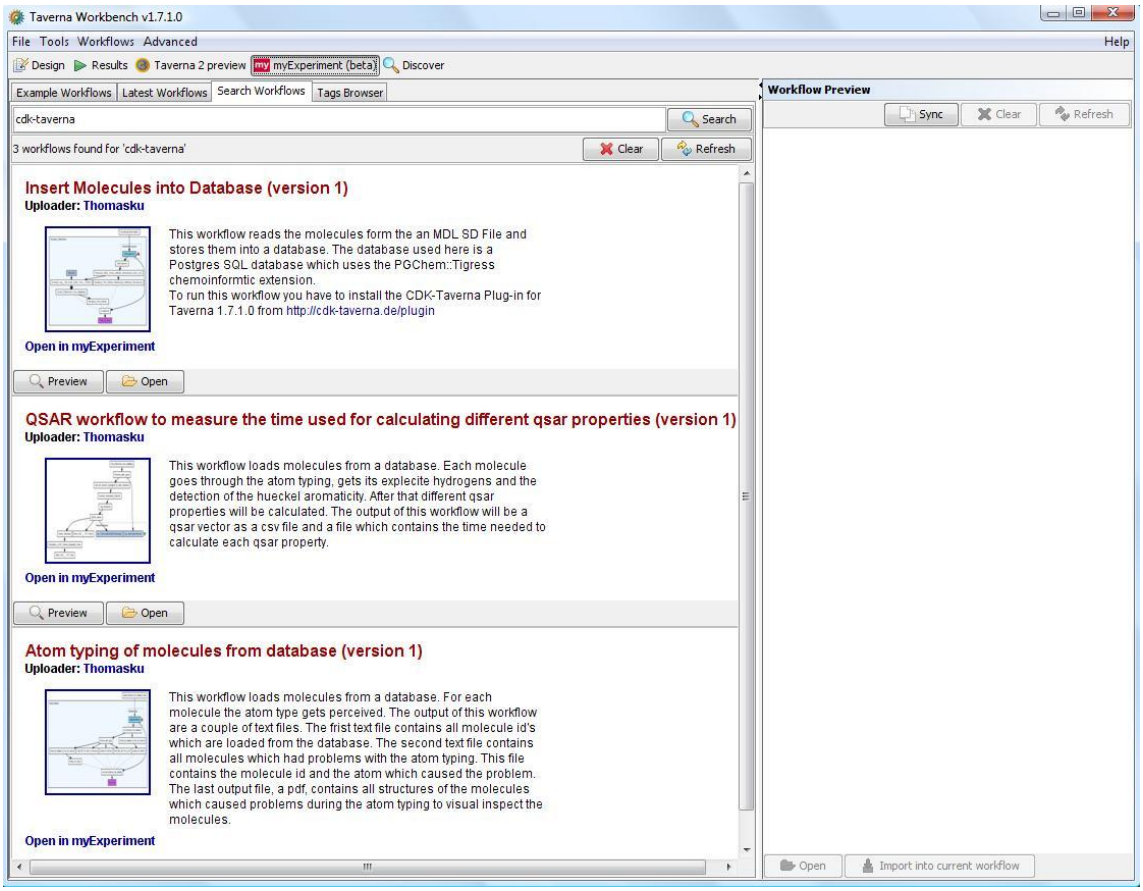


Figure 4.7: The *myExperiment* plug-in enables the search of workflows, visualize the workflows and provides the possibility to download them directly into the workbench.

4.1.6 Taverna 2

The *Taverna* team is working on a complete redesign of the system. The new *Taverna* design includes

- A new graphical user interface which supports the graphical designing of workflows
- Improved performance via iteration and streaming
- Management of large volumes of data
- Better remote workflow execution
- Integration with grid resources
- Decreased memory "footprint"
- Support for provenance capture

Currently the *Taverna* team has released beta versions of the upcoming *Taverna* 2.

4.2 The Chemistry Development Kit

The *Chemistry Development Kit* (CDK) (58) (59) is a Java library for structural chemo- and bioinformatics. The CDK originated under the aegis of *Christoph Steinbeck* as a successor to some of his old libraries in autumn 2000. Over the years, the CDK library has evolved into a fully blown cheminformatics package with code extending from QSAR calculations up to 2D and 3D model building. The following chapters, will introduce briefly some of the mayor functionalities of the CDK. The information presented in the chapter are based on the articles (58) (59) , the website and the source code.

4.2.1 2D Structure Graphical Handling

For any cheminformatics-related program, the ability to display and manipulate 2D drawings of chemical structures is one of the most important features. Consequently, integrated and based on the CDK, *JChemPaint* has been developed (60). For the graphical handling of chemical structures, the library includes the capability of generating coordinates (see chapter 4.2.2) for those chemical structures, which have

been created by a structure generator (see chapter 4.2.3) as chemical graphs without coordinates. Figure 4.8 shows *JChemPaint*.

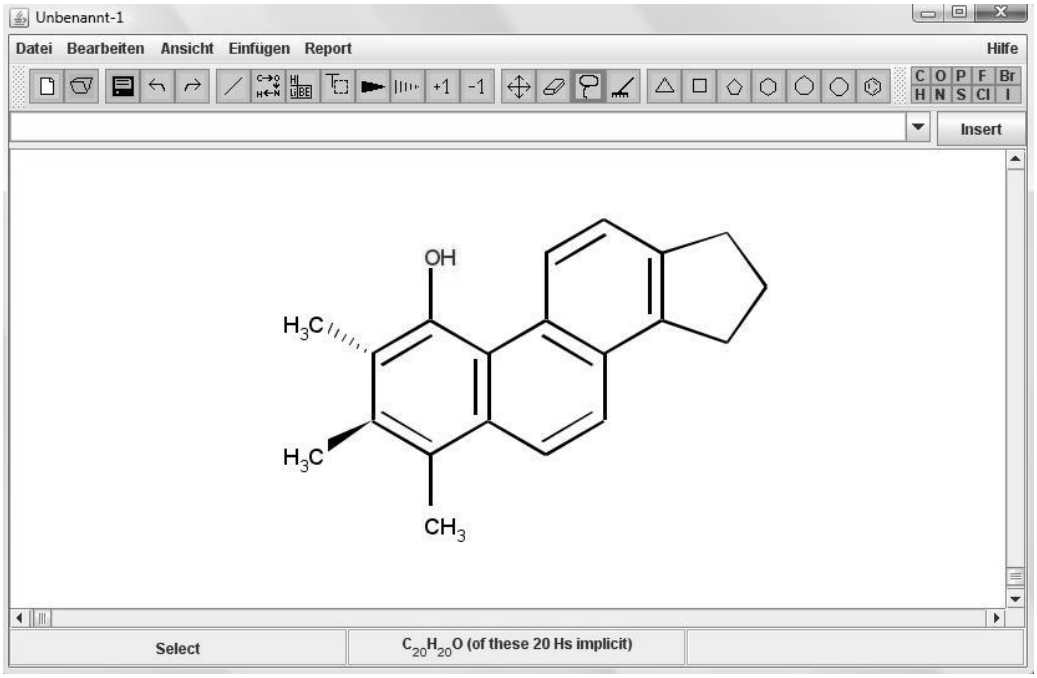


Figure 4.8: *JChemPaint* is a CDK based 2D structure editor

4.2.2 Structure Diagram Layout

Virtual combinatorial chemistry, virtual screening or computer-assisted structure elucidation handles chemical structures frequently as one-dimensional graphs. In any of these key fields of cheminformatics, however, it is required to present the structure in graphical form to a chemist. This has necessitated a tool to generate 2D or 3D coordinates. The process of the generation of coordinates has been termed *Structure Diagram Generation* (61). The CDK features a powerful 2D structure diagram generator.

4.2.3 Structure Generators

The CDK contains some simple structure generators. *SENECA* (62) a tool for the computer-assisted structure elucidation uses these generators. Based on a specific molecular formula, the *SingleRandomStructureGenerator* class generates a totally random structure within the constitutional space. The *RandomGenerator* than uses these

erratically generated structures for small random moves in the constitution space, based on an algorithm suggested by *Faulon* (63).

4.2.4 Ring Searchers

The structure diagram generator, for example, uses an implementation of *John Figueras'* fast algorithm for finding the *Smallest Set of Smallest Rings* (SSSR) (64). These packages contains classes, besides the ring searchers, for partitioning a given ring system into *AtomContainers* which leads to one *AtomContainer* for each ring. For the aromaticity detection, it is essential to compute the *Set of All Rings* (SAR). Therefore, the CDK implements an efficient and fast algorithm given by *Hanser et al*. (65).

4.2.5 Aromaticity Detection

The CDK contains a package for the various definitions of aromaticity. Each definition has at least one way of detecting the aromaticity. Currently the CDK contains the *CDKHueckelAromaticityDetector* class, which implements the well-known *Hückel* rule. After the SAR detection algorithm by the *Hanser et al*. (see section 4.2.4) this class starts with the largest detected ring, where it counts the number of alternating bonds or triple bonds and takes into account the free electrons of heteroatoms. According to the *Hückel* rule, the class checks whether the ring contains $4n + 2$ π-electrons. All atoms and bonds of this ring are marked as aromatic before the search continues with the remaining rings.

4.2.6 Isomorphism

The determination of identical chemical structures is one of the important capabilities of a cheminformatics library. The Isomorphism package of the CDK contains a versatile module for *Maximum Common Substructure Searches* (MCSS). The MCSS, the most general case of graph matching, allows the determination of structure identity, of subgraph matching and of maximum common substructure searches.

4.2.7 File Input/Output

The CDK follows a generalized approach for the file input and output. The *ChemObjectReader* and the *ChemObjectWriter* classes are the base interfaces for all file formats. Thus, two separated classes implementing these interfaces represent each file format. Among others the CDK supports IO classes for XYZ, MDL molfile(66), PDB(67) and CML(10).

4.2.8 SMILES

The *Simplified Molecular Line Entry Specification* (SMILES) provides a compact and relatively simple string representation of molecular constitutions (68). The CDK supports this widely used coordinateless interchange format. Graph isomorphism checks are also possible based on the specification of unique (canonical) SMILES(69). To support this, the CDK provides a canonical SMILES generator, which implements all of the SMILES standards, including chirality.

4.2.9 InChI™

The *IUPAC International Chemical Identifier* InChI™(70) is a non-proprietary identifier for chemical substances. This identifier is usable with printable and electronic data sources and thus enables an easier linking between diverse data compilations. The CDK provides the possibility of creating a chemical substance from an InChI™ and vice versa.

4.2.10 Fingerprints

Fingerprints are mainly used as prefilters for isomorphism checking. These prefilters are commonly used for searching for molecules in databases. As a result, the CDK contains different fingerprint implementations. The *Fingerprinter* class, for example, produces *Daylight*-type fingerprints (71) or the *EStateFingerprinter* generates a 79 bit long fingerprint using the *E-State* fragments described by *Hall and Kier* (72).

4.2.11 Molecular Descriptors

The CDK contains packages for atomic, bond, atompair and molecular descriptors. Molecular descriptors create numeric values for a mathematical characterization of the structure and the environment of a molecule. These values can be used for database searches, *Quantitative Structure-Activity Relationships* (QSAR) and *Quantitative Structure-Property Relationships* (QSPR). Each descriptor implementation of the CDK contains meta-data as a supplement. This meta-data includes information regarding the author, version, title of implementation and a reference to a dictionary describing the descriptors. The dictionary was developed within the independent Open Source QSAR (73) project and thus is not CDK specific. It contains information such as the reference to the original literature, mathematical formulae specifying the descriptor, links to related descriptors and other details of the algorithm. Table 1 gives an overview of the implemented molecular descriptors of the CDK.

Table 1: A summary of descriptor types currently available in the CDK (50)

Class	**Implemented descriptors**	**Reference**
Constitutional	Atom and bond counts, molecular weight	
	Aromatic atom and bond counts	
	Hydrogen bond donor / acceptor counts	
	Rotatable bond count	
	Proton type	
	Pi-contact of two atoms	(74)
	Proton RDF	(75)
	Rule of Five	(76)
	XLogP	(77)
Topological	χt indices (0χt and 1χt)	(78)(79)(80)
	χv indices (0χv and 1χv)	(78)(79)(80)
	Wiener number	(81)
	Zagreb index	(82)
	Vertex adjacency information	
	Atomic degree	
	Petitjean Number	(83)
	κ shape indices (1κ, 2κ, 3κ)	(84)(85)(86)
Geometric	Gravitational indices	(87)

	Shortest path bond count	(74)
	Moment of inertia	(88)
	Distance in space	(74)
Electronic	Sigma electronegativity	
	Proton partial charges	
	Van der Waals radii	
	Number of valence electrons	
	Polarizability (effective, sum, difference)	
Hybrid	BCUT	(89)(90)
	WHIM	(91; 92)
	Topological surface area	(93)

4.2.12 Tools

The CDK contains tools for various problems, for example:

- The *ConnectivityChecker* class checks whether a given chemical graph is connected
- The *PathTools* class provides methods for finding the shortest path in a molecule between two given atoms.
- The *MFAnalyser* class contains methods for retrieving the molecular formula for a given molecule
- The *HOSECodeGenerator* class produces *HOSE* codes (94) for a given molecule.
- The *BremserOneSphereHOSECodePredictor* class uses *HOSE* codes for the prediction of expectation ranges for carbon-13 NMR chemical shifts (95).

4.3 Other libraries used

This chapter briefly describes some of the other main libraries used within this project.

4.3.1 PostgreSQL Database with the Pgchem::tigress extension

Besides the need to access the databases in the public domain, the creation and use of local database instances is an important feature for a workflow system. The *PostgreSQL*

(96) Open Source relational database system has earned a strong reputation for reliability, data integrity and correctness after over 15 years of active development. Based on the *PostgreSQL* database system, *Ernst-Georg Schmidt* developed *Pgchem::tigress* (97), an Open Source cheminformatics cartridge. This cartridge provides its functionality based on a wrapper around the checkmol/matchmol (98) molecular analyser and the *OpenBabel* (99) computational chemistry package, plus some database functions, datatypes, *a Generalised Search Tree* (GiST) index and auxiliary tables. All functionality of the *Pgchem::tigress* extension is accessible through SQL statements. This includes the following functionality:

- Exact and substructure search on molecules
- Searching by functional groups
- Calculation of chemical properties such as molecular formula and molecular weight
- Tanimoto similarity searching

Pgchem::tigress supports the MDL V2000 and V3000 molfile, SIMLES and InChIs as input and output formats for molecular structures.

4.3.2 Weka

Weka (100) is an Open Source project, which contains a collection of machine learning algorithms for data mining tasks. It supports tools for data pre-processing, classification, regression, clustering, association rules and for data visualisation. *Weka* is distributed under the GNU Public License and written in *Java*. Besides the *Java* library, *Weka* supports a graphical user interface, which includes data visualisation and an environment for comparing learning algorithms (see Figure 4.9).

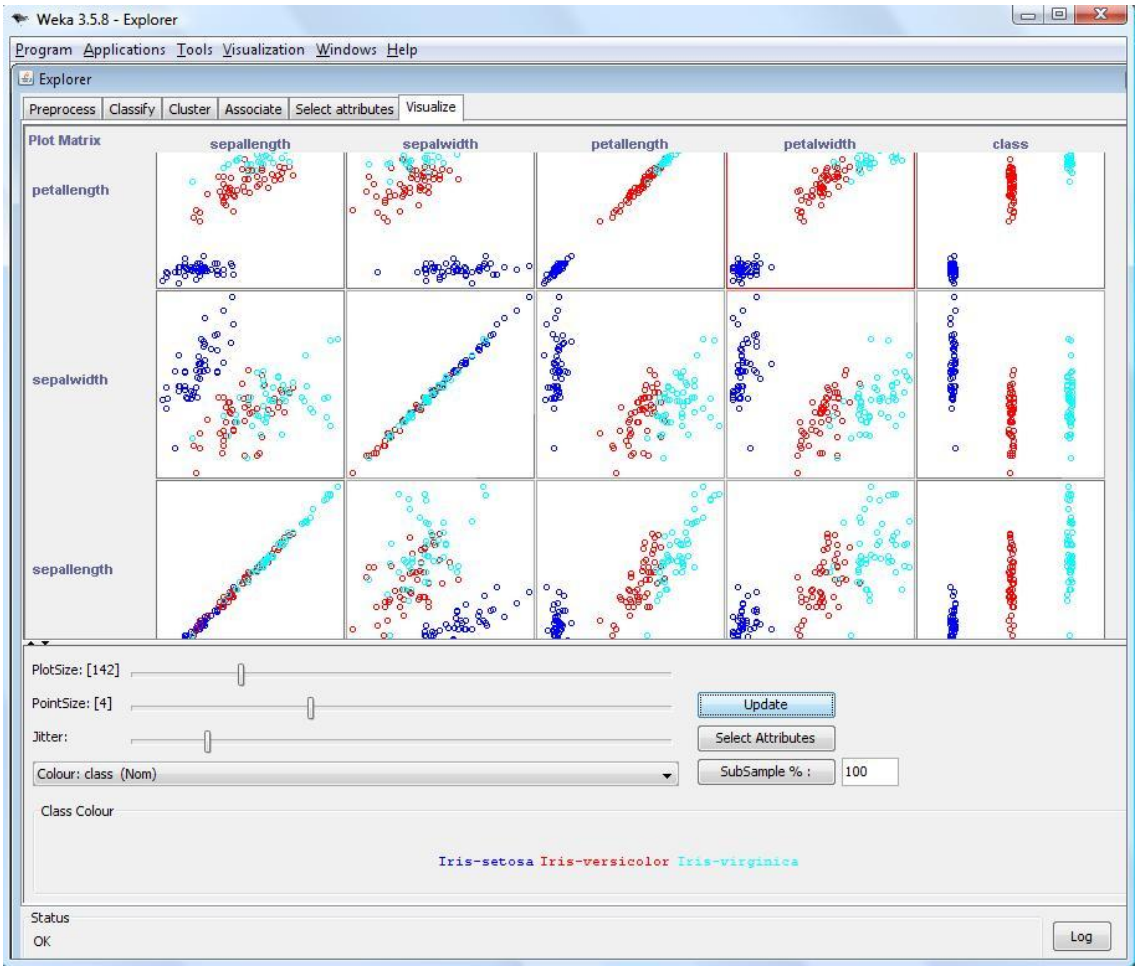


Figure 4.9: The *Weka* Explorer enables the applying of different data mining methods to datasets by using a graphical user interface and the visualisation of the experimental results.

4.3.3 Bioclipse

Bioclipse (101) is a Java-based, Open Source, visual platform for chem- and bioinformatics based on the *Eclipse Rich Client Platform* (RCP). *Bioclipse* uses a plug-in architecture that inherits functionality and visual interfaces from Eclipse. It contains features for chem- and bioinformatics, and extension points that allow the extension by plug-ins to provide further functionality. This includes a 2D molecular structure editor and a 3D visualisation of molecules (see Figure 4.10).

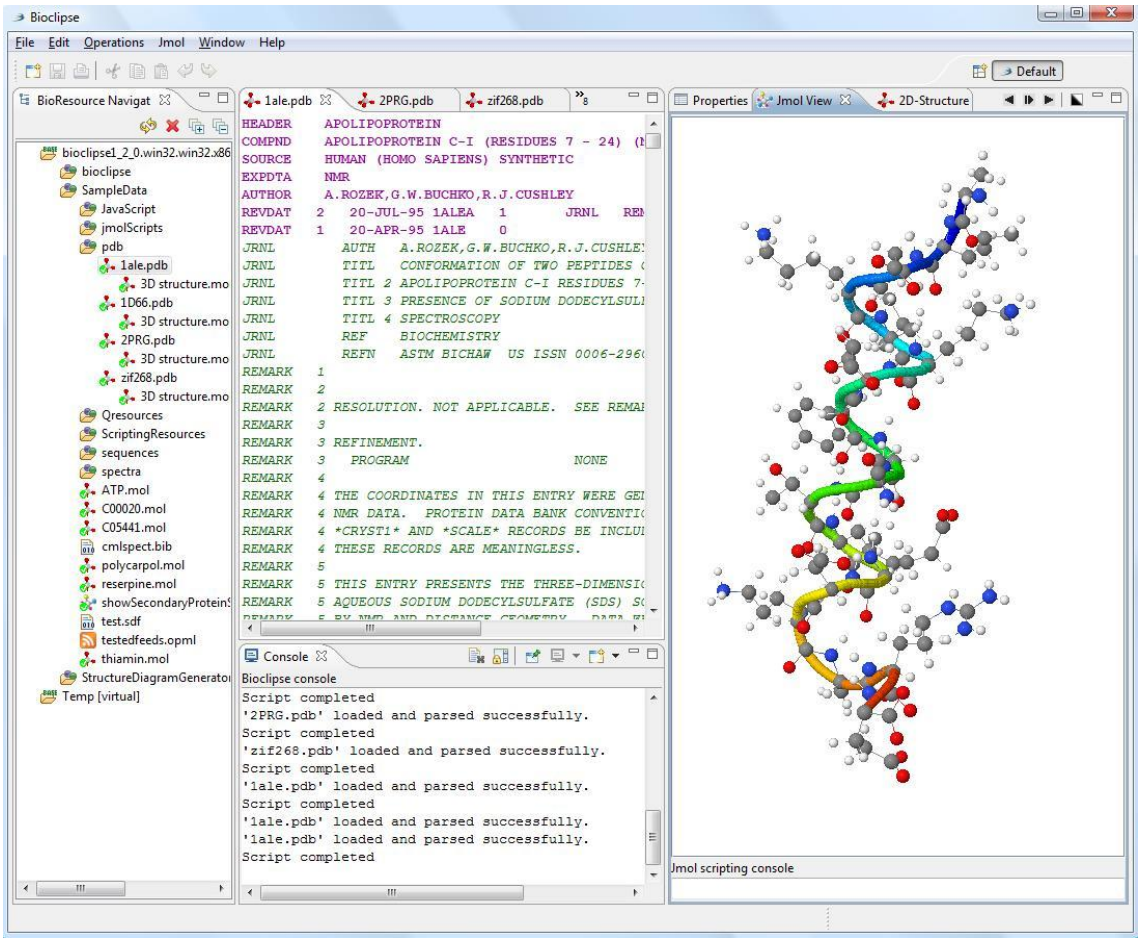


Figure 4.10: Bioclipse contains different visualisation and editing components for chem- and bioinformatics.

4.4 Machine Learning

The amount of data and the availability of readily usable computational power are increasing exponentially in today's world. The increase in computational power is still following *Moore's* law (102) - from the mid-1970s about a doubling of the transistors on an integrated circuit every twenty-four months (and the cost being halved).

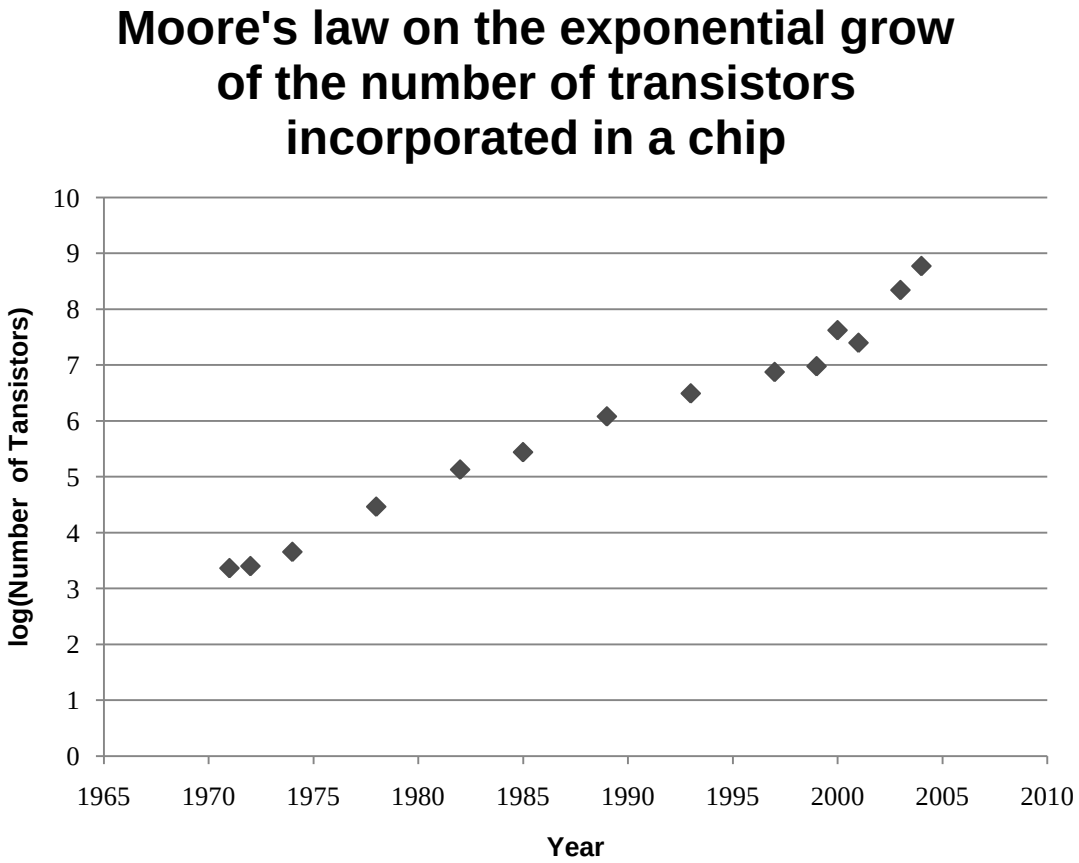


Figure 4.11: The number of transistors incorporated in a chip follows *Moore's* law. This diagram shows a logarithmic plot of the number of transistors incorporated in chips with the year of its introduction (103)

This free availability of computational power and the omnipresence of personal computers are some reasons for the *Information Explosion* (104). Machine learning is a promising method for finding information hidden in huge volumes of data and for making it available in a more "user friendly" form (100).

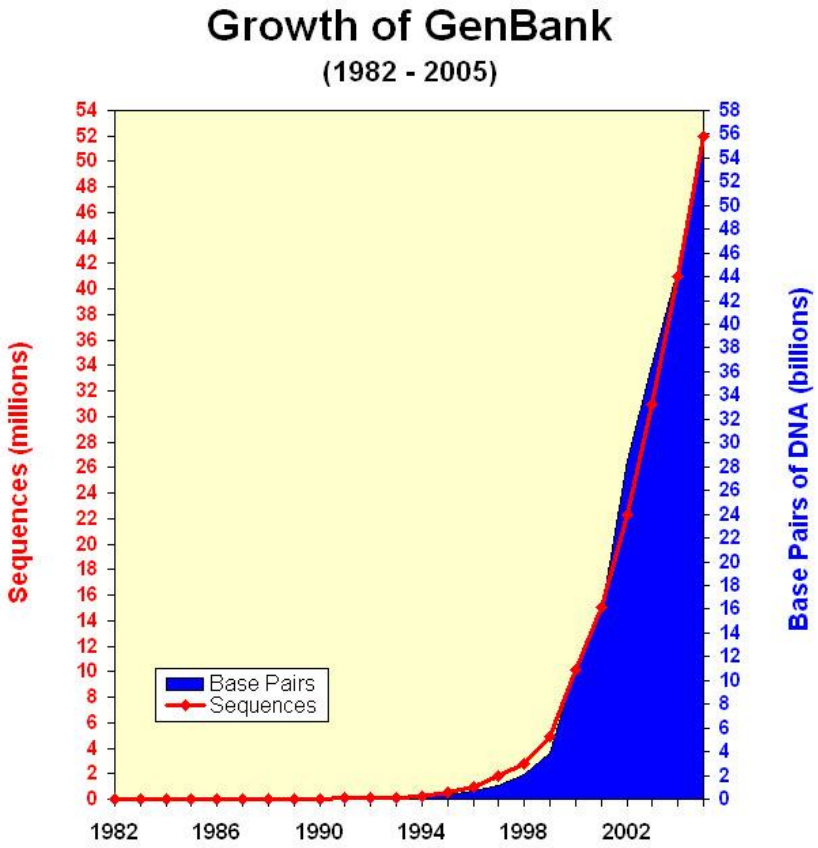


Figure 4.12: The exponential growth of *GenBank* is a typical indication of the effects of the information explosion. (105)

Machine learning as sub-field of artificial intelligence tries to enable computers to "learn". Generally, learning as a process can be divided into the categories of inductive and deductive learning (106). Inductive machine learning methods are responsible for the extraction of pattern and rules. Machine learning is defined today as "the study of computer algorithms that improve automatically through experience" (107) and its techniques are used in wide range of applications including search engines, natural language recognition (108), speech (109) and handwriting recognition, bioinformatics and cheminformatics. Figure 4.13 shows the disciplines which influence Machine Learning. (110)

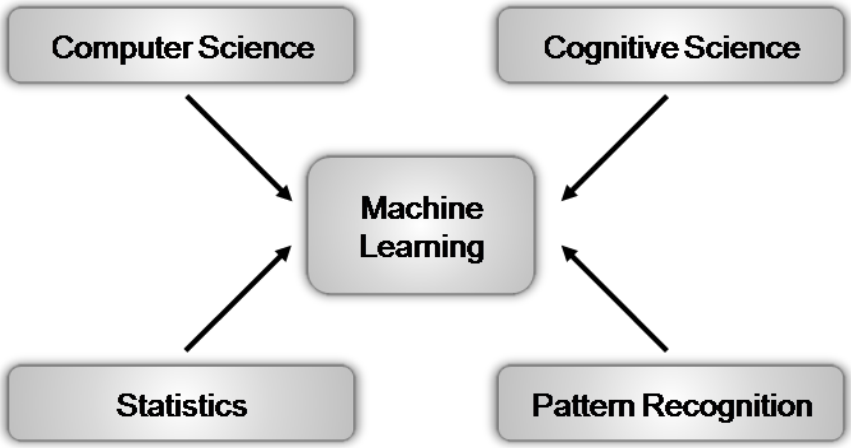


Figure 4.13: Disciplines involved in the process of Machine Learning (110)

For the process of machine learning a dataset usually gets divided into training and test datasets. The training dataset is used to train and build a mathematical model, which is evaluated applying a test dataset. The generated model should generalize from its training data and be able to predict outputs. Models, which only have memorized their inputs, will not be able to generalize. The process of machine learning differs between two learning concepts, supervised and unsupervised learning

4.4.1 Supervised Learning

Supervised learning uses a training dataset, which consists of pairs of input objects, and desired output objects to build a model. During the training, the model tries to fit its specific adaptive modelling function to generate the correct output for a newly given input. The comparison of the model's output with the correct output can lead to an error. The task of the training is to minimize these errors by adaption of the parameters of the modelling function. Figure 4.14 shows the process of supervised machine learning for a real world problem.

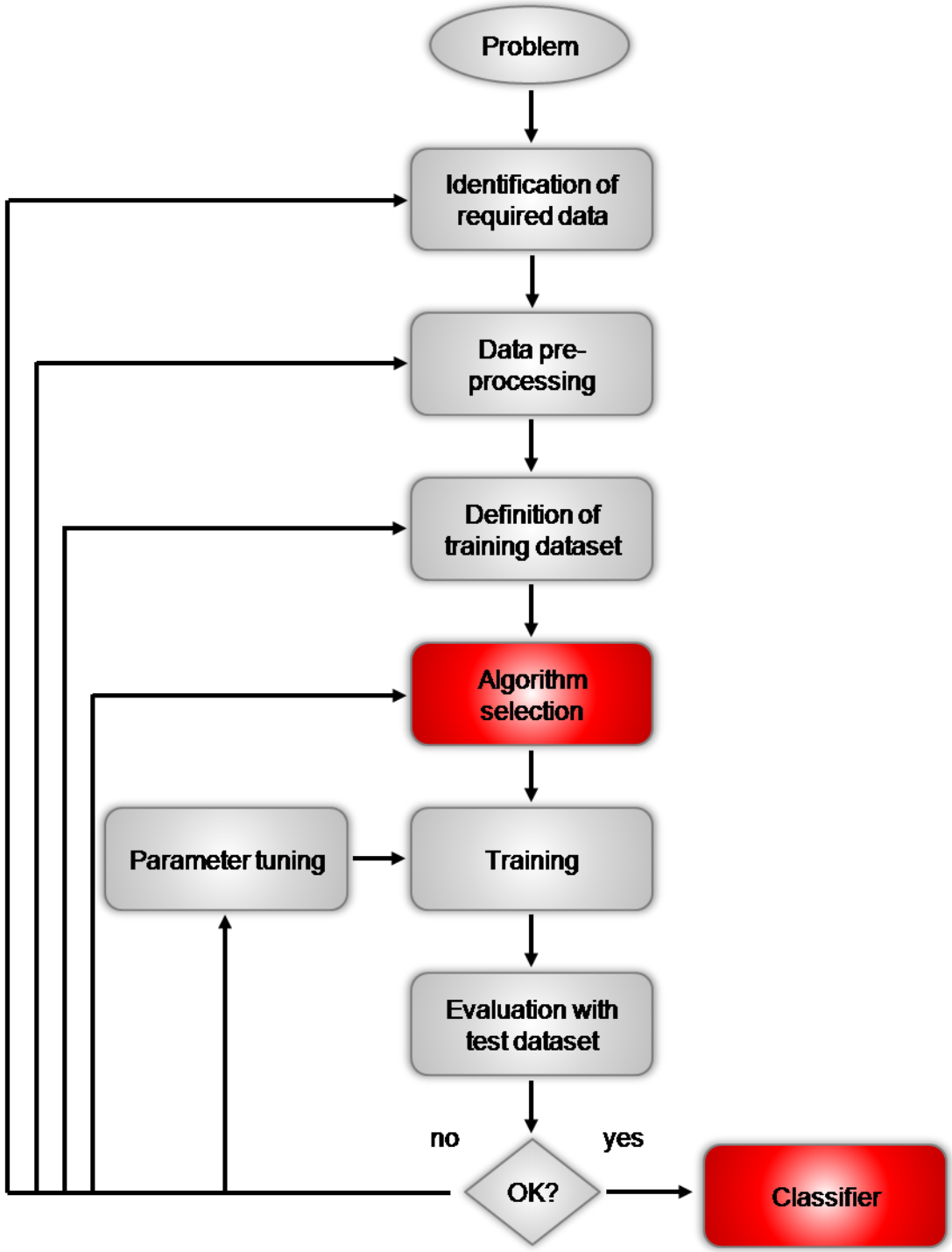


Figure 4.14: Process of supervised machine learning (111)

Decision trees, perceptron-type artificial neural networks or support vector machines commonly use the supervised machine learning strategy.

4.4.2 Unsupervised Learning

The unsupervised learning method requires a set of input data only to detect patterns within the data. Typical applications for unsupervised machine learning methods are clustering, data compression and outlier detection. Methods based on unsupervised learning are *Self-Organizing Maps* (SOMs), also known as *Kohonen* networks or the *Adaptive Resonance Theory* (ART).

4.4.3 Artificial Neural Networks

The brain is a complex, nonlinear and parallel computer that has the ability to perform tasks such as pattern recognition, perception and motor control. The idea for *Artificial Neural Networks* or *Neural Networks* (NN) was inspired by the study of the central nervous system. There the neurons with their axons, dendrites and synapses build the most significant processing element. The artificial neuron (AN) is a model of the biological neuron. The *McCulloch-Pitts* (112) neuron, a model AN, receives signals from other ANs or the environment. Each AN makes a spatiotemporal integration of the input signals. Usually the sum of each input signal is weighted and passed through a nonlinear function. This function is known as the activation function or the transfer function.

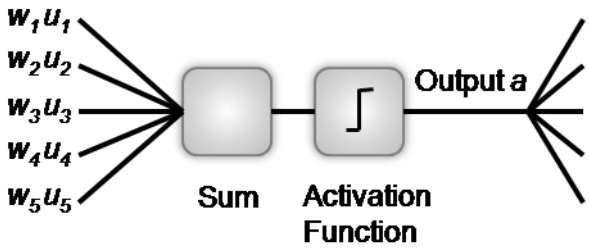


Figure 4.15: *McCulloch-Pitts Neuron* containing the spatiotemporal integration of the input signals and the activation function. The input u_i represents a weighted signal and w_i the weight.

Figure 4.16 shows an ANN as a layered network of ANs, e.g. the widely used perceptron-type ANN contains an input layer, a hidden layer and an output layer. Two AN layers are fully or partially connected.

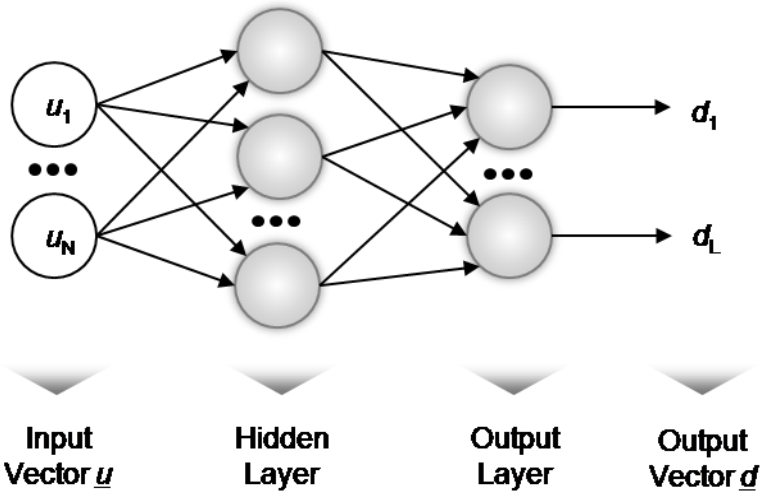


Figure 4.16: This three-layer perceptron represents a ANN which is built from an feed-forward interconnected group of nodes.

The research of the last decades involved many different types of ANNs' (113)

- Single-layer ANNs, such as the *Hopfield* network (114)
- Multilayer feed-forward ANNs, including back propagation, functional link and product unit networks
- Temporal ANNs, such as *Elman* (115) and *Jordan* recurrent networks and time delay networks
- Self-organizing ANNs, such as the *Kohonen* self-organizing feature maps (116)
- Combined feed forward and self-organizing ANNs, such as the radial basis function networks (117)

A large variety of applications use different ANN types – typical are data mining, speech recognition, image processing and robot control.

4.4.4 Self-Organizing Feature Maps

Teuvo Kohonen developed self-organizing feature maps (SOM) (116) also known as topologically ordered maps or *Kohonen* self-organizing feature maps. *Kohonen* was inspired from studies on the self-organization characteristics of the human cerebral cortex, which showed that the motor cortex, somatosensory cortex, visual cortex and auditory cortex are represented by topologically ordered maps (113) (118).

The self-organizing feature map is a multidimensional scaling method. It allows a projection of a multidimensional input space to a discrete output space, which is usually a two-dimensional grid.

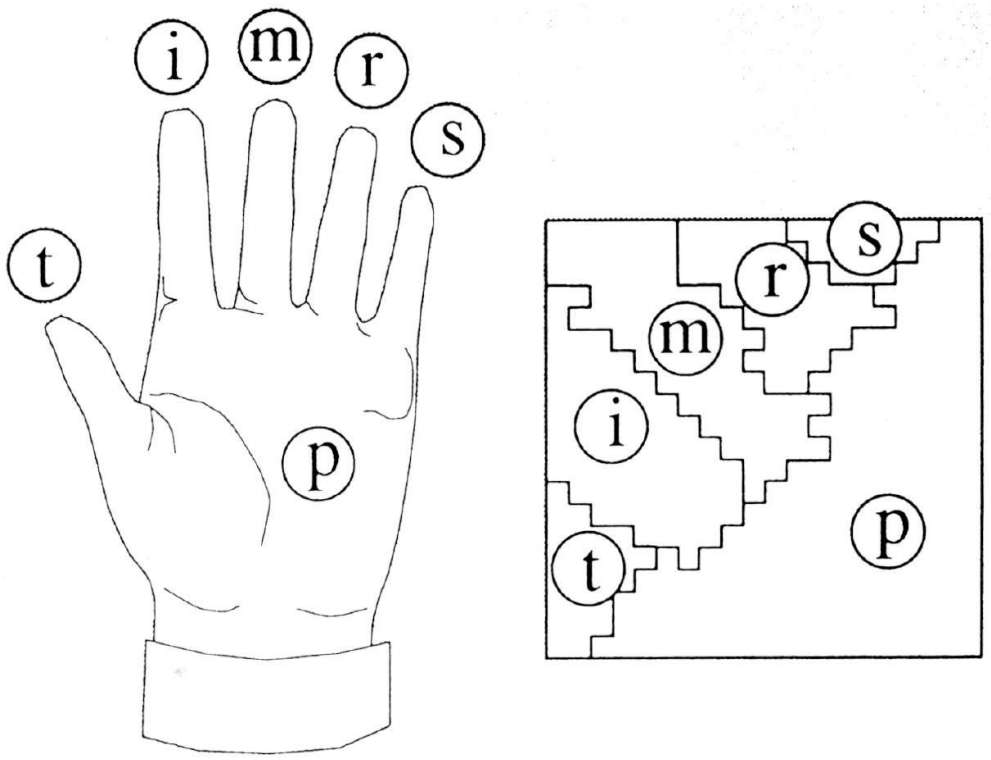


Figure 4.17: A self-organizing feature map projects a multidimensional input space onto a two-dimensional grid. (119)

This form of unsupervised learning is based on a competitive “winner takes all” principle. The self-organizing maps consist of compounds called nodes or neurons. Each neuron is associated with a weight vector of the same dimension as the input vector and a position in the output space. A two-dimensional hexagonal or rectangular grid is the usual arrangement of the neurons. In order to place an input vector from data space onto the map, the algorithm searches the neuron with the closest weight vector to the vector from data space. The algorithm then assigns the coordination of this neuron to the input vector from data space. See Figure 4.11 for a schematic structure of a self-organizing feature map.

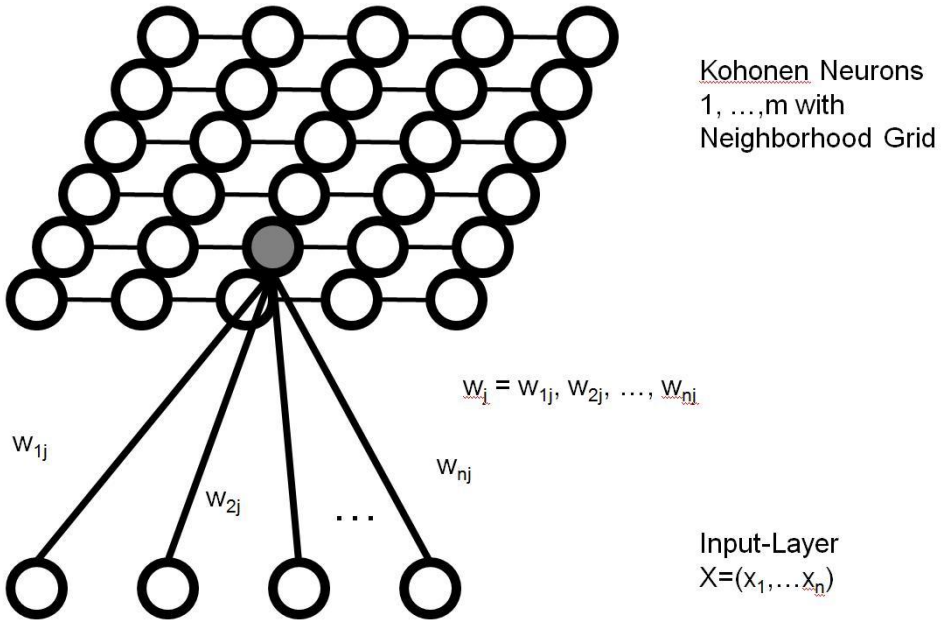


Figure 4.18: Schematic structure of a self-organizing feature map

4.4.5 Adaptive Resonance Theory

The *Adaptive Resonance Theory* (ART) (120) is a neural network architecture developed by *Stephen Grossberg* and *Gail Carpenter*. It is developed for an automatic unsupervised classification of open-categorical problems.

The basic ART system typically consists of a comparison field and a recognition field composed of neurons, a vigilance parameter and a reset module. The vigilance parameter has considerable influence on the system. A high vigilance parameter resulted in very detailed memories with many fine-grained categories, while a lower vigilance parameter produces memories that are more general and thus it produces fewer but more-general categories. The comparison field receives the one-dimensional input vector and transfers this vector to its best match in the recognition field. The best match is a single neuron, whose weights vectors most closely match those of the input vector. Proportional to the quality of this match, the non matching neurons of the recognition field receive or obtain a negative signal from the winning neuron, which inhibits their output accordingly. This allows each neuron to represent a category to which the input vectors are classified. After the input vector classification, the reset module is used to compare the strength of the recognition match to the given vigilance parameter. The training of the network starts if the strength of the match is higher than

the vigilance threshold. Otherwise, until a new input vector is applied, the winning neuron gets inhibited. During the search procedure, the reset function disables each recognition neuron one by one until a recognition match satisfies the vigilance parameter. If all matches are below the vigilance threshold, a new neuron is added to the recognition neurons. The new one gets adjusted towards the matching input vector. The Figure 4.19 shows the schematic flow of the ART 2A algorithm and Figure 4.20 shows the structure of the ART 2A neural network.

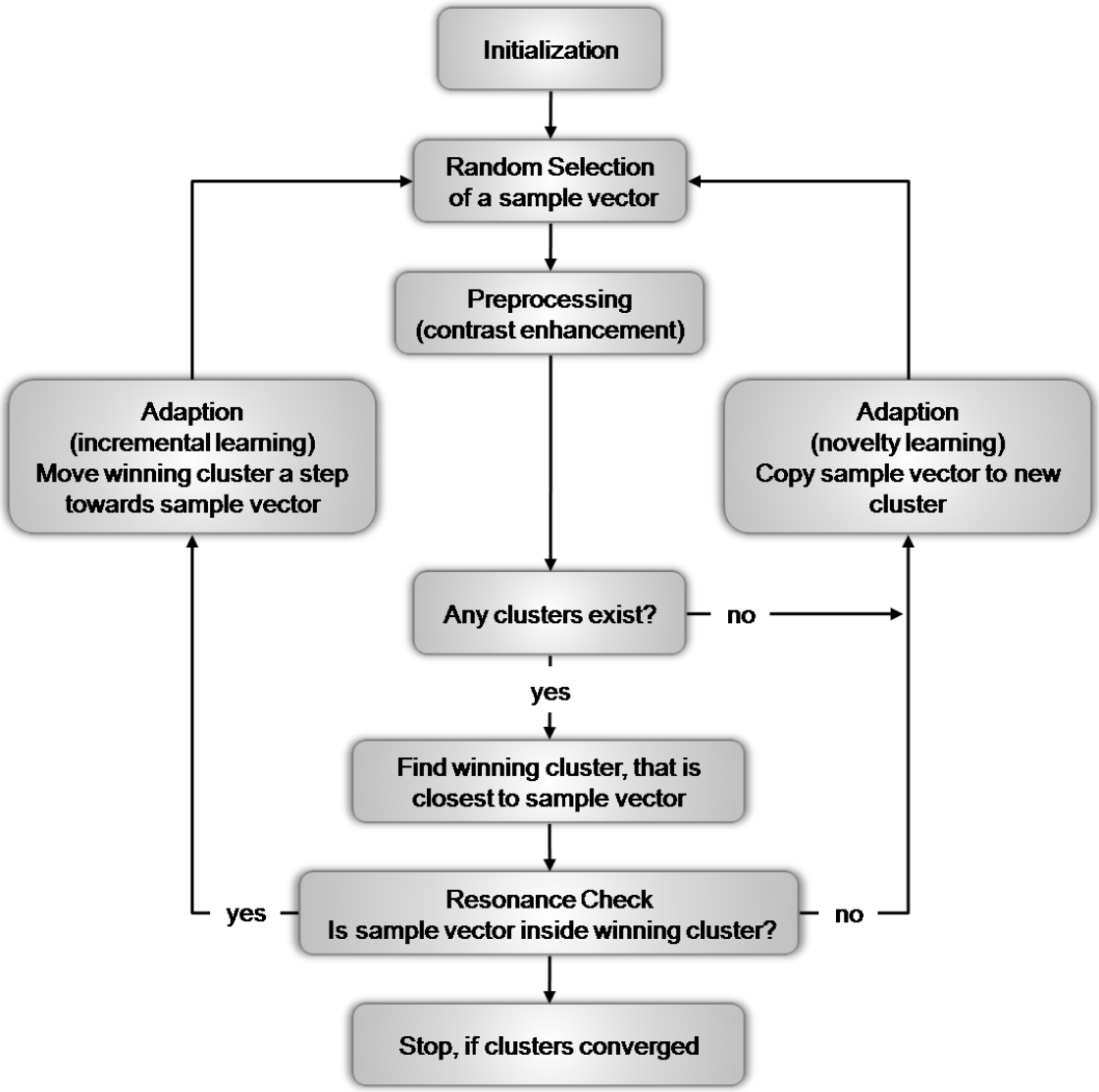


Figure 4.19: The schematic flow of the ART 2A algorithm

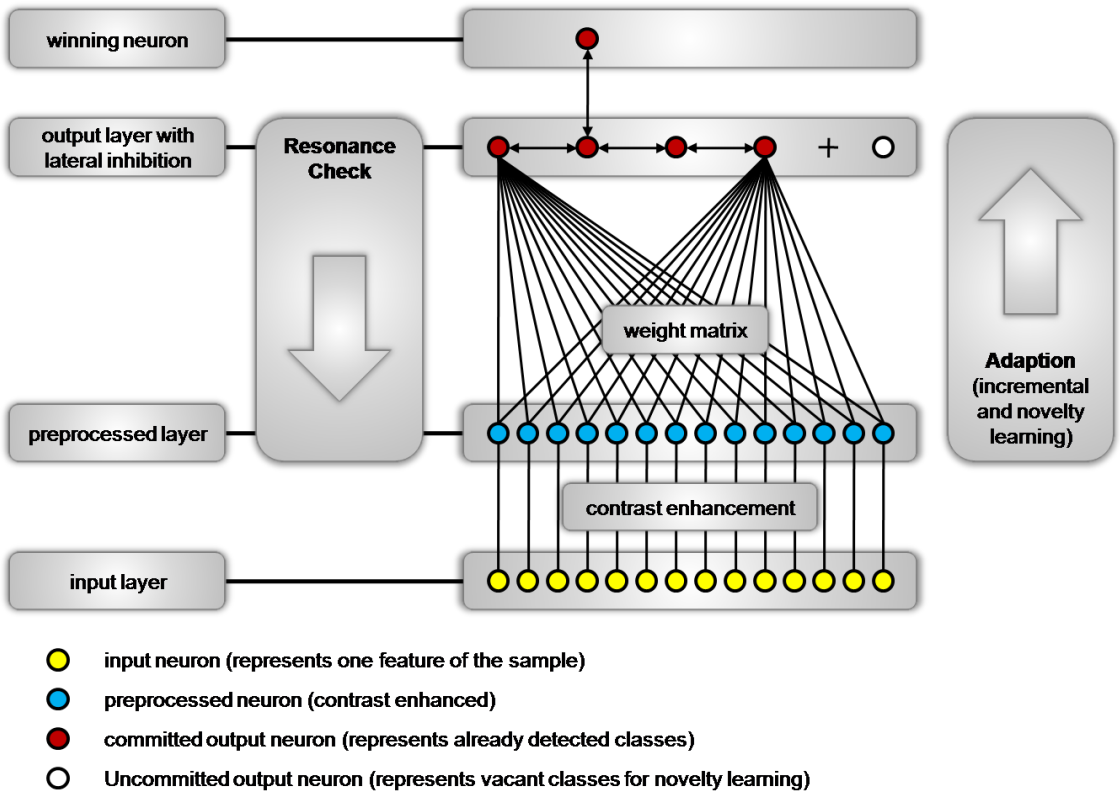


Figure 4.20: Schema of the ART 2A neural network with its different layers

The literature describes different types of ART algorithms:

- *ART 1* (120) (121) accepts only binary inputs and is the simplest variety of ART networks.
- *ART 2* (122) supports continuous inputs as extended network capability.
- *ART 2-A* (123) has a drastically accelerated runtime and is a streamlined form of ART-2. A detailed mathematical description of the ART 2-A algorithm is given in the Appendix of chapter 8.2.
- *ART 3* (124) simulates rudimentary neurotransmitter regulations of synaptic activity. Therefore, it incorporates simulated sodium and calcium ion concentrations into the system equations. This leads to a more physiologically realistic means of partially inhibiting categories.
- *Fuzzy ART* (125) add an implementation of fuzzy logic into the ART pattern recognition and enhanced its generalizability.

- *ARTMAP* (126) combines two modified ART-1 or ART-2 units to build a supervised learning structure.
- *Fuzzy ARTMAP* (127) uses the fuzzy ART units to build the ARTMAP, which results in a corresponding increase of efficacy.

5 Software and Methods developed during this work

This chapter contains information about the new methods and algorithms created specifically during this work.

5.1 The CDK-Taverna Plug-in

Taverna as an Open Source workflow environment with its extensible architecture (see chapter 4.1.1) forms the base for the *CDK-Taverna* plug-in. The goal of the plug-in is to add cheminformatics functionality to *Taverna*, using different extension points of *Taverna* to integrate all its functionality. The CDK provides the cheminformatics functionality used within this plug-in.

5.1.1 Plug-in Architecture

Taverna provides a list of *Service Provider Interfaces* (SPI) for the extension. The plug-in packages all implemented interfaces and adds its functionality to *Taverna*. A list of *Taverna's* available SPIs shows Table 2.

Table 2: Available *Taverna* SPI's (128)

Interface	**Description**
LocalWorker	A simple type of processor that doesn't require the full invocation infrastructure. It appears as a Local Java Widget in the Service Panel and works on a set of inputs to produce a set of outputs.
PerspectiveSPI	Defines a perspective, which describes a Workbench layout comprising of a set of UI components.
ProcessorActionSPI	Defines an action that can be performed on a Processor or set of Processors. These actions are collected and presented within the menu when right-clicking on that Processor in the Advanced Model Explorer, such as Processor configuration, metadata handling etc.
ProcessorInfoBean	Defines information about a Processor component.
RendererSPI	Provides Rendering capability for a particular type of data object, such as displaying a Graph or an Image.

ResultMapSaveSPI	Provides a customisation capability for saving a collection of DataThing objects that result from running a workflow, for example, saving to an Excel file.
ScavengerActionSPI	Provides actions capable of being operated upon a Scavenger object in the Service Panel, such as Scavenger configuration.
ScavengerHelper	Provides the ability to add a Scavenger to the Service Panel that requires additional information before being added, such as providing a WSDL url when adding a WSDL processor. This appears as a menu item when right-clicking on Active Processors in the Service Panel.
Scavenger	A Scavenger is created by the ScavengerHelper, but also can be defined as an SPI. If defined as an SPI it appears in the Service Panel under Local Services, for example, String Constant.
ScuflModelActionSPI	Defines an action that can be performed on the ScuflModel, and appear on the Advanced Model Explorer toolbar. Add Nested Workflow is an example.
WorkflowEventListener	Provides the ability to listen to events occurring during an enacting workflow. The LogBook plug-in uses this SPI to gather Provenance.
UIComponentFactorySPI	A factory for a UIComponentSPI. These toplevel UI components appear within a layout and can be added to a workbench layout as part of a perspective. The Service Panel, and the Advanced Model Explorer are UIComponentSPI's that are discovered through its own UIComponentFactorySPI.

The *CDK-Taverna* project implements some of these extensions, which lead to an integration of the CDK functionality as a so-called *Local Worker*. The execution of this worker is performed on each local machine. All workers provided by the *CDK-Taverna* plug-in implement the *CDKLocalWorker* interface. The detection of this interface from each worker performs an extension of *Taverna's Scavenger SPI*, the *CDKScavenger*. The *AbstractCDKProcessorAction* implements *Taverna's ProcessorActionSPI* for adding user interfaces to specific worker. The use of this SPI allows the addition of, for example, file chooser dialogs for workers such as the file reader or writer. Figure 1.1 shows the schema of the *CDK-Taverna* plug-in architecture.

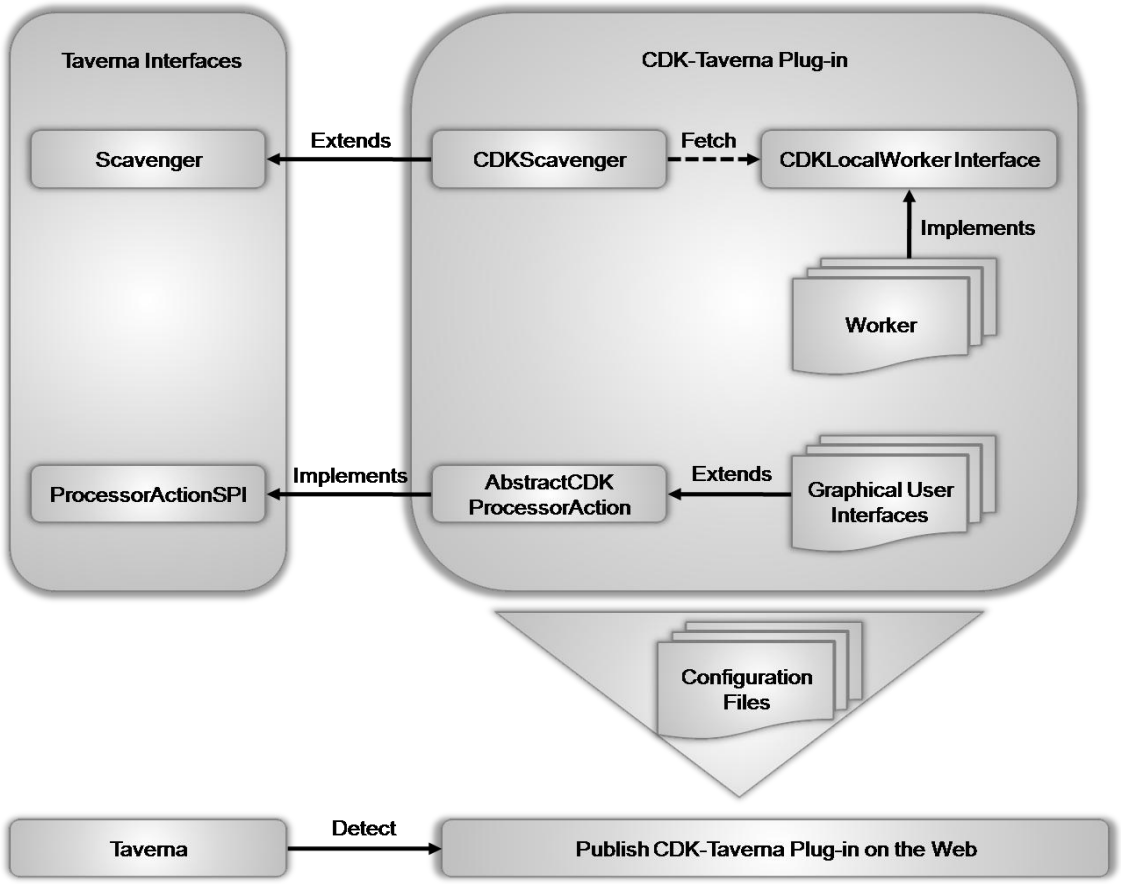


Figure 5.1: The architecture of the *CDK-Taverna* plug-in

Maven (129), a software management tool, packages the entire classes and configurations file of the *CDK-Taverna* plug-in. This package is stored in a public *Maven* repository. Besides this package, two configuration files describing the plug-in are needed for *Taverna* to find, download and install it on each local *Taverna* installation. The first configuration file contains a list of available plug-ins with their distinct file names. The second configuration file contains the description of the plug-in with information about the name, a description, the version number, the target version number and the repository location of the installable plug-in. The Appendix 8.2 contains a copy of the two configuration files from the *CDK-Taverna* plug-in.

To install the *CDK-Taverna* or any other plug-in in *Taverna* the user needs to add a URL of the configuration files. For the *CDK-Taverna* plug-in, the configuration files are publicly available at the project website: http://www.cdk-taverna.de/plugin.

5.1.2 Functional Overview

The *CDK-Taverna* plug-in provides a large variety of about 160 different workers. Table 3 contains a task oriented overview of these workers. The appendix at chapter 8.4 contains a list of all workers with a short description of their functionality.

Table 3: Overview of *CDK-Taverna* workers

Group	Number of Workers
File I/O	15
Database I/O	7
Molecular descriptors	42
Atom descriptors	27
Bond descriptors	6
ART2A classifier and result analysis worker	10
SimpleKMean and EM clusterer (uses Weka)	3
SMILE tools	2
Inchi Parser	2
Miscellaneous	50

5.1.3 Pgchem::Tigress as Database Backend

The *CDK-Taverna* plug-in contains different workers to use a *PostgreSQL* database with the *PGChem::Tigress* cheminformatics cartridge (see chapter 4.3.1) as a backend:

- Writing of molecules into the database
- Loading of molecules from the database
- Performing substructure searches on the database
- Adding QSAR values as properties of different molecules
- Loading vectors of QSAR values of specific molecules

Table 4 shows the main table of the database backend, which stores the molecules and some molecule related properties. For the molecule column of this table a GiST index is created, which allows a fast substructure search.

Table 4: This schema describes the molecules table for the database backend. The left column contains the names of the different table columns whereas the right column represents the data type of each column.

molecules	
id (PK)	serial
molecule	molecule
comment	text
originalid	character varying
origin	character varying

Besides the molecules table, a table for the QSAR values is created for the work of this project. This table contains the molecule identifier and a column for each value calculated by a QSAR descriptor.

For the configuration of each database worker the *CDK-Taverna* plug-in provides a user interface. This user interface allows the setting of a database URL, a username, a password and the SQL query (see Figure 5.2).

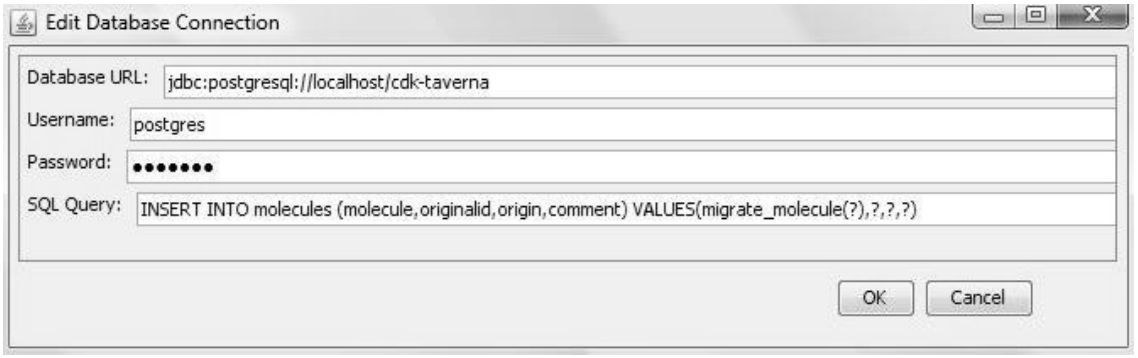


Figure 5.2: This user interface allows the editing of the database configuration.

5.1.4 Iterative Handling of Large Datasets

During a typical task of a cheminformatics workflow many thousands of molecules need to be processed. An example workflow would be the sorting of molecules into aromatic and non-aromatic molecules. The design of the current *Taverna* version provides directly no possibility to perform things such as for- or while loops on a set of workers. To be able to handle many thousands of molecules the *CDK-Taverna* plug-in uses more or less a hack to provide an iterative handling of large datasets. Figure 5.3

shows such a workflow. For providing an analogue to a for-loop this workflow contains four important workers.

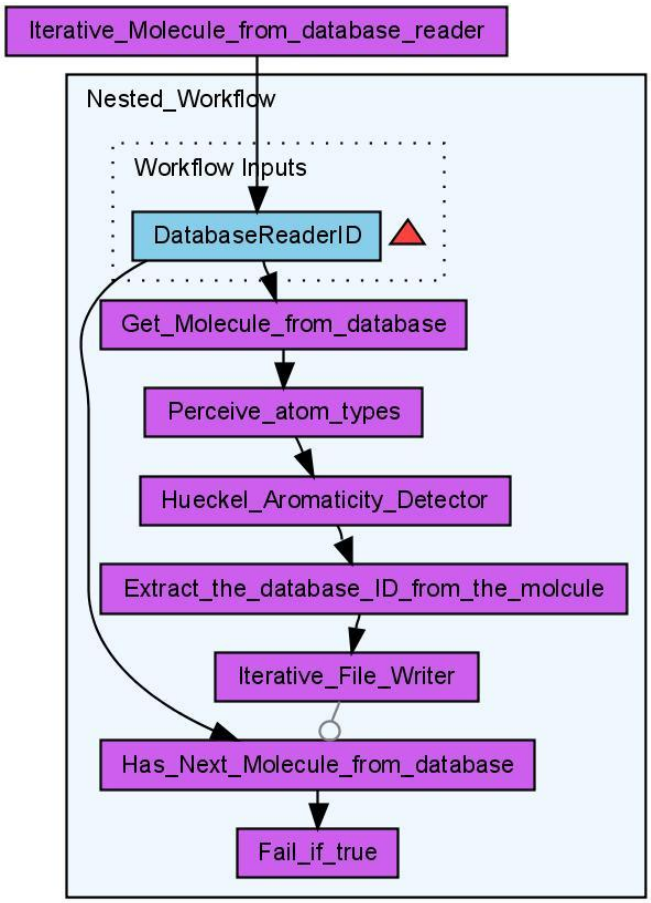


Figure 5.3: This workflow shows an iterative loading of molecules from a database. After the perception of the atom types, each molecule goes through the detection of the *Hückel* aromaticity. At the end all identifier of aromatic molecules are written to a text file. (130)

The *Iterative_Molecule_From_Database_Reader* initialises and configure a database connection. This includes a SQL query to select distinct molecules form the database, which contains the SQL keywords *LIMIT* and *OFFSET*. These keywords allow the splitting of the database result set into subsets. The initialised database connection is stored into a registry and the output of the first worker is a link to this database connection.

The following workers are all part of a nested workflow, which allows a specific retry configuration. The *Get_Molecule_From_Database* worker uses the stored database

connection to load one result subset each time. After the loading of a subset of molecules, these can be processed in various ways. The example workflow perceives the atom types, before each molecule goes through the detection of the *Hückel* aromaticity process. As result of this workflow all identifier of aromatic molecules are written to a text file.

The *Has_Next_Molecules_From_Database* worker starts its work after the *Iterative_File_Writer* has finished its job. This worker checks whether the database query contains more subsets or not. If the query contains more subsets will the output of this worker be the value *true*. Otherwise will the output of this worker be the value *false*.

The *Fail_If_True* worker throws an exception if it gets as input the value *true*. This worker is configured as critical, which allows a failure of the whole nested workflow. The retry configuration of the nested workflow allows the rerun of the nested workflow. This time the *Get_Molecule_From_Database* worker will load the next subset of the database query. This construction of workflows enables *Taverna* to perform a for-loop analogue.

5.2 Reaction enumeration

Markush structures are generic chemical structures, which contains variable patterns such as “Heterocyclic”, “Alkyl” or “R = Methyl, Isopropyl, Pentyl”. Such structures are commonly used within patents for describing whole compound classes and are named after *Eugene A. Markush* who described these kind of structures firstly in his US patent in the 1920s. In the process of reaction enumeration, *Markush* structures are used to design generic reactions. These reactions are useable for the enumeration of large chemical spaces, which includes the generation of chemical target libraries. The result of the enumeration is an important base for patents and for the *High Throughput Screening* (HTS) laboratories in drug development. Within HTS experiments, a couple of years ago the enumeration of large libraries was driven to about O(10.000-100.000) molecules in one run. These large and often unspecific HTS libraries did not prove to be successful in general. Today the enumeration is based on more targeted libraries with a reduced size of O(1.000) molecules.

For a reaction enumeration, a given reaction contains different building blocks, which are needed for the enumeration (see Figure 5.4). Each reactant of the reaction represents a building block. A scientist then selects a number of molecules for each reactant and the reaction enumeration creates a list of all possible products. The list of products then passes a virtual screening before at last a scientist decides which products will be synthesized.

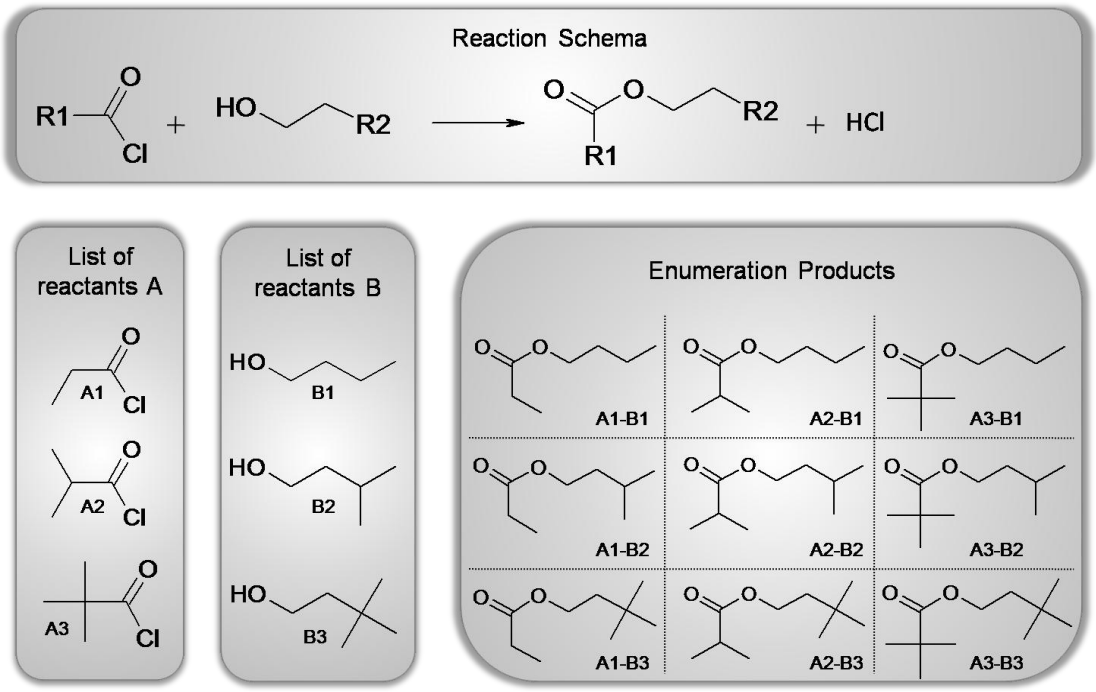


Figure 5.4: This reaction enumeration example contains two building blocks. For each building block, a list of three reactants is defined. This enumeration results in nine different products.

5.2.1 Reaction Enumeration algorithm

For showing the different steps of the algorithm the reaction from Figure 5.4 (Reaction Schema) and two reactants A1 and B1 (see Figure 5.4) are used. These inputs would lead to the result molecule A1-B1 from Figure 5.4.

The first step of the algorithm deletes the pseudo atom *R1* from the first reactant of the reaction (see Figure 5.5).

Figure 5.5: In the first step of the reaction enumeration, the pseudo atom *R1* is deleted.

In the next step, the rest of the first reactant from Figure 5.5 is used to perform a substructure search on the reactants (see Figure 5.6).

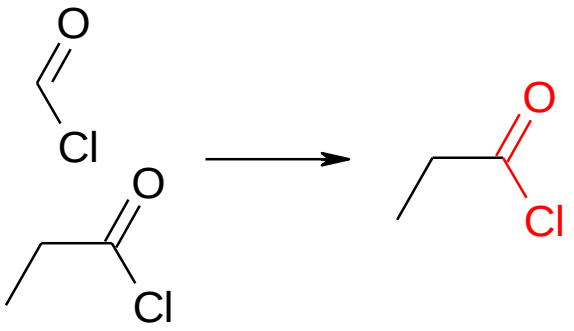


Figure 5.6: The second step performs a substructure search on the reactants.

The third step removes the substructure found at Figure 5.6 from the reactant and sets a flag on the atoms, which contains broken bonds (see Figure 5.7).

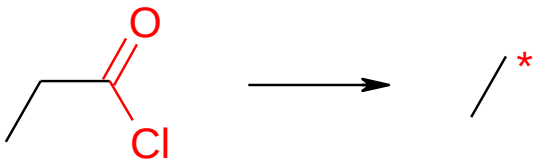


Figure 5.7: The third step removes the substructure from the reactant.

The following step removes the same pseudo atom *R1* from the product of the reaction (see Figure 5.8). This has to be the same pseudo atom as it was in the first step. Again all atoms set a flag, which contains broken bonds from the removing process.

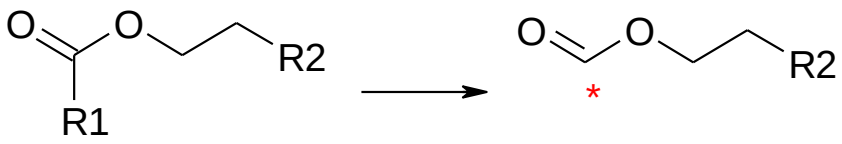


Figure 5.8: This step removes the pseudo atom *R1* from the product of the reaction.

At this point two molecule fragments are available which contains marked atoms. The next step combines the fragments from Figure 5.7 and Figure 5.8 by the creation of a new bond (see Figure 5.9).

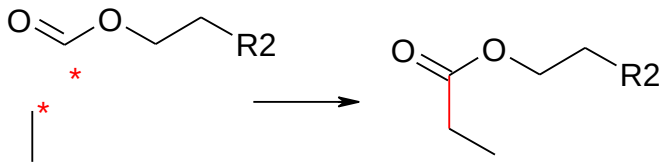


Figure 5.9: In this step, the two molecule fragments are combined.

In this example, the product molecule of the reaction contains two pseudo atoms. In the further steps, the second still remaining pseudo atom is replaced in the way as the first, described in the steps above. This leads to the product A1-B1 of Figure 5.4. After the recombination of the reactant fragments with the product fragment the creation of new 2D coordinates is necessary. This example shows only the creation of one product but normally a list of different reactants leads to many different products (see Figure 5.4).

5.3 GNWI.CIP

GNWI.CIP (Computational Intelligence Packages) is an (LGPL) Open Source project currently under development and financed by the *GNWI – Gesellschaft für naturwissenschaftliche Informatik mbH* (131). *GNWI.CIP* is a set of packages based on *Mathematica* (132), a commercial software system for mathematical computation and visualisation. *Mathematica* is extensible with a *C*-like programming language and may be interfaced from the *Java*(133) as well as the *Microsoft.Net* (134) platform. *GNWI.CIP* comprises packages for scientific data transformation, graphical visualisation and linear as well as non-linear data fitting and machine learning.

GNWI.CIP methods can be accessed via *Taverna* workers on the *Java* platform. Parameters of the method calls to *GNWI.CIP* are passed as string variables. Return values are again strings or graphics primitives (JPEGs, metafiles etc.). A *Taverna* worker encapsulates one or more *GNWI.CIP* method calls and handles further treatment of return values, e.g. graphics display.

Since *GNWI.CIP* is currently under development the following results (see section 6.3) are proofs of concept that open the perspective of an integration of high-level "experimental mathematics" into the *Taverna* based scientific pipelining. This can be a crucial advantage for Chemoinformatics method development and validation: As an example a neural network or support vector regression method based on *Mathematica* can be realized with a few lines of high-level code in contrast to hundreds or thousands of lines of error-prone code in traditional programming languages such as C or Java.

6 Results

This chapter contains the results of this work. It demonstrates the created workflows, the possibilities of these workflows and the results of workflows applied to real world research problems.

6.1 Validation of Software and Databases

The validation of software and databases was one of the mayor tasks of this project, especially the validation of the CDK against the content of chemical databases in the public domain such as *ChEBI* and *ChEMBL*. *ChEMBL* will be available to the public at the beginning of 2009 and will provide an open access to large-scale drug discovery data. (27) The curation of databases is a resource consuming task and within the life sciences an ongoing discussion with the headline “Curation Crisis”. In times where the extraction of chemical information from different content such as articles, books and websites is automatically performed by tools such as *OSCAR* or *OSRA* (135) the validation of the data is an important step. The beginning of this chapter shows how to store chemical structures into a database and how to review them for further processing. The end of this chapter shows results of atom typing analysis using the CDK atom typing detection performed on different database.

6.1.1 Storage of Molecules in a Database

For writing molecules into a database different possible workflows are thinkable depending on the format in which the molecules are stored before they get loaded into the database and on the number of molecules.

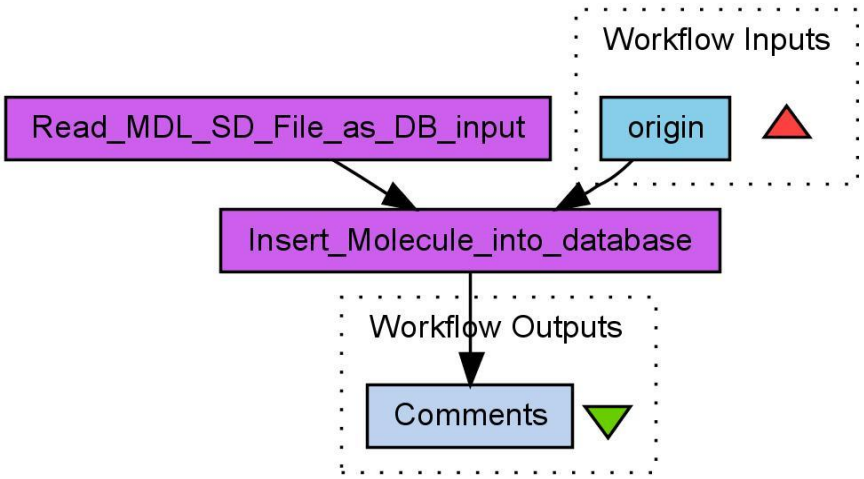


Figure 6.1: This workflow stores molecules into a database. The molecules originally are stored in a MDL SD file. (136)

This example workflow loads molecules into a database from a MDL SD file (see Figure 6.1). The *Read_MDL_SD_File_as_DB_input* worker loads the MDL SD file and outputs the content of the file as plain text. The *Insert_Molecule_into_database* worker gets the content of the MDL SD file and the value of the workflow input called *origin*. The workflow input is used to add an origin to each molecule which is loaded into the database. The workflow output contains information about the success of the database insert process. This workflow is usable for small to mid-sized MDL SD files up to ten thousands of molecules. The limitation on the size of files loaded by this workflow is caused only by the available memory of the local computer.

For loading large datasets into the database the iterative handling described in chapter 5.1.4 is preferred. The workflow shown in Figure 6.2 allows the loading of large datasets and was used to load an MDL SD file with over 400,000 molecules in 3 hours on a year 2004 dual core Intel Xeon workstation.

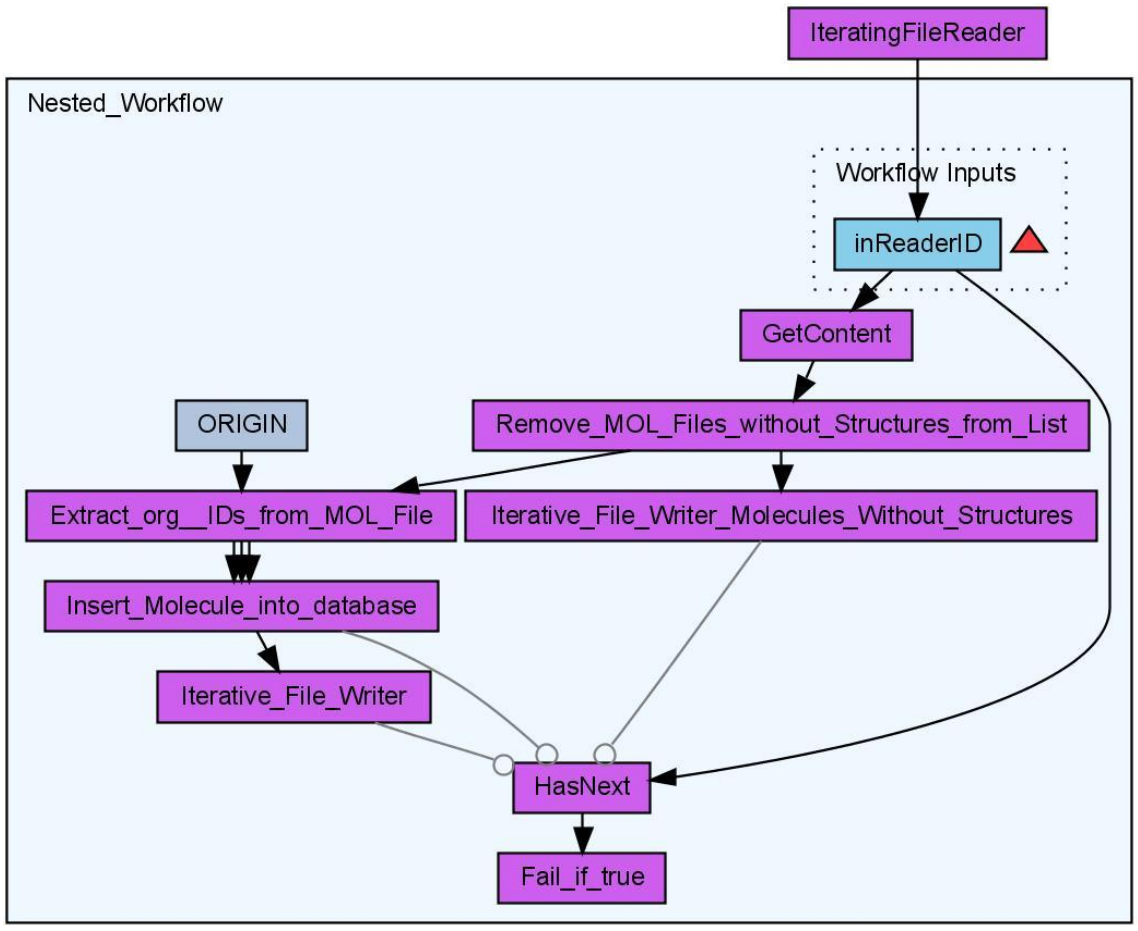


Figure 6.2: This workflow loads datasets into the database using an iterative approach. (137)

After loading a set of molecules from the file with the *GetContent* worker, the *Remove_Mol_Files_without_Structures_from_List* worker removes the molecules without structures. These molecules normally contain only an identifier and are relicts of the creation of the MDL SD file from other databases. The *Iterative_File_Writer_Molecules_Without_Structures* worker writes the removed content to a text file to store the information about the removing process. The *Extract_org_IDs_from_MOL_File* worker processes the valid content of the MDL SD file. Beside the file content, this worker gets the origin of the data as input and extracts the original molecule identifier from the content of each molecule if it is available. All the information, the molecule, the origin and the original identifier than gets uploaded to the database by the *Insert_Molecule_into_database* worker. The *Iterative_File_Writer* worker writes information about the success of the loading of molecules into the

database to a file. Every retry of the nested workflow processes 100 molecules loaded from the MDL SD file.

6.1.2 Retrieval of Molecules from a Database

After the storing, the loading of molecules from a database is another important task for a Cheminformatics toolkit. The *CDK-Taverna* plug-in provides again two different ways for loading molecules from a database. The first workflow shown in Figure 6.3 fetches the molecules from a database with the *Get_molecules_from_database* worker. With the loaded molecules, the *Create_PDF_File_with_2D_Structures* worker creates a PDF document, which contains a table showing the 2D structures of each molecule.

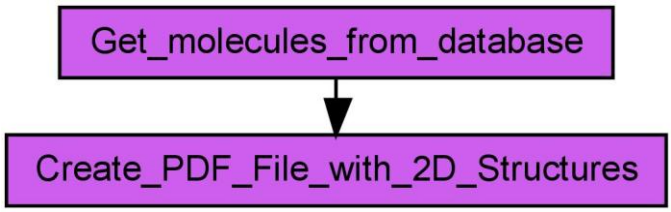


Figure 6.3: This workflow creates a PDF document, which shows the molecules loaded from the database. (138)

This kind of workflow could be used for the loading of hundreds or thousands of molecules. The second type of workflows for loading molecules from databases is designed to load from many thousands up to millions of molecules. Again this workflow uses the iterative approach for the creation of workflows, which is described in chapter 5.1.4. The last chapter already showed an example of an iterative loading of molecules from a database (see Figure 5.3).

6.1.3 Import of the ChEBI Database

The *Chemical Entry of Biological Interest* (ChEBI) (3) database currently contains over 13,000 structures, which are downloadable on the Web as a *tabular separated values* (tsv) file. This file represents a data table and contains (beside the molecular structure in different formats), an identifier, a compound identifier, the type of entry, and the dimension of the structure. The available data format created a need for a specific worker to extract the information from the tsv files (see Figure 6.4).

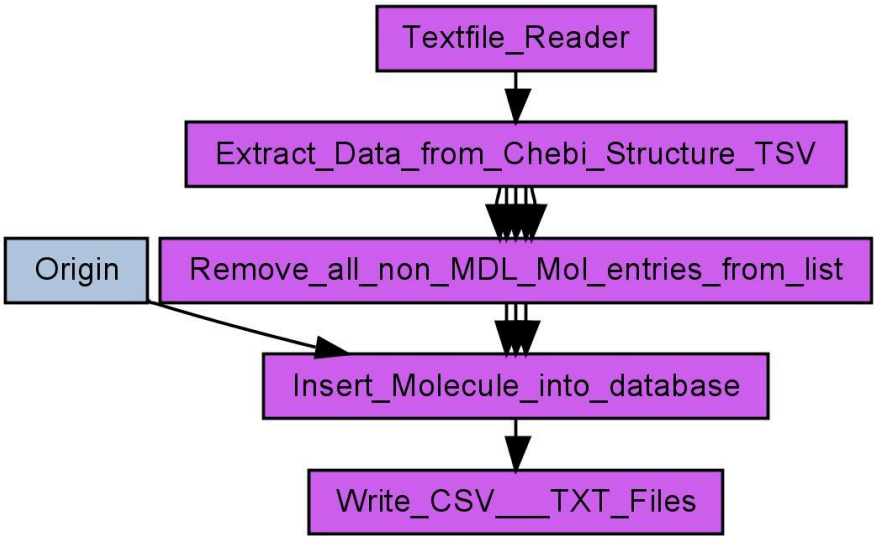


Figure 6.4: This workflow extracts the ChEBI dataset and inserts the molecules into a local database. (139)

The *Extract_Data_from_Chebi_Structure_TSV* worker gets the content of a *ChEBI* tsv file and separates each column of the data table. The *Remove_all_non_MDL_Mol_entries_form_list* worker removes all molecules which are not stored in the MDL mol format before the *Insert_Molecule_into_database* worker uploads the extracted molecules into a database.

6.1.4 Validation of CDK Atom Types

The calculation of different physiochemical properties is only feasible if the atom types are detected correctly for each molecule. The CDK contains methods for the detection of defined atom types. These methods allow also the identification of unknown atom types. With the help of these methods a validation of the CDK and of the processed data is possible. The detection of an unknown atom type by the CDK indicates whether the CDK is missing this specific atom type or the molecule contains chemically nonsensical atom types such as five bonded nitrogen's. The following workflow is used to detect such unknown atom types.

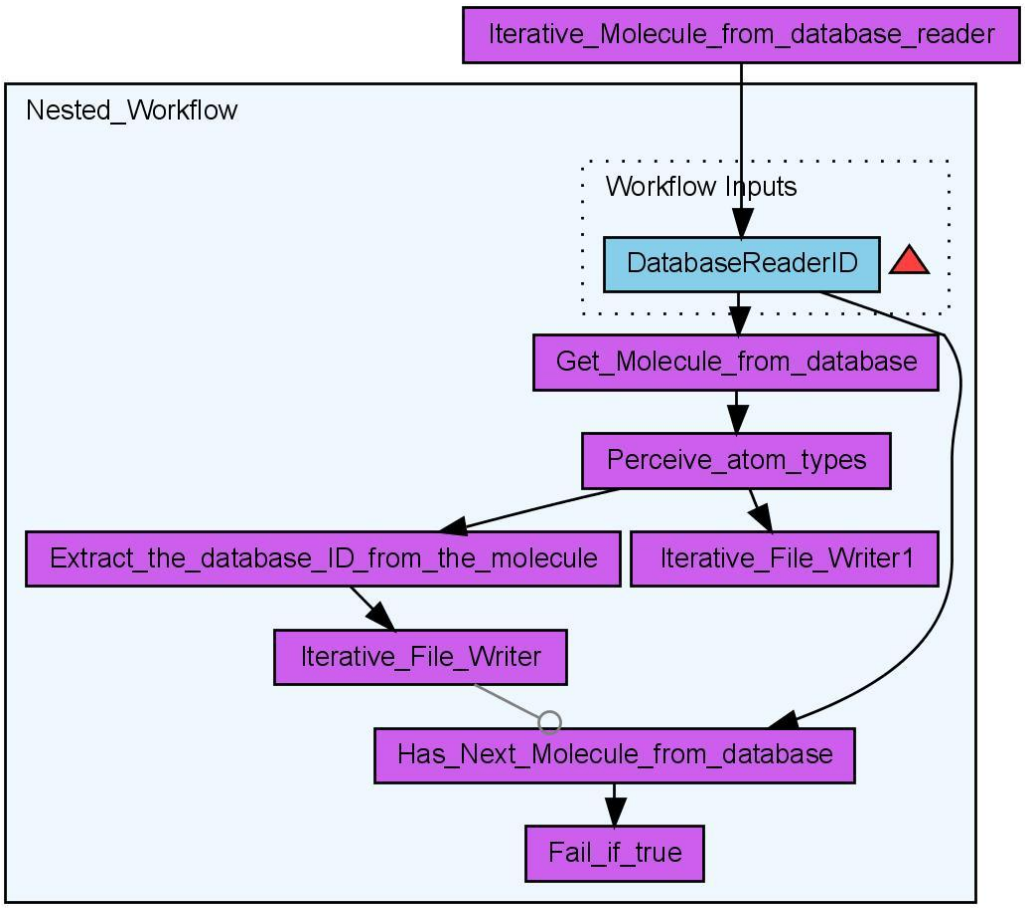


Figure 6.5: This workflow shows the iterative loading of molecules from a database and searches for molecules with (for the CDK) unknown atom types. (140)

Figure 6.5 shows another workflow which uses an iterative loading of molecules from a database to enable the processing of thousands of molecules. This workflow searches molecules which contain unknown atom types. Therefore it tries to perceive the atom types of each molecule using the *Perceive_atom_types* worker. Than the *Extract_the_databaseID_from_the_molecule* worker extracts the database identifiers from the molecules with unknown atom types. As results of this workflow two text files are created, the first contains the identifier of the molecules and is created by the Iterative_File_Writer. The second file contains information about which atom of which molecule is unknown to the CDK.

A second workflow is used to visualise the outcome. The workflow shown in Figure 6.6 loads one of the result text files through the *Textfile_Reader* worker and the

Analyse_Atom_Typing_Result worker creates a PDF document which visualises the outcome of the atom type perception.

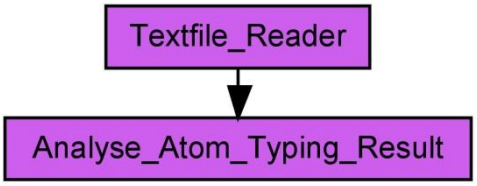


Figure 6.6: Workflow for analysing the result of the workflow shown in Figure 6.5 (141)

During the work for this project three different databases are used for the validation purposes. The *ChEBI* database, the *ChEMBEL* database and, through the cooperation with *InterMed Discovery*, a *Natural Product* lead-discovery company, a processing of the *Chapman & Hall Chemical Database* was possible. On the following pages, three diagrams, one for each database, show the distribution of the detected unknown atom types. A summary of the result is shown in Table 5

Table 5: The table shows the summary of the validation of the CDK atom types while processing the different databases.

Database	**ChEBI**	**ChEMBL**	**Chapman & Hall**
Number of Molecules	12367	440036	187502
Number of Molecules with Unknown Atom Types	1035	3067	1350
% of Molecules with Unknown Atom Types	8.37 %	0.70 %	0.72 %
Total Number of Unknown Atom Types	2414	5520	1817

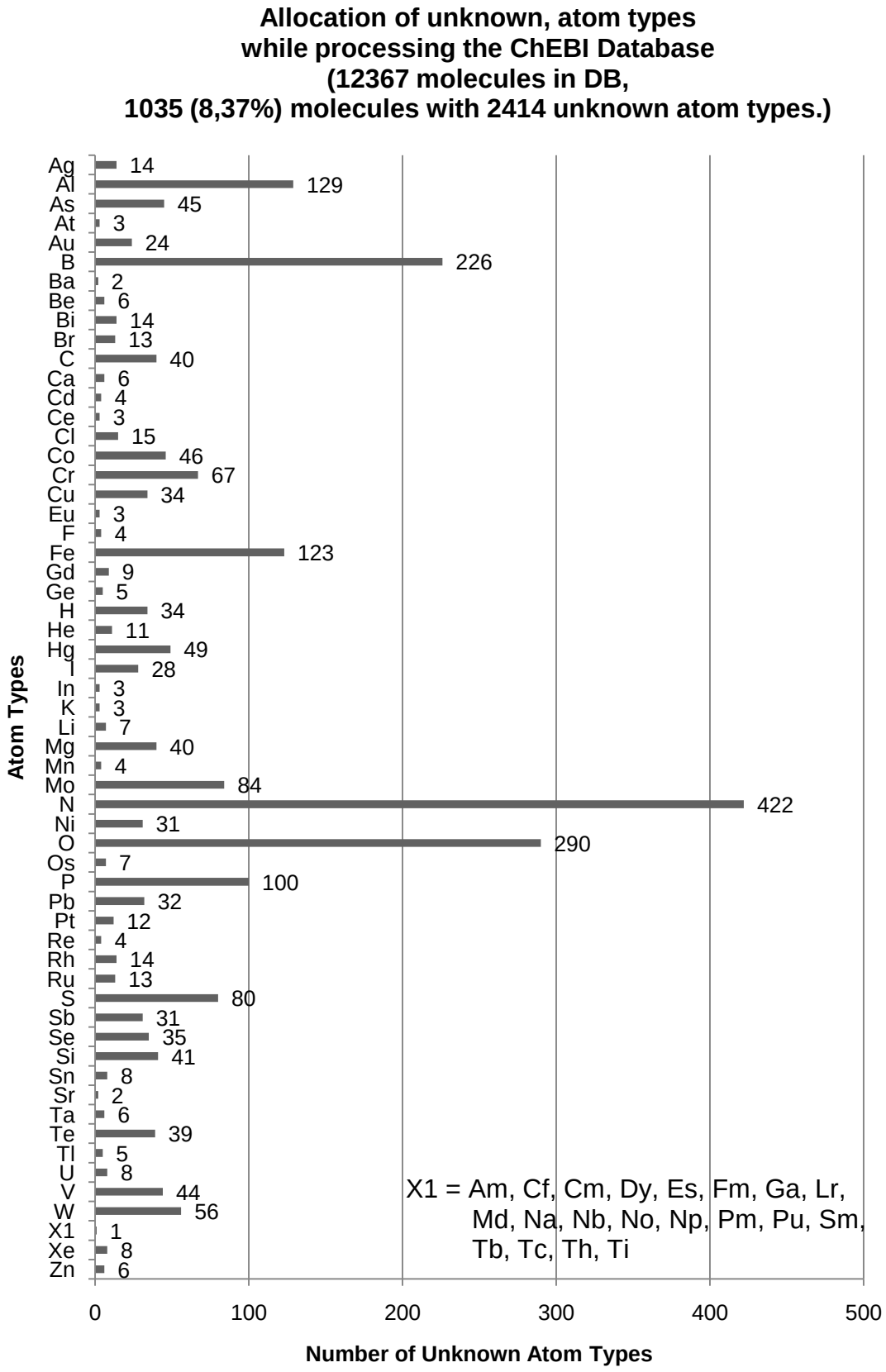


Figure 6.7: The diagram shows the allocation of the unknown atom types detected during the analysis of the *ChEBI* database.

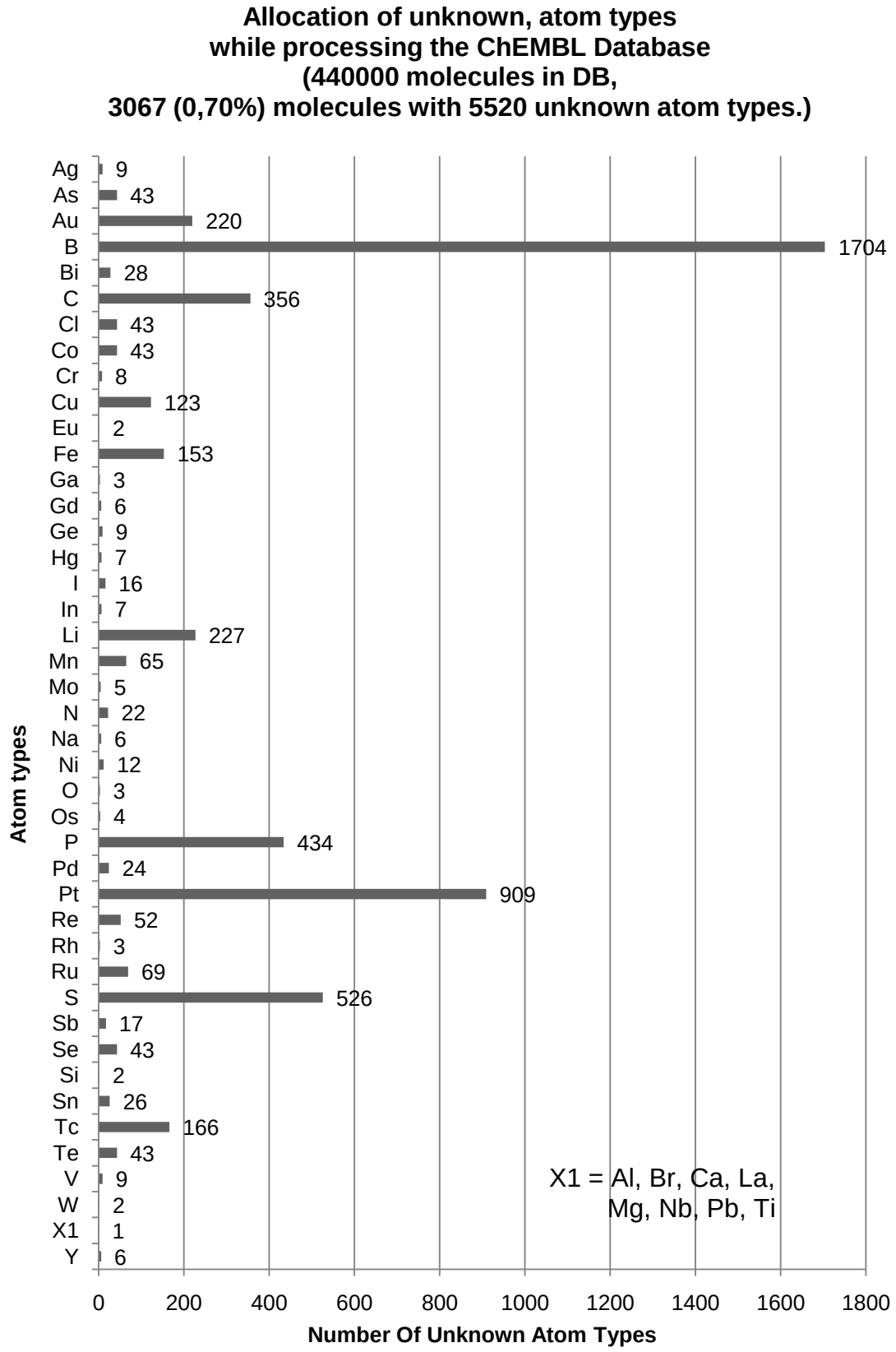


Figure 6.8: The diagram shows the allocation of the unknown atom types detected during the analysis of the *ChEMBL* database.

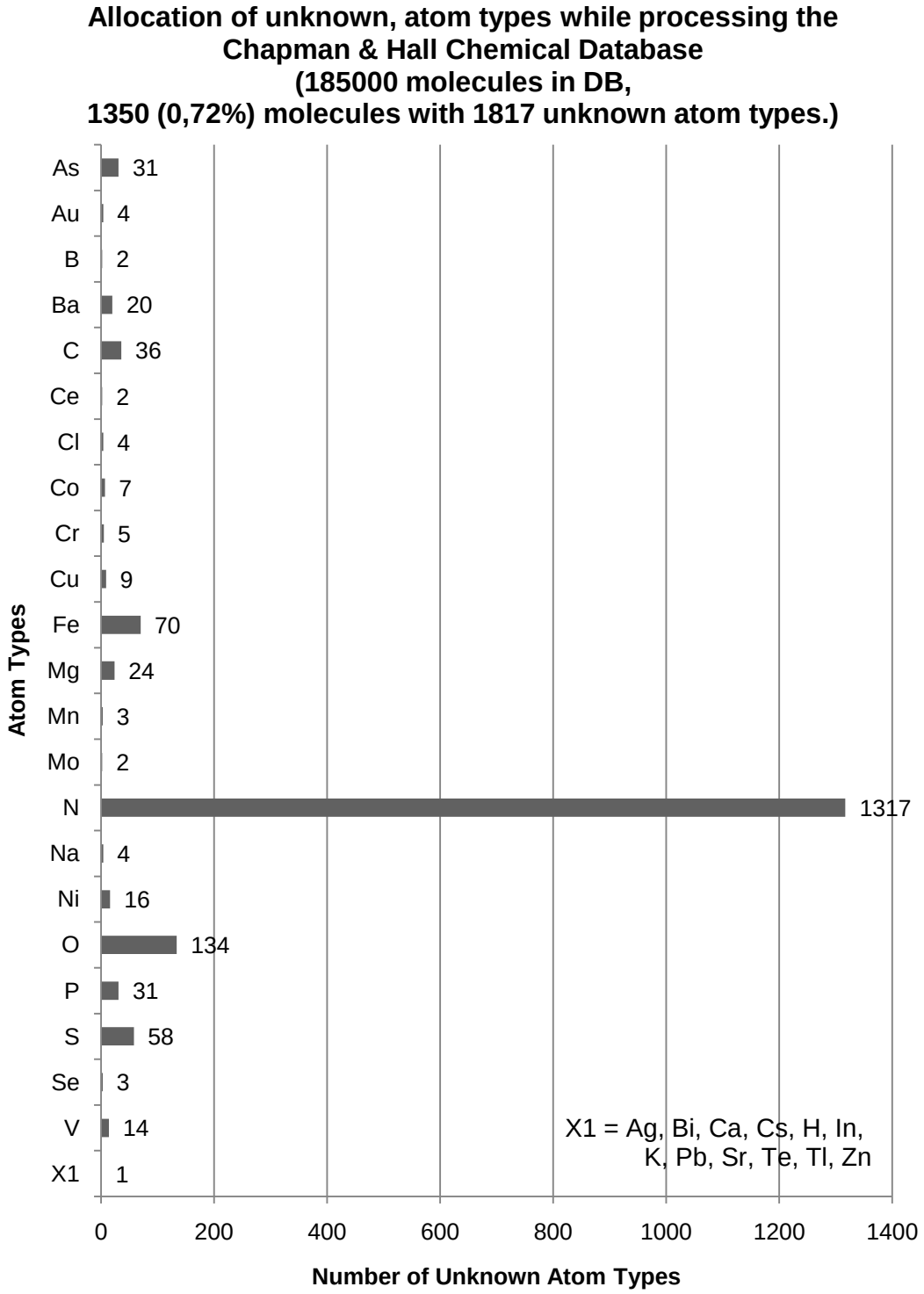


Figure 6.9: The diagram shows the allocation of the unknown atom types detected during the analysis of the *Chapman & Hall Chemical Database*.

The analysis of the different databases shows that the CDK atom types match the atom types of the database quite well. The *ChEMBL* (0.70 %) and the *Chapman & Hall Chemical Database* (0.72 %) contain only a few molecules which contain unknown atom types. Only the *ChEBI* database (8.37 %) has a quite large number of unknown atom types.

The results of this analysis obviously show that the CDK is unsatisfactory regarding atom types of heavier atoms which lead to problems with coordination compounds containing metals.

The quite large number of unknown boron atom types within the *ChEBI* (226 / 9.4 %) and especially the *ChEMBL* (1704 / 30.9 %) database indicates a lack of missing atom types within the CDK. Although only a small number of molecules contains these boron atom types, 33 in *ChEBI* and 144 in *ChEMBL* (see Figure 6.10). On the other hand the *Chapman & Hall Chemical Database* only contains two unknown boron atom types. The reason for the low number is caused through the absence of molecules containing boron.

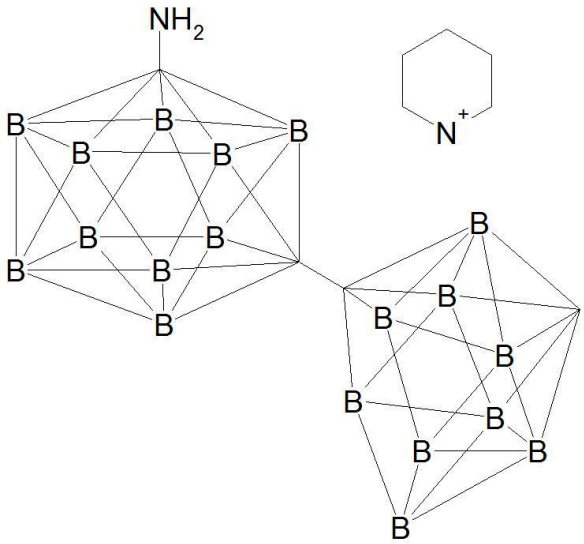


Figure 6.10: This molecule is one example, which contains multiple, for the CDK, unknown boron atom types.

A closer look on the missing nitrogen atom types from the *ChEBI* and *Chapman & Hall Database* shows that these databases heavily use five bonded nitrogen's, for example, to code a nitro functional group (see Figure 1.2). The CDK does not contain an atom type

for this coding of nitrogens. This shows one of the mayor problems of data curation because different databases use different coding standards for molecular structures. There is an ongoing discussion on this within the CDK community as to whether this kind of atom types should be supported or not. The five bonded nitrogen's are chemically not correct but widely used within different databases. A solution for such a problem would be to detect these kinds of atom types and perform a curation to a chemically correct molecular structure. It would be a major step for the public databases to provide valid structures within their tables. The difficulty with this task is the changing of chemical structures and all the consequences of this change such as a change in the 2D coordinates.

6.2 Reaction Enumeration Workflow

The implementation of the reaction enumeration algorithm described in chapter 5.2 is used to create the workflow shown in Figure 6.11.

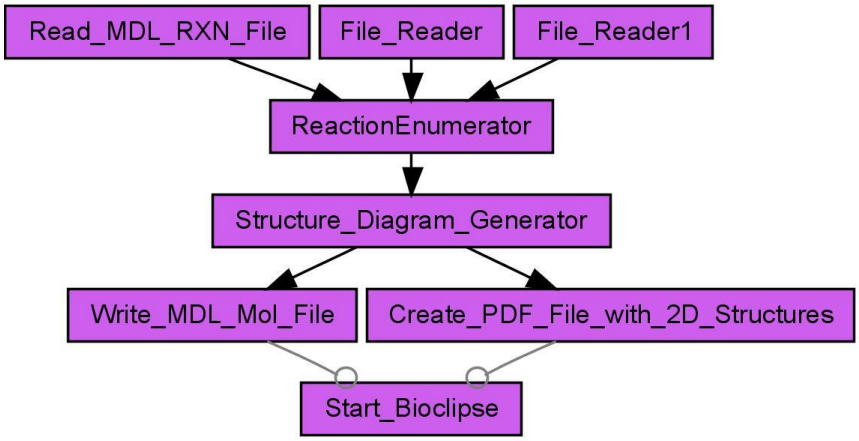


Figure 6.11: This workflow performs a reaction enumeration. Consequently it loads a generic reaction from a MDL rxn file and two reactant lists from MDL SD files. The products from the enumeration are stored as MDL mol files. Besides these files a PDF document showing the 2D structure of the products is created. At the end Bioclispse will start up to inspect the results. (142)

Within this workflow the *Read_MDL_RXN_File* worker loads a generic reaction from a MDL reaction file and the *File_Reader* worker loads the list of reactants from MDL SD files. The *ReactionEnumerator* worker performs the enumeration based on the generic reaction and the list of reactants. For each enumerated product the

Structure_Diagram_Generator worker creates new 2D coordinates before products are stored as MDL MOL files by the *Write_MDL_Mol_File* worker. For a visual inspection of the result the enumerated structures are drawn as chemical spreadsheets within a PDF document. The *Start_Bioclispe* worker is used to start Bioclipse (101) (see Figure 6.12) for an inspection of the created results.

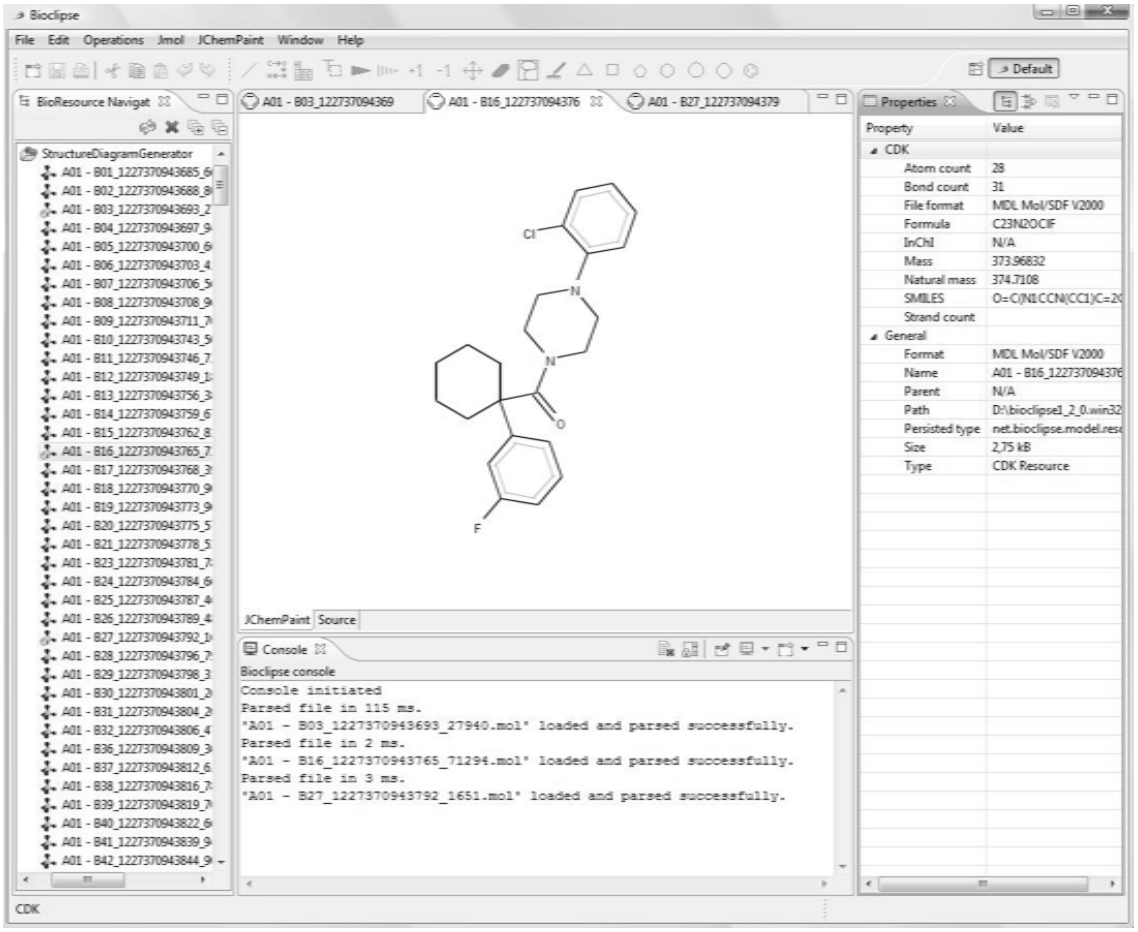


Figure 6.12: The Bioclipse platform is written in Java and usable for chem- and bioinformatics. Here used to inspect the results created by the reaction enumeration workflow.

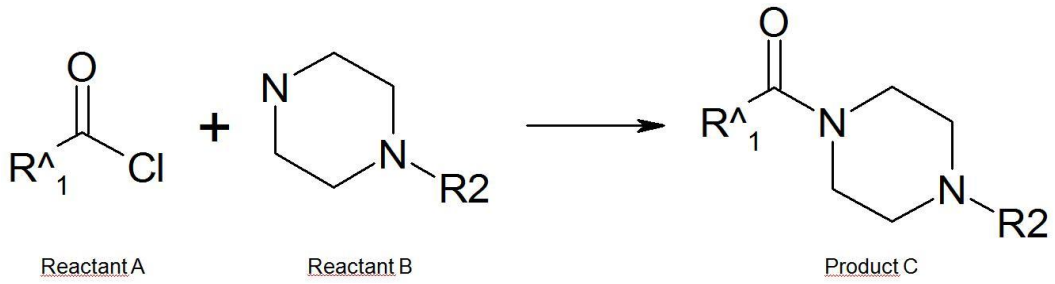
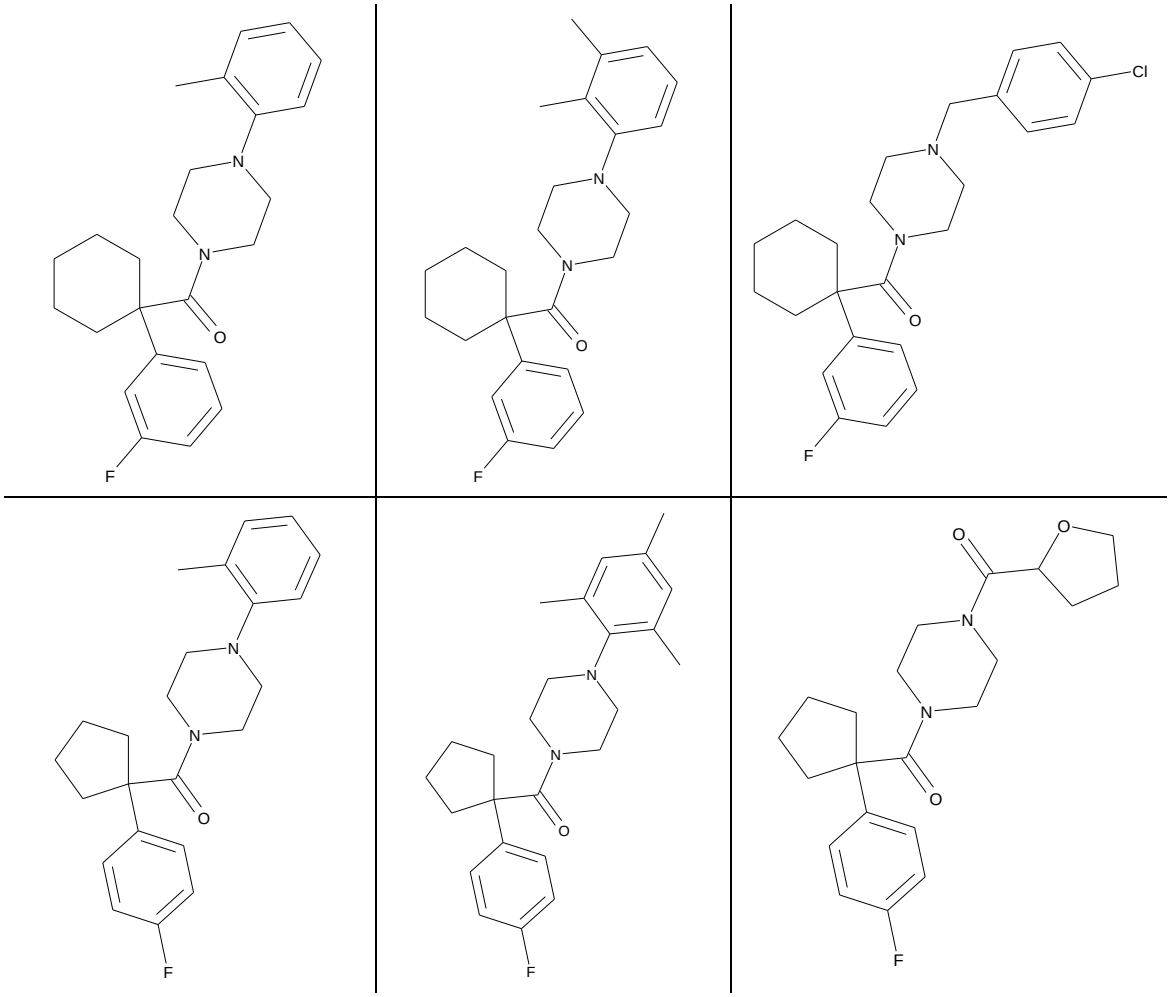


Figure 6.13: This generic example reaction is used to build a small chemical library for a virtual screening analysis.



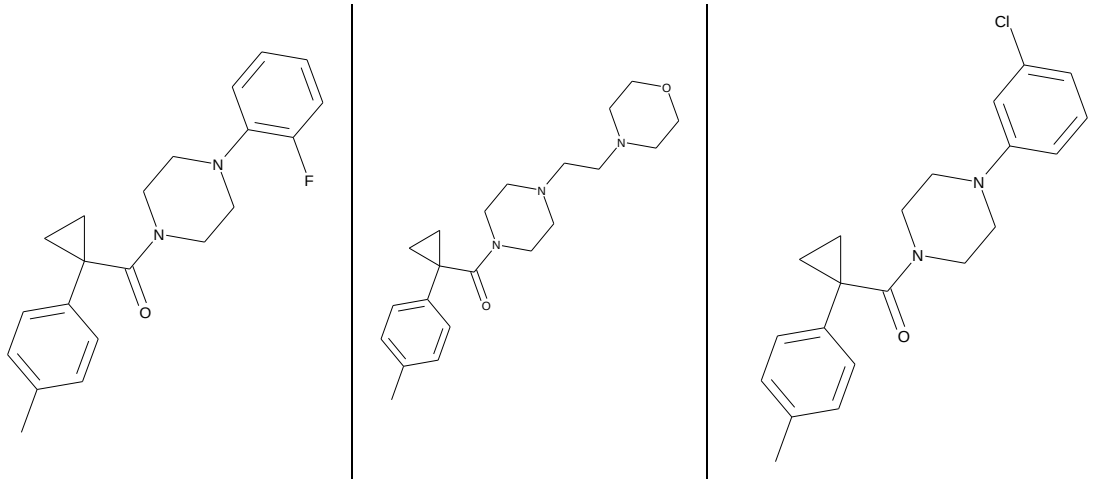


Figure 6.14: These molecules are a subset of the enumerated products from the generic reaction shown in Figure 6.13.

This workflow is a starting point for creation of chemical libraries based on generic reactions for virtual screening analysis. The example reaction shown in Figure 6.13 was used for an enumeration with a list of 12 reactants for the reactant A and 64 reactants for the reactant B. This enumeration results in 768 products, a subset of them is shown in Figure 6.14.

The current implementation of this algorithm is limited to two reactants per generic reaction and only one generic group per reactant, but addresses the majority of industrial enumeration requirements.

6.3 Machine learning workflows with *GNWI.CIP*

Taverna-GNWI.CIP integration was tested with a machine learning problem using kinetics data of an adhesive polymer model system taken from the literature[1] (143): The "time to maximum temperature" (t_{Tmax}, a measure for the hardening characteristics) is determined as a function of different compositions of the adhesive system:

[1] The experimental data are part of the test data package of *GNWI.CIP*.

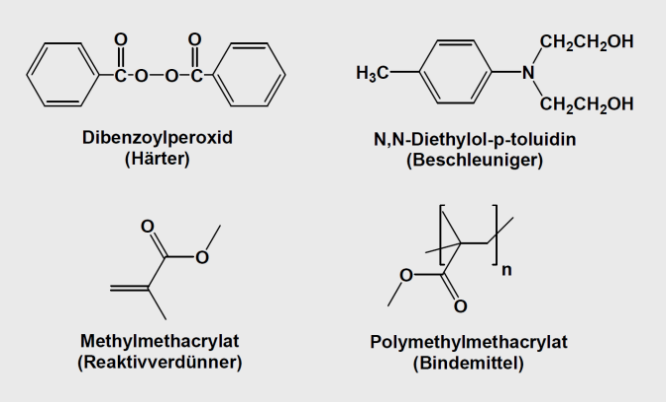


Figure 6.15: Composition of the adhesive polymer model system (from top left to bottom right): Hardener, accelerator, diluter (MMA), polymer (PMMA)

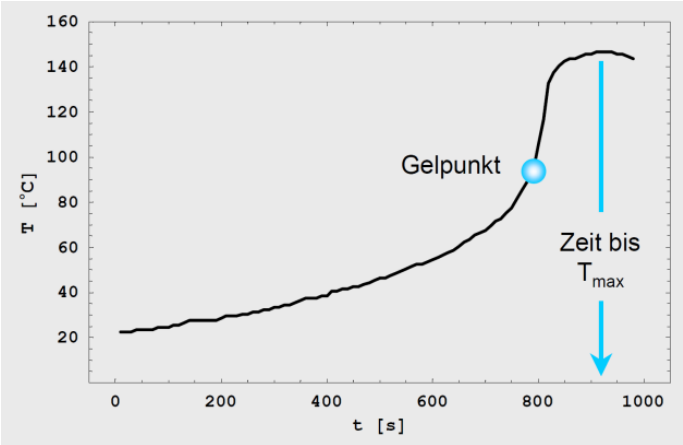


Figure 6.16: Hardening kinetics: Definition of the "time to maximum temperature" t_{Tmax}

Theoretical predictions of this kind of kinetic parameters for a polymer mixture are impossible so machine learning is employed to set up a prediction model from the experimental data. Input parameters are the compositions of the four components of the adhesive model system and the single output parameter is t_{Tmax}.

The workflow consists of a data access worker to get the experimental data in form of a string representation suited for input into the *GNWI.CIP* machine learning methods, machine learning workers that encapsulate the complete *GNWI.CIP* machine learning steps and visualisation workers that show the results.

A principal problem of all machine learning methods is the so called "overfitting" of the experimental data which may be detected by visual inspection or data segmentation into

training and test sets – which in turn is a principal problem. The following figures show a distinct "overfitting" for a perceptron-type neural network with "too many" hidden units and a dramatic "overfitting" of a support vector machine with an unfavourable wavelet kernel that lead to models with a low, up to a vanishing, predictivity (for all figures: MMA/PMMA mass ratio: 80/20, Input 2: Hardener in g, Input 3: Accelerator in g, Output 1: t_{Tmax} in s):

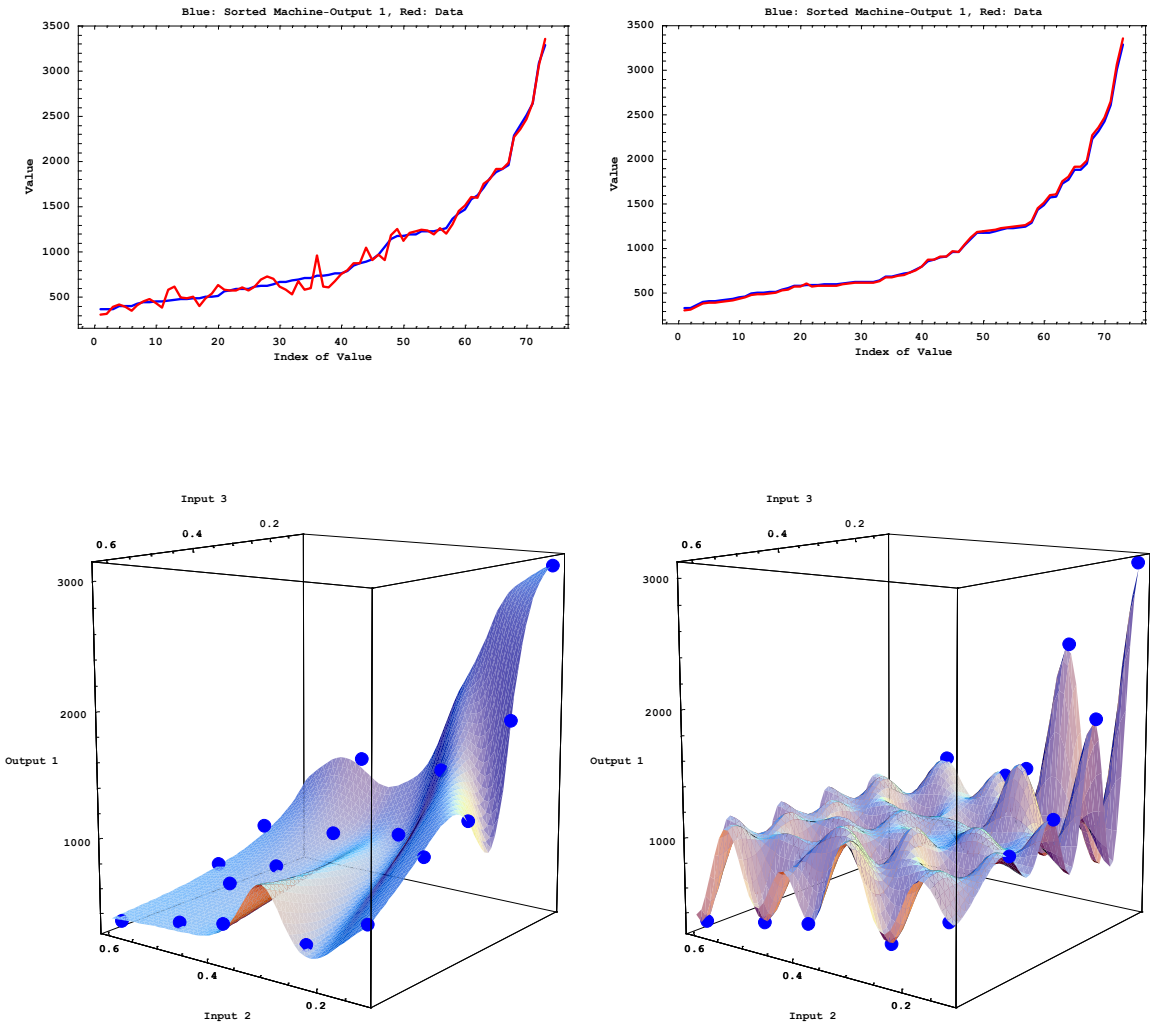


Figure 6.17: The left side shows a perceptron-type neural network with 9 hidden units and the right side a wavelet kernel support vector machine with kernel parameter of 0.1

Without visual inspection of the fitted model surface the deviation plots could be misleading and pretend a "good fit". Satisfactory prediction results without "overfitting"

and within the experimental error limit of about 15% can be obtained with an adequate choice of the hidden units and the kernel function respectively:

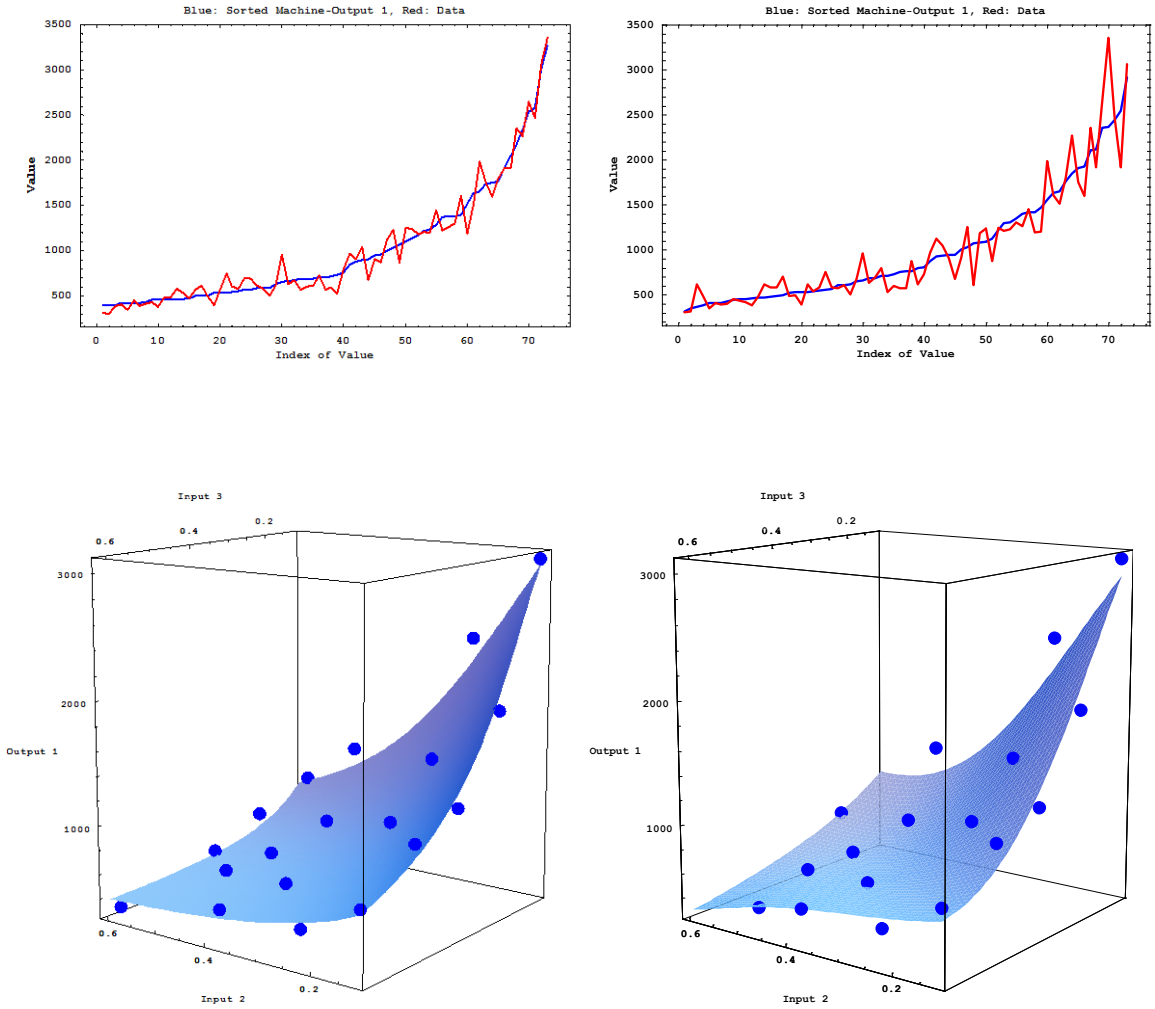


Figure 6.18: The left side shows the result of a perceptron-type neural network with 5 hidden units and on the right side the result of a wavelet kernel support vector machine with kernel parameter of 0.9

GNWI.CIP also provides methods for training and test set generation, enrichment and optimization which may help to address the "overfitting" problem. These methods can be implemented by the use of pre-processing *Taverna* workers which are inserted into machine learning workflows.

6.4 Data Analysis Workflows

During this project, different kind of workers has been created to support the whole cycle of chemical data analysis: From the creation of chemical data, via filtering and pre-processing, to the analysis and finally the comparison of different data analysis procedures. This chapter starts with the description of two different kinds of substructure searches. This enables a filtering or removal of molecules containing specific substructures. After this short excurse the calculation of physiochemical properties is shown for workflows using different data analysis algorithms. At the end of this chapter a chemical diversity analysis is demonstrated that uses publicly available as well as proprietary drug-development databases.

The proprietary data used within the data analysis is provided through the cooperation with *InterMed Discovery* (144), a natural product lead-discovery company. The work on the data of the proprietary in-house database from *InterMed Discovery* and on the data of the *Chapman & Hall Chemical Database* was performed within the premises of *InterMed Discovery*.

6.4.1 Substructure Search on a Database

A Cheminformatics database cartridge allows the indexing of molecules, which leads to a very fast substructure search.

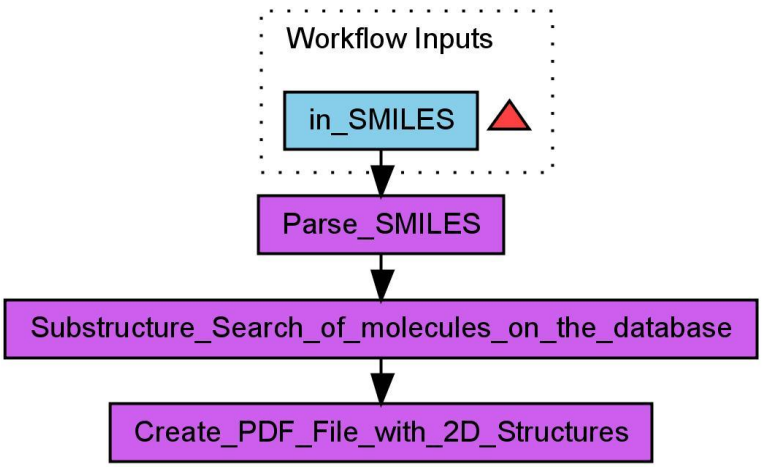


Figure 6.19: This workflow performs a substructure search on the database. The SMILES as workflow input represents the substructure for this search. (145)

The workflow shown in Figure 6.19 performs a substructure search on the molecules of a database. The *Parse_SMILES* worker takes the SIMES from the workflow input and processes it to an internal molecular representation.

The *Substructure_Search_of_molecules_on_the_database* worker gets the molecule and performs a substructure search on the database. Depending on the SQL query, this worker can perform a substructure search or an exact search. As result all matched molecules are loaded from the database for further processing. In this case the matched molecules are summarised with their 2D structures into one PDF document.

6.4.2 Topological Substructure Search Workflow

Besides the substructure search on a database shown in chapter 6.4.1 the *CDK-Taverna* plug-in provides a worker to perform a topological substructure search. The performance of this kind of substructure search is not comparable with the database substructure search because this one cannot refer to an index.

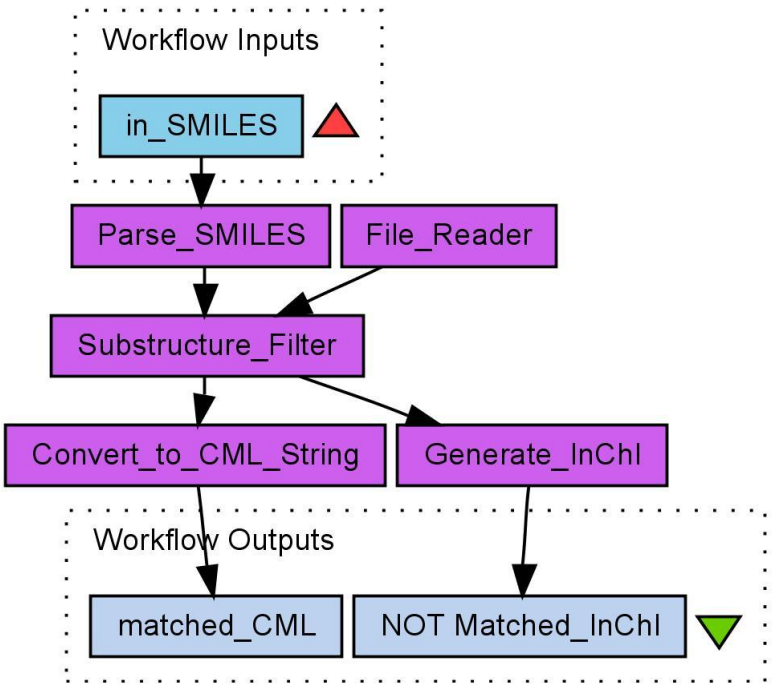


Figure 6.20: This workflow performs a topological substructure search. (146)

The topological substructure search workflow shown in Figure 6.20 loads the molecules to be searched from a MDL SD file with the help of the *File_Reader* worker. The substructure for the search has to be entered in the SMILES format as workflow input. After the parsing of the SMILES from the *Parse_SMILES* worker, the *Substructure_Filter* worker performs the topological substructure search. As result of this workflow, the structures, which contain the substructure, are stored in the CML format. The *Generate_InChI* worker generates an InChI™ for each of the other structures.

6.4.3 Calculation of Molecular Descriptors

The generation of chemical data used in a data analysis is a crucial step within a data analysis cycle. For chemical problems usually physiochemical properties are calculated to support QSAR or QSPR analysis. Within this work we used the molecular descriptors provided by the CDK to generate the data for the analysis.

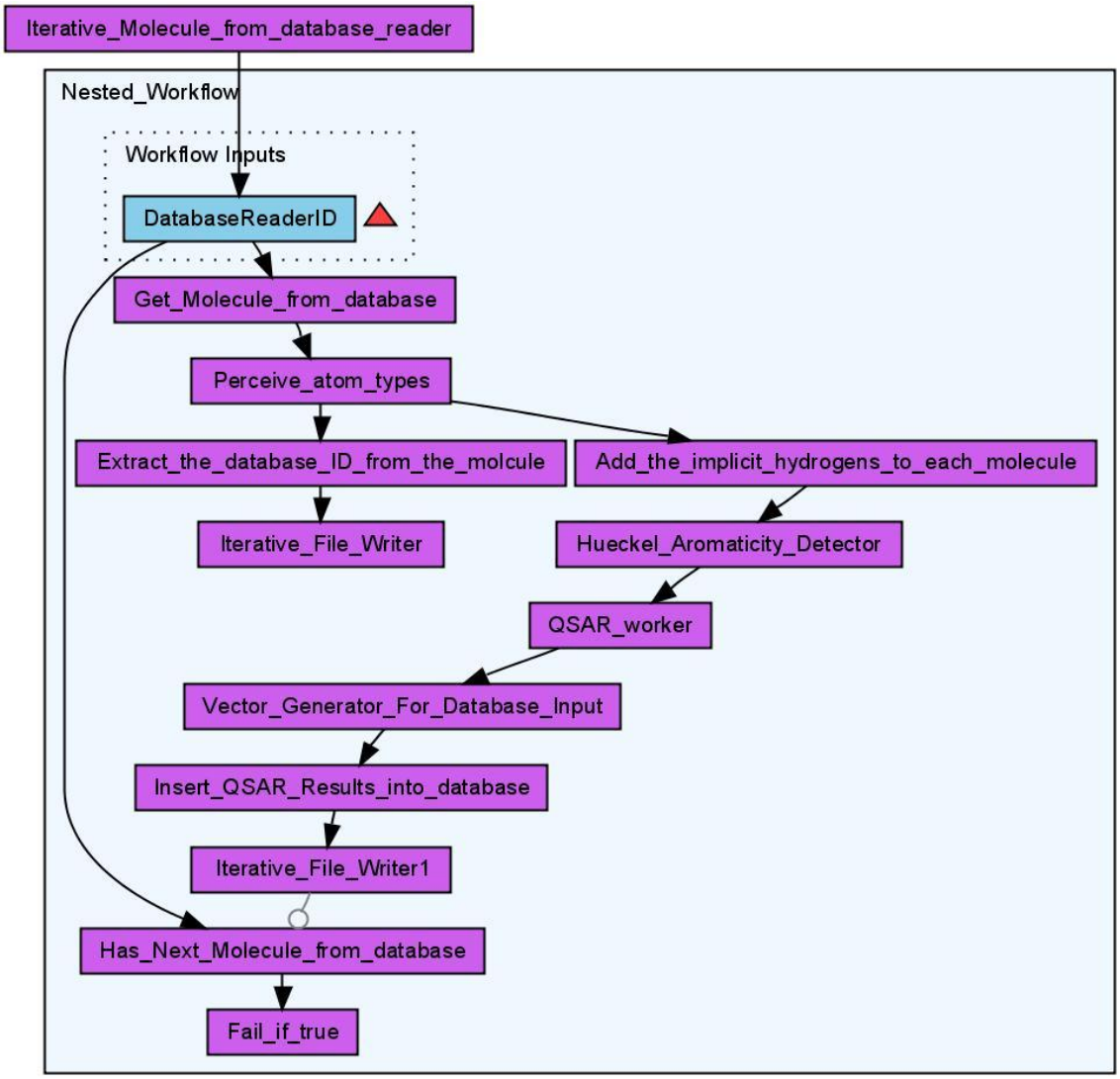


Figure 6.21: This workflow loads iteratively molecules from a database. Then it perceives the atom types, adds the hydrogen's and detects the aromaticity of each molecule before it performs the calculation of QSAR descriptors. The result of the calculation is stored in a database table. (147)

The workflow shown in Figure 6.21 is used to calculate the molecular descriptors and store the results on a database table. The workflow uses an iterative loading of molecules from a database approach to be able to calculate descriptors for many thousands of molecules. After the *Get_Molecule_from_database* worker fetches the molecules from the database, the perceiving of atom types is performed by the *Perceive_atom_types* worker. Each molecule than gets it implicit hydrogen's before the aromaticity is detected by the *Heuckel_Aromaticity_Detector*. These are the requirements before the *QSAR_worker* worker can calculate the different descriptors for each molecule. This worker provides a user interface for choosing the descriptors to

calculate (see Figure 6.22). From the calculated properties the *Vector_Generator_For_Database_Input* worker generates a vector which is used to store the values in a database table for further processing.

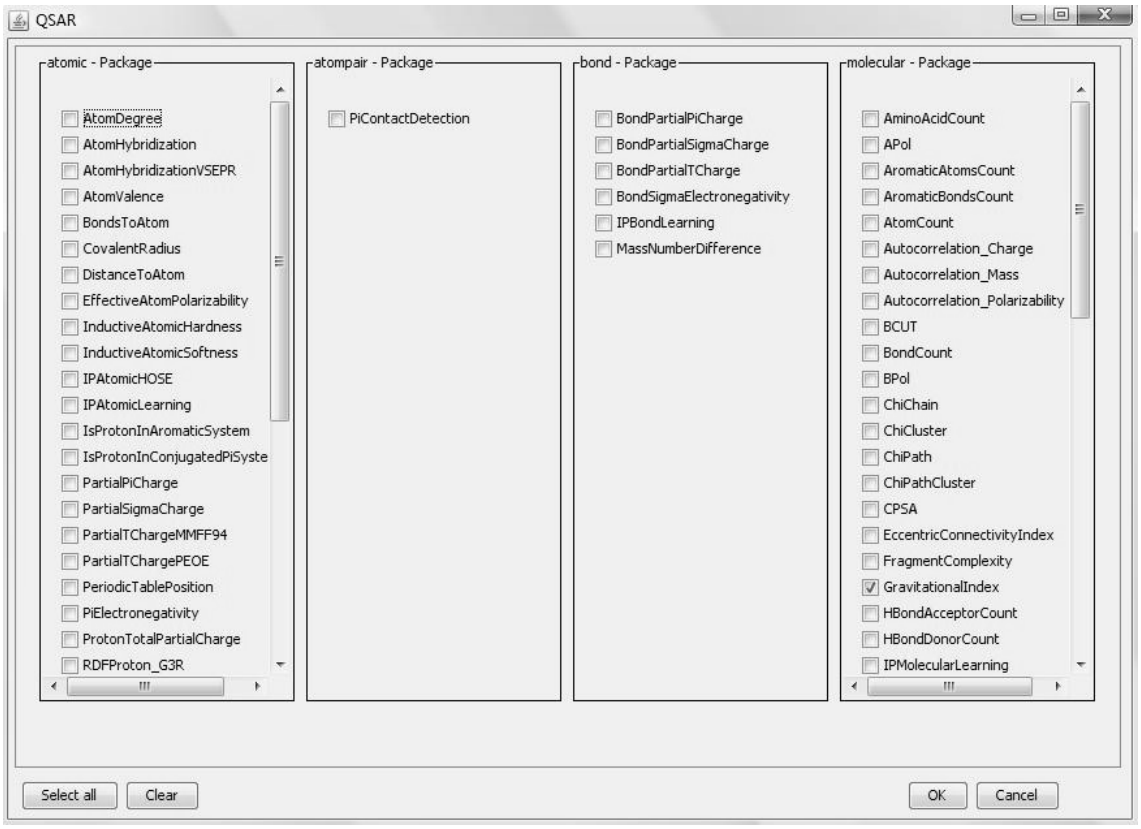


Figure 6.22: This user interface of the *QSAR_worker* worker shows the available descriptors and enables the user to select the descriptors to calculate.

The database table for storing the calculated physiochemical properties contains a column for each descriptor value. Only the molecular descriptors are calculated during the work of this project and from a performance respective the time needed to calculate each descriptor varies significantly. An overview of the time needed to calculate some molecular descriptors is shown in Figure 6.23.

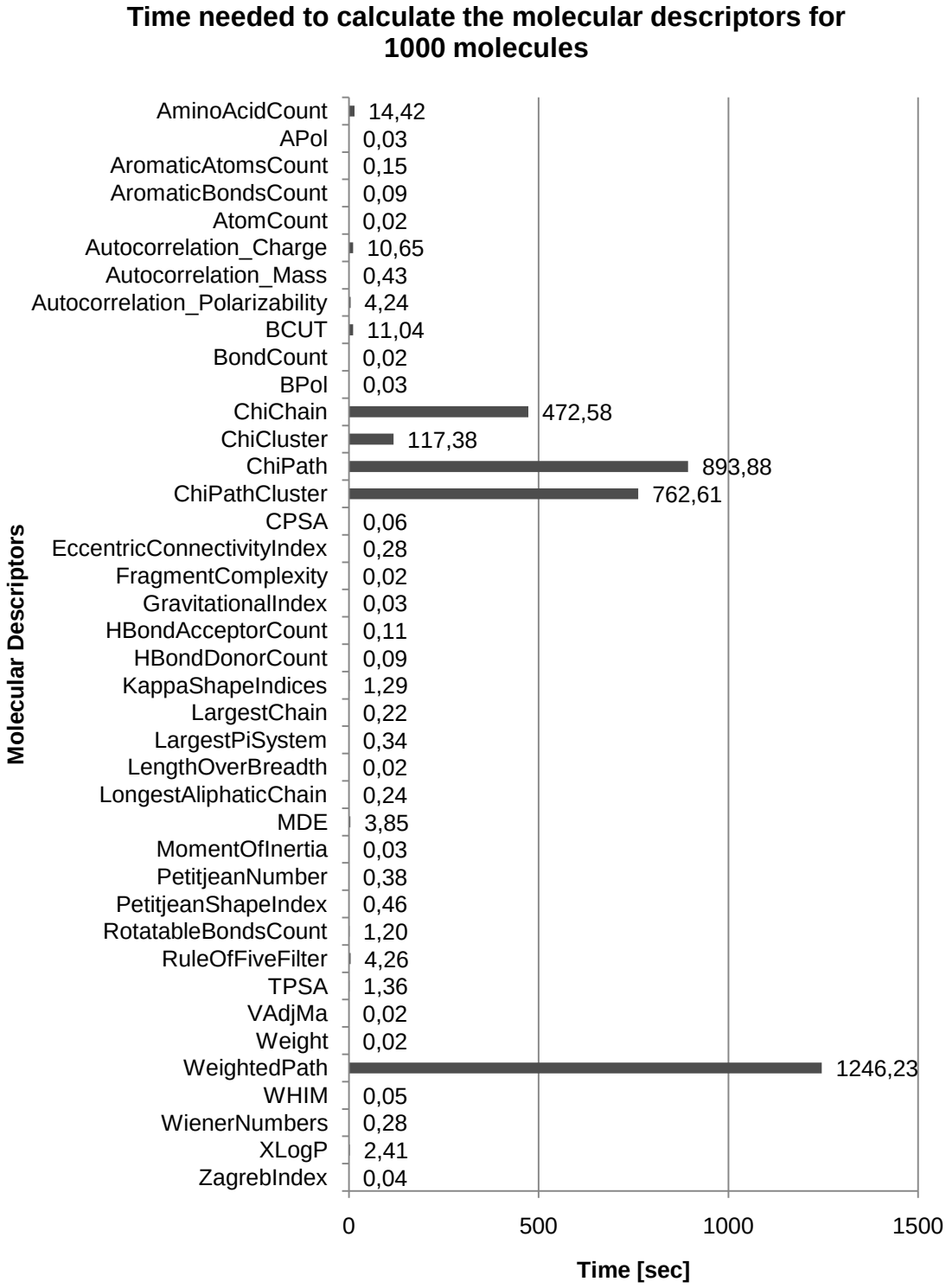


Figure 6.23: The diagram shows the time needed for each descriptor to calculate the descriptor values for 1000 molecules.

The analysis of the time needed to calculate the different descriptors shows that the CDK performs quite well for the majority of molecular descriptors.

6.4.4 Molecular Weight Distribution

A molecular weight distribution contains valuable information about large chemical datasets. It allows an examination of the diversity of the chemical dataset regarding the size of the molecules.

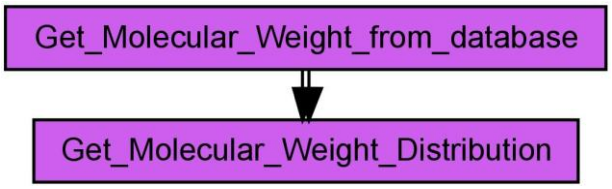


Figure 6.24: This workflow creates a chart, which shows the molecular weight distribution for a given set of molecules from a database query. (148)

The small workflow shown in Figure 6.24 creates a molecular weight distribution. Therefore the *Get_Molecular_Weight_from_database* worker uses the possibility of the cheminformatics database cartridge to calculate the molecular weight and outputs this value for each molecule. The *Get_Molecular_Weight_Distribution* worker creates from the different molecular weights a distribution and outputs a diagram as result. During the work for this project the molecular distributions are created for the *ChEBI* (see Figure 6.25), *ChEMBL* (see Figure 6.26), the *Chapman & Hall Chemical Databases* (see Figure 6.27) and the proprietary database from *InterMed Discovery* (see Figure 6.28).

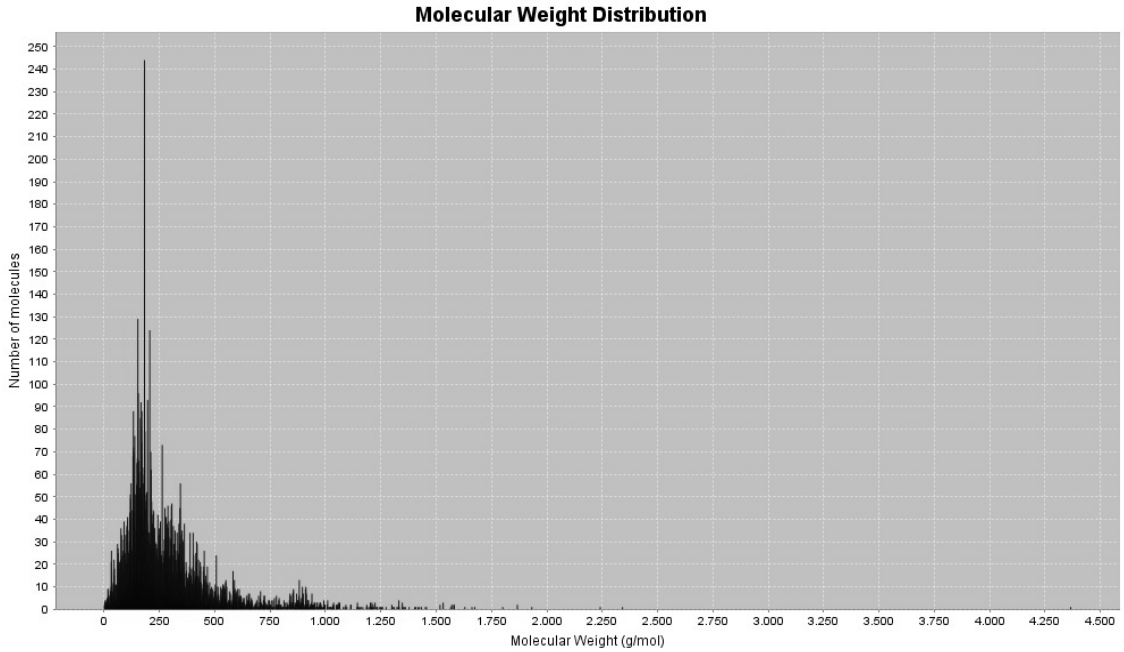


Figure 6.25: The diagram shows the molecular weight distribution of the *ChEBI* database.

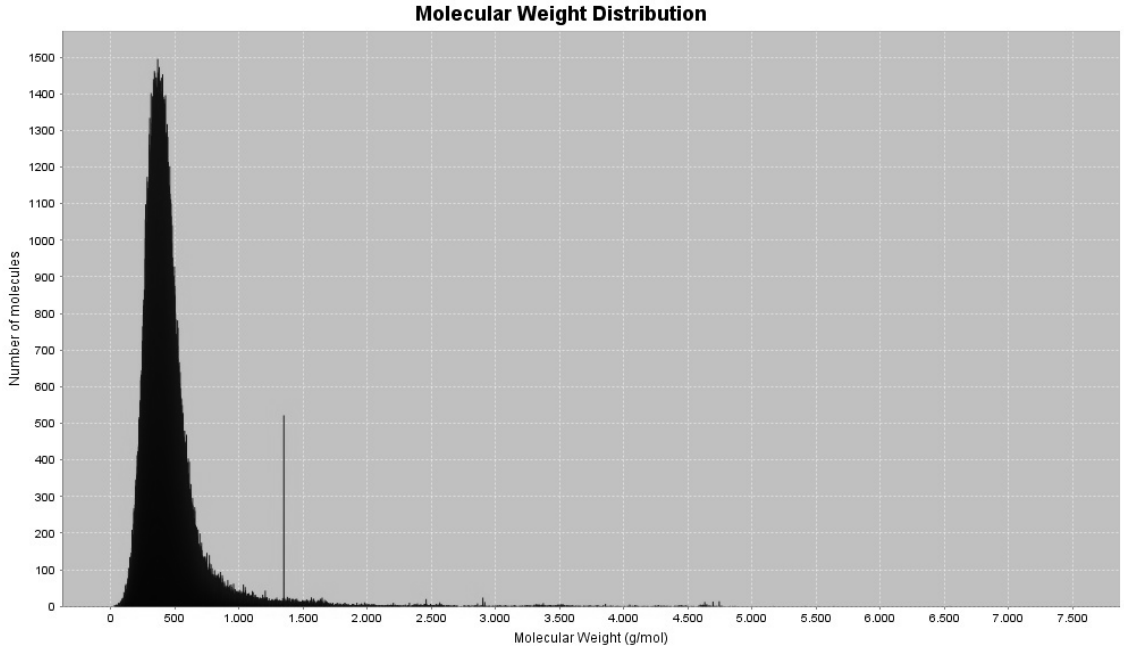


Figure 6.26: The diagram shows the molecular weight distribution of the *ChEMBL* database.

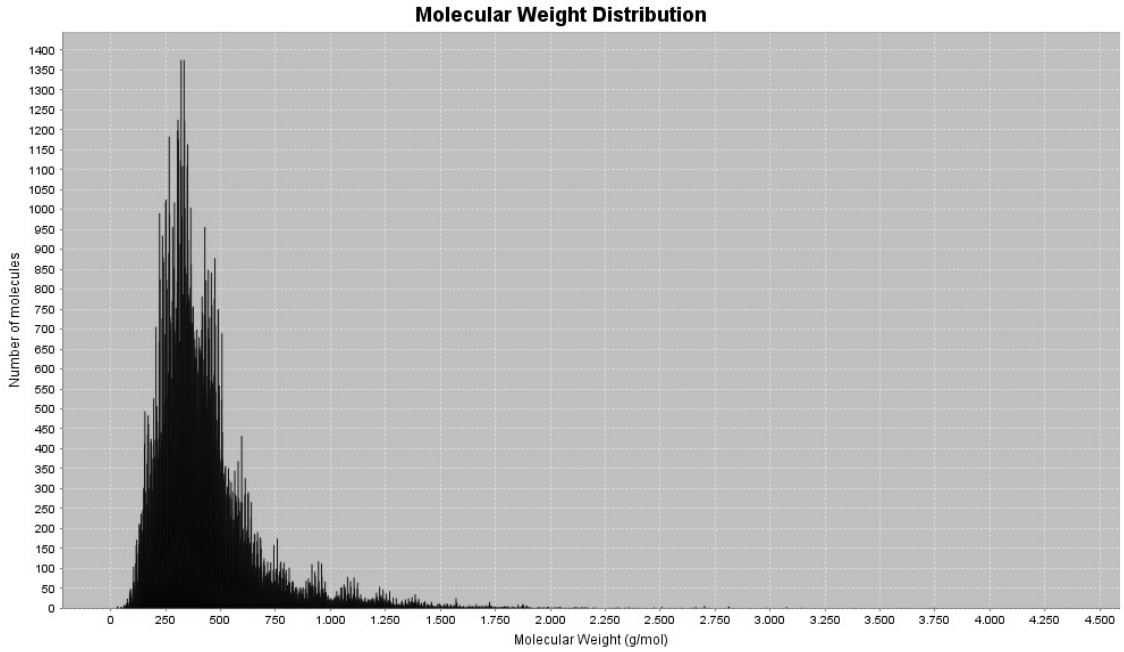


Figure 6.27: The diagram shows the molecular weight distribution of the *Chapman & Hall Chemical Database*.

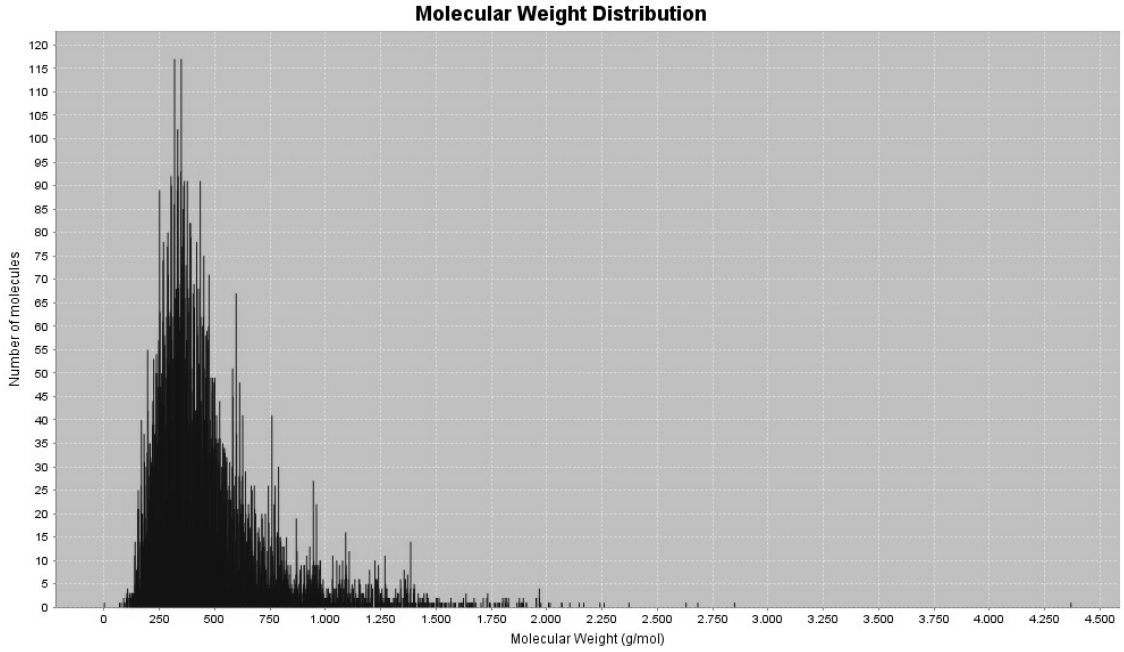


Figure 6.28: The diagram shows the molecular weight distribution of the proprietary database from *InterMed Discovery*.

The molecular weight distribution from the *ChEBI* database shows a maximum at approximately 225g/mol whereas the *ChEMBL* database has its maximum at

approximately 400 g/mol. These two databases have a slightly different molecular weight distribution indicated through the moving to a higher maximum from the *ChEMBL* database. The molecular weight distribution of the *ChEMBL* database shows an artefact at 1347 g/mol. Over five hundred molecules contain exactly the same molecular mass of 1347.63002 g/mol and have the structure showing in Figure 6.29. This artefact indicates a curation problem in the time of building this database.

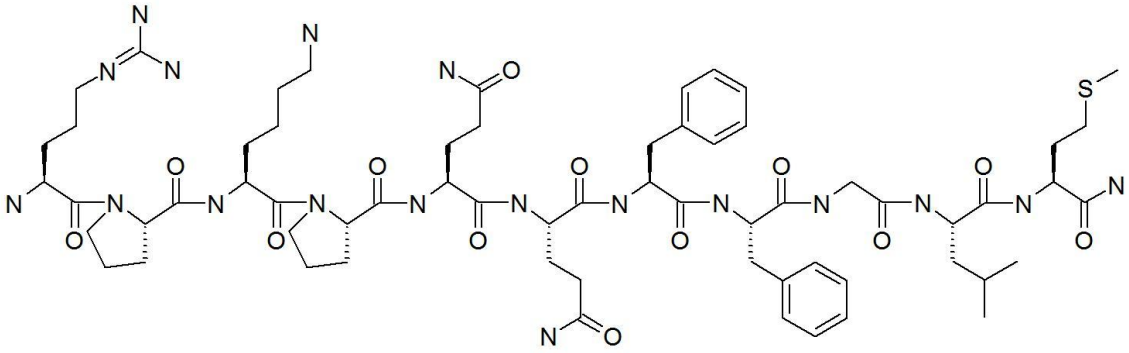


Figure 6.29: The *ChEMBL* database contains this molecular structure over five hundred times.

The *Chapman & Hall Chemical Database* and the proprietary database from *InterMed Discovery* have similar molecular weight distributions with a maximum at approximately 350 g/mol. Only with the information of the molecular weight distributions a differentiation of these databases is not possible.

6.4.5 Classification Workflows

During the work on this project classification workflows are mainly used to perform an unsupervised clustering. For the majority of classifications an implementation of the ART 2-A algorithm (described in chapter 4.4.5 and chapter 8.2) developed by *Stephen Grossberg* and *Gail Carpenter* performs this task. This algorithm was chosen because of its capability to automatically classify open-categorical problems and compared with the "k^{th}-nearest-neighbour"-clustering the ART 2-A algorithm is computationally less demanding.

Figure 6.30 shows an example workflow which performs an ART 2-A classification. In this, the *Get_QSAR_vector_from_database* worker loads for a specific SQL query a list of data vectors from the database. This worker provides options to inspect the result

vector which includes checks for values such as *Not a Number* (NaN) or *Infinity*. The options allow choosing the threshold for removing whole components of the vectors or single vectors and the removing of components whose min value equals its max value.

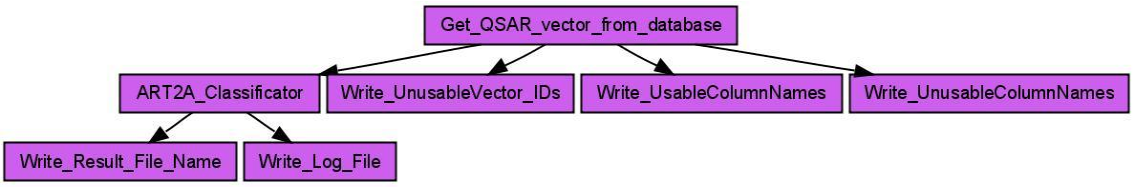


Figure 6.30: This workflow loads a data vector from a database and performs an ART2A classification. (149)

After the loading and a first cleaning of the data vector the classification is performed using the *ART2A_Classificator* worker. For the configuration of this worker different options are editable.

- For a linear scaling of the input vector to values between 0 and 1.
- To switch between deterministic random and random random for the selection of the vectors to process.
- To switch the convergence criteria of the classification.
- To set the required similarity for the convergence criteria.
- To set the maximum classification time.
- To set the number of classifications and a range for the vigilance parameter.

The implemented ART 2-A algorithm contains two possible convergence criteria. It converges if the classification does not change after one epoch or if the scalar product of the classes between two epochs is less than the required similarity. As output the worker stored the result within a compressed XML document. For the visualisation of the results this document is processed again. For this visualisation different workers are available depending on the aim of the task.

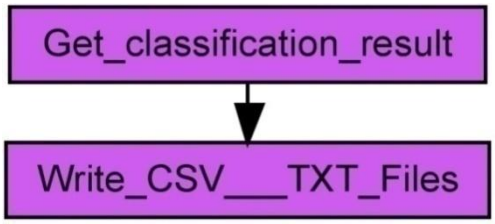


Figure 6.31: This workflow loads selected ART 2-A classification results and creates a table which contains different properties for a comparison between the different classifications.

The workflow shown in Figure 6.31 loads selected ART 2-A classification results and creates a table which shows the results of the different classifications such as the number of detected classes, the number of vectors in a class or whether the classification converged or not.

With the last two workflows a screening of the vigilance parameter of an ART 2-A classification is possible. The diagram (see Figure 6.32) shows the number of cluster while screening of the vigilance parameter depending on the distribution of the data. To get a reasonable outcome from an ART 2-A classification, the result of such an analysis should equal the green curve of this diagram.

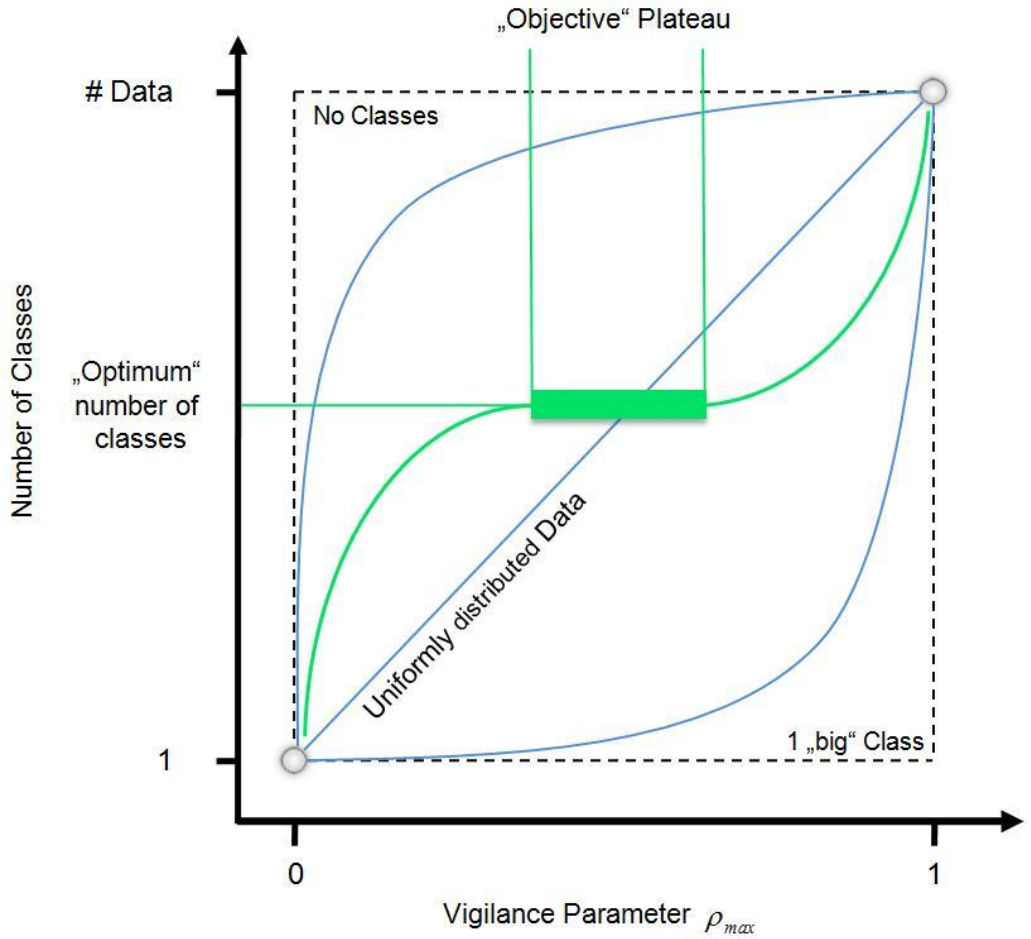


Figure 6.32: This diagram schematically shows the possible allocation of the classes/cluster while varying the vigilance parameter of an ART 2-A classification. The blue curves indicates the detected number of cluster if the distribution of the data, used for this clustering, showing no or one big class. The green curve indicates a distribution of the data which contains an objective plateau showing an “optimum” number of classes.

For performing this and the following analysis procedures of this chapter, descriptors from the CDK are calculated for each molecule. The analyses are performed on the combined molecules from the *ChEBI*, the *Chapman & Hall Chemical Database* and a proprietary database from *InterMed Discovery*. For all molecules of these three databases only 2D coordinates are available. This reduces the number of descriptor because descriptors using 3D coordinates for its calculations couldn’t be considered. A list of descriptors used for this analysis is contained in the appendix at chapter 8.5. The correlation of the number of clusters and the vigilance parameter from this analysis is shown in the diagram (see Figure 6.33). This diagram shows that for a wide range of the vigilance parameter values, the number of detected classes remains constant. Up to a

vigilance parameter of 0.5 the number of detected classes is between 5 and 7. This result is an indication of a stable classification with an “objective” plateau and an “optimum” number of detected classes using a vigilance parameter less than 0.5.

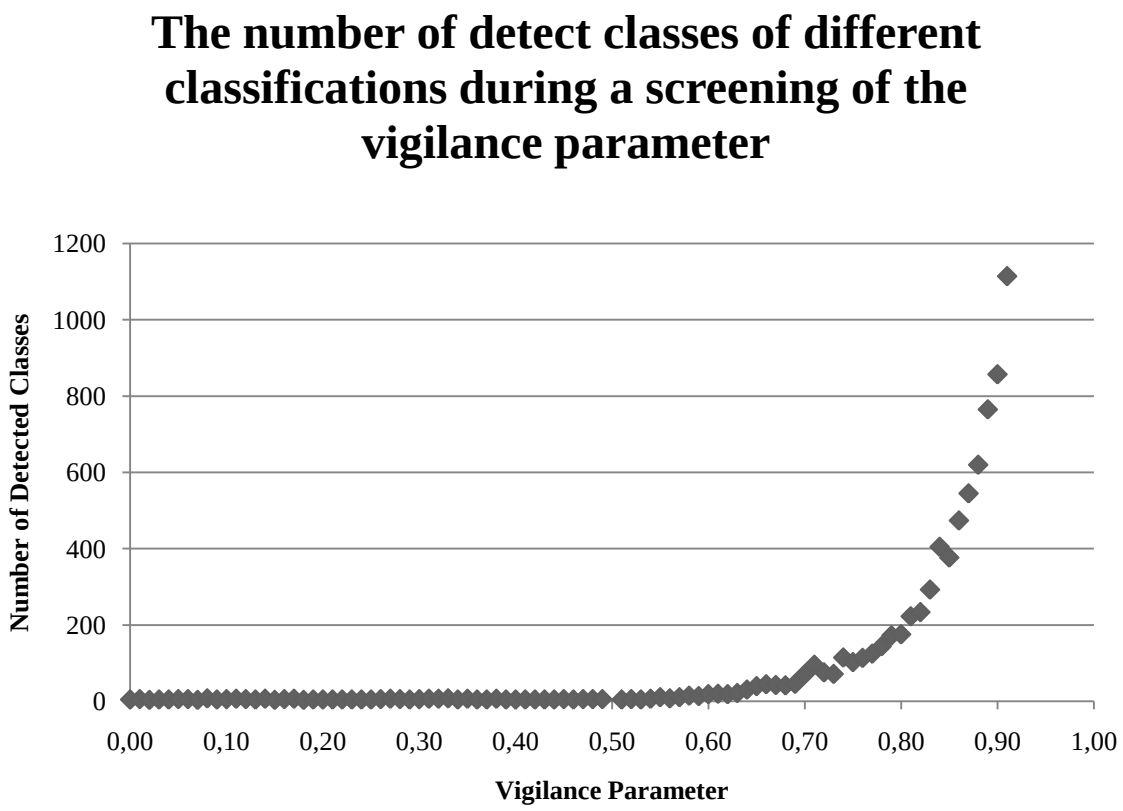


Figure 6.33: The diagram shows the dependency of the vigilance parameter with the number of detected classes of an ART 2-A classification. This allows the gradual configuration of classification properties resulting from a low number of large clusters to a high number of small clusters.

6.4.6 Chemical Diversity Analysis

To analyse the chemical diversity of different database a hierarchical clustering approach was performed using the workflows described in the last chapter. This analysis was performed on the *ChEBI*, the *Chapman & Hall Chemical Database* and a proprietary database from *InterMed Discovery*. Both, the *Chapman & Hall* and the database from *InterMed Discovery* contain mainly natural products used within the context of drug discovery. The *ChEBI* database contains molecules of biological interest which could be an important source for the discovery of new drugs. The molecular weight distributions of these three databases shown in chapter 6.4.4 do not allow an analysis of the chemical diversity. The molecular weight distributions of these databases

differ slightly but not very significantly. The hierarchical clustering approach means that after a first clustering the detected clusters are classified again up to a point where small clusters could be inspected visually.

For the chemical diversity analysis the ART 2-A classification workflow shown in Figure 6.30 was used. After the screening of the vigilance parameter shown in Figure 6.33 a value of 0.05 for this parameter was chosen for this analysis. As convergence criteria, the scalar product of the classes between two epochs should be less than the required similarity. The required similarity is set to a value of 0.99. After the classification the workflow shown in Figure 6.34 allows a visual overview of the composition of an ART 2-A classification result.

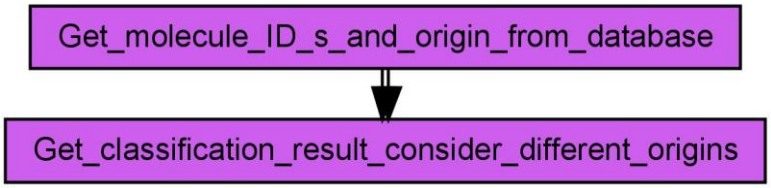


Figure 6.34: This workflow loads the result of an ART 2-A classification and visualizes the outcome. The created diagram shows the allocation of the data origin within each cluster. Besides the diagram the PDF document contains a table showing the details of the analysis. (150)

The workflow above creates a PDF document which contains a diagram (see Figure 6.35) and a spreadsheet (see Table 6).

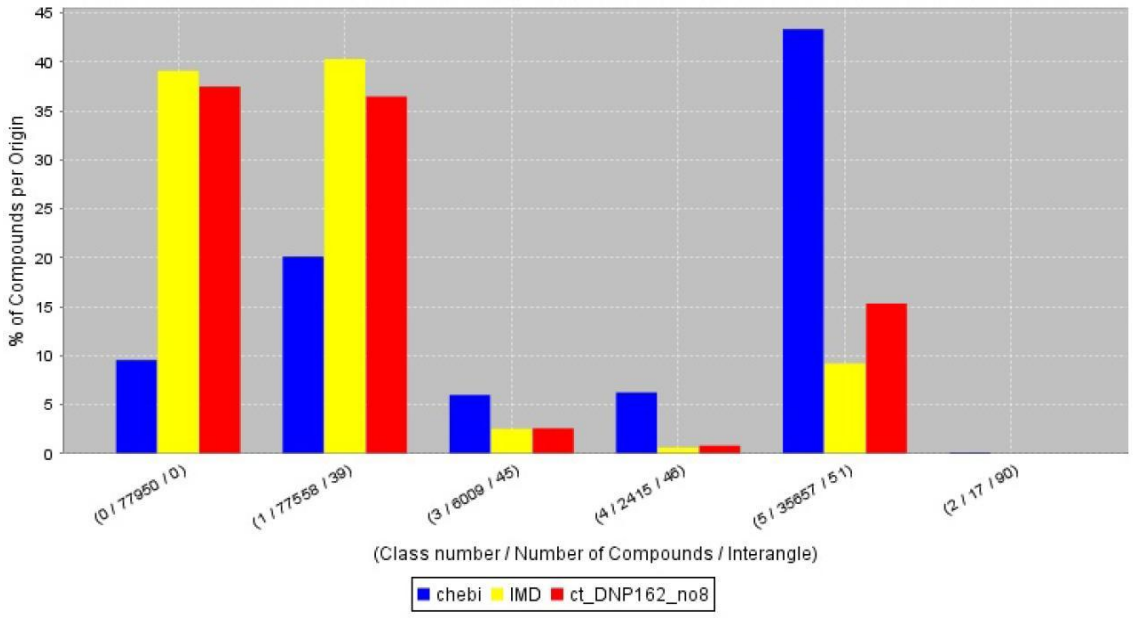


Figure 6.35: This diagram shows the composition of the different detected classes of the ART 2-A classification. The ct_DNP162_no8 represents the molecules from the *Chapman & Hall Chemical* Database. The IMD values represent the molecules from the proprietary database of *InterMed Discovery*.

The ART 2-A classification algorithm provides a possibility to define similarity between two clusters. For this it calculates the angle between two centroid vectors. This results in an angle between 0° and 90° where 90° represents the largest distance between two clusters. The classes within the diagram shown in Figure 6.35 are ordered. The first position of the diagram belongs to the largest cluster and the others are ordered regarding their similarity to the largest cluster.

Table 6: This table shows the detailed composition of the detected classes from the ART 2-A classification.

Vigilance Parameter: 0.05
Number of detected classes: 6

Cluster Number	Angle to largest class	Number of Compounds in Cluster	Number of Compounds from IMD	% of IMD	Number of Compounds from ChEBI	% of ChEBI	Number of compounds from ct_DNP162_no8	% of ct_DNP162_no8
0	0	77950	6465	39.104	1183	9.565	70302	37.494
1	39	77558	6666	40.319	2489	20.125	68403	36.481
3	45	6009	417	2.522	741	5.991	4851	2.587
4	46	2415	104	0.629	773	6.250	1538	0.820
5	51	35657	1523	9.212	5363	43.362	28771	15.344
2	90	17	0	0.0	17	0.127	0	0.0

The ART 2-A classification of the three databases detected six clusters. The population of different classes indicates a similar coverage of the chemical space from the *Chapman & Hall Chemical Database* and the proprietary database from *InterMed Discovery* whereas the *ChEBI* database covers a distinctly different chemical space. Noteworthy in this classification is the cluster no. 2, which contains only 17 molecules and all molecules originating from the *ChEBI* database. This cluster is orthogonal to the largest cluster and contains only single atoms, small molecules, ions or radicals such as Br, F, Li, I, SiH_3, Al^{3+}, S^- or O^-.

After the first classification (which created five clusters with many thousand molecules and only one small cluster), a further classification of the large cluster was necessary to get a deeper insight into the constitution of the chemical space covered by these three databases. A clustering of the largest cluster using the same classification properties resulted in four different clusters, which are shown in the diagram (see Figure 6.36). Each of these four clusters still contains more than six thousand molecules, which would be far too much for-visual inspection by a scientist.

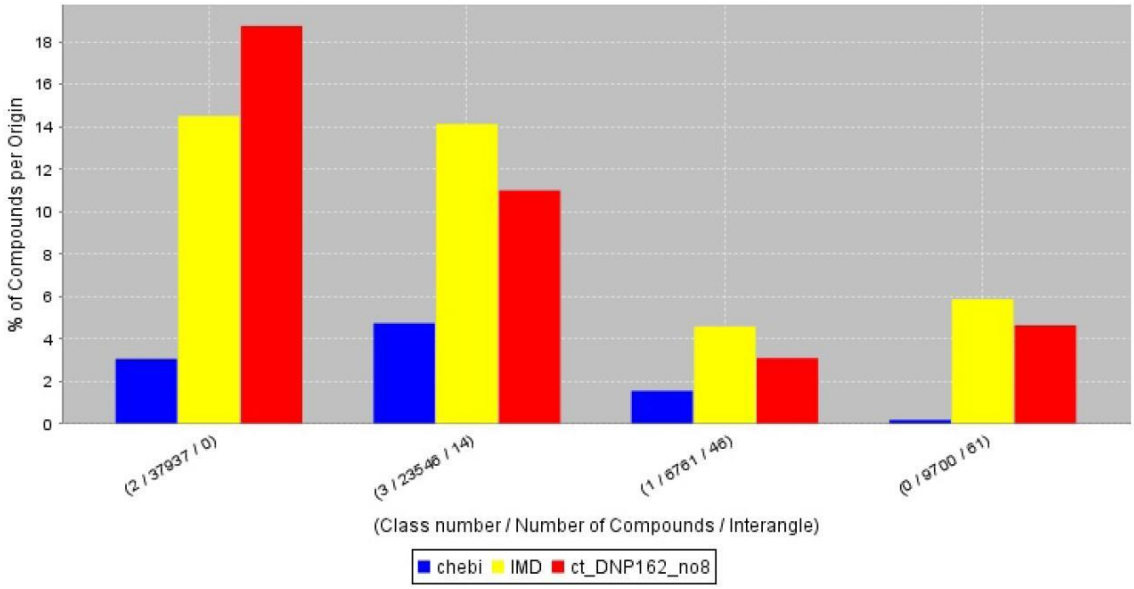


Figure 6.36: The ART 2-A classification of the largest cluster from the classification result shown in Figure 6.35 leads to four different clusters.

The clustering result shown above confirmed the first result obtained when examining the results from the coverage of the chemical space from the *Chapman & Hall*

Chemical Database and the *InterMed Discovery* database which was gained from the first classification.

Table 7: This table shows the detailed composition of the detected classes from the classification of the largest cluster of the results from Figure 6.35.

Vigilance Parameter: 0.05
Number of detected classes: 4

Cluster Number	Angle to largest class	Number of Compounds in Cluster	Number of Compounds from IMD	% of IMD	Number of Compounds from ChEBI	% of ChEBI	Number of compounds from ct_DNP162_	% of ct_DNP162_ no8
2	0	37937	2399	14.510	379	3.064	35159	18.751
3	14	23546	2336	14.129	587	4.746	20623	10.999
1	46	6761	759	4.591	192	1.552	5810	3.099
0	61	9700	971	5.873	23	0.186	8706	4.643

Another ART 2-A classification of the largest cluster from the classification result shown in Figure 6.36 using the same algorithm configuration leads to twelve detected classes shown in Figure 6.37.

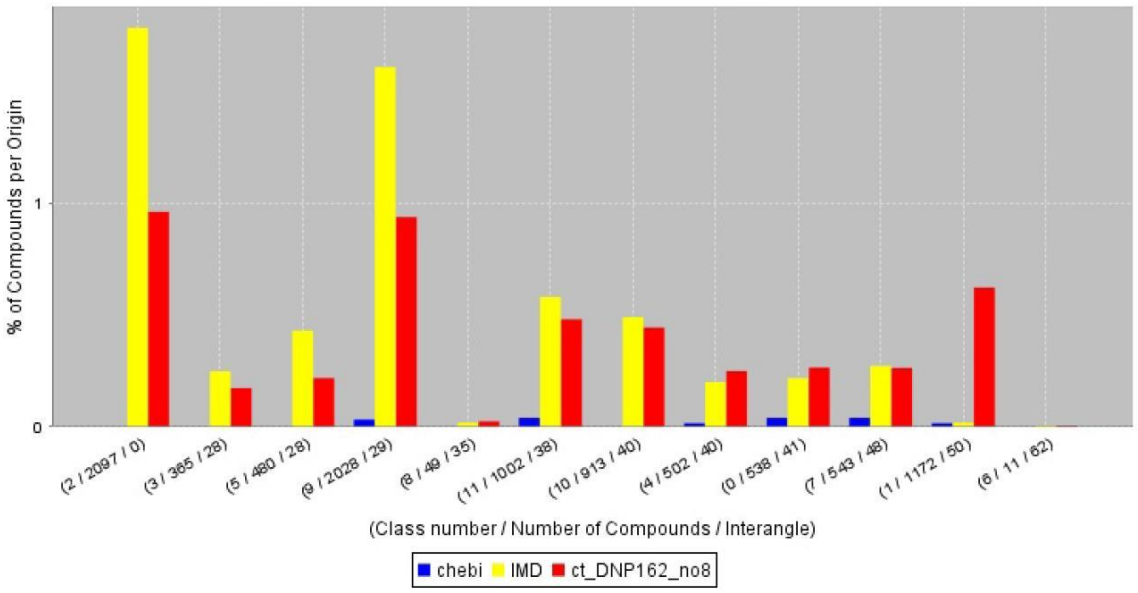


Figure 6.37: This diagram shows twelve clusters, which are the result of another ART 2-A classification of the largest cluster from the classification result shown in Figure 6.36.

Table 8: The table shows the detailed composition of the detected classes from the classification of the largest cluster of the results from Figure 6.36

Vigilance Parameter: 0.05
Number of detected classes: 12

Cluster Number	Angle to largest class	Number of Compounds in Cluster	Number of Compounds from IMD	% of IMD	Number of Compounds from ChEBI	% of ChEBI	Number of compounds from ct_DNP162_no8	% of ct_DNP162_no8
2	0	2097	295	1.784	0	0.0	1802	0.961
3	28	365	41	0.248	0	0.0	324	0.173
5	28	480	71	0.429	0	0.0	409	0.218
9	29	2028	266	1.609	4	0.032	1758	0.938
8	35	49	3	0.018	0	0.0	46	0.025
11	38	1002	96	0.581	5	0.040	901	0.481
10	40	913	81	0.490	0	0.0	832	0.444
4	40	502	33	0.200	2	0.016	467	0.249
0	41	538	36	0.218	5	0.040	497	0.265
7	48	543	45	0.272	5	0.040	493	0.263
1	50	1172	3	0.018	2	0.016	1167	0.622
6	62	11	1	0.006	0	0.0	10	0.005

The third hierarchical stage of the clustering detected twelve clusters each contains fewer than 2100 molecules. The majority of clusters contain fewer than 600 molecules, which would be a reasonable number for a visual inspection of the constitution of a cluster. This classification result represents a chemical space which is not covered by the *ChEBI* database. It can be seen that the other two databases only slightly differ in the coverage of the chemical space. Noticeable are the clusters no. 2, 5 and 9 where the database from *InterMed Discovery* shows a larger population than the *Chapman & Hall Chemical Database*. The cluster no. 1 contains almost exclusively molecules from the *Chapman & Hall Chemical Database* (molecules shown in Figure 6.39).

Each of the next four figures contains some molecules of a distinct cluster. The number of molecules showed from each origin does not correlate with the constitution of the clusters. The first figure (Figure 6.38) shows molecules from the cluster no. 0 where the *Chapman & Hall Chemical Database* slightly better covers the chemical space. The third figure (Figure 6.40) shows molecules from cluster no. 3 where the database from *InterMed Discovery* has a better coverage. The last of the four figures (Figure 6.41) shows all molecules from the smallest cluster of this classification.

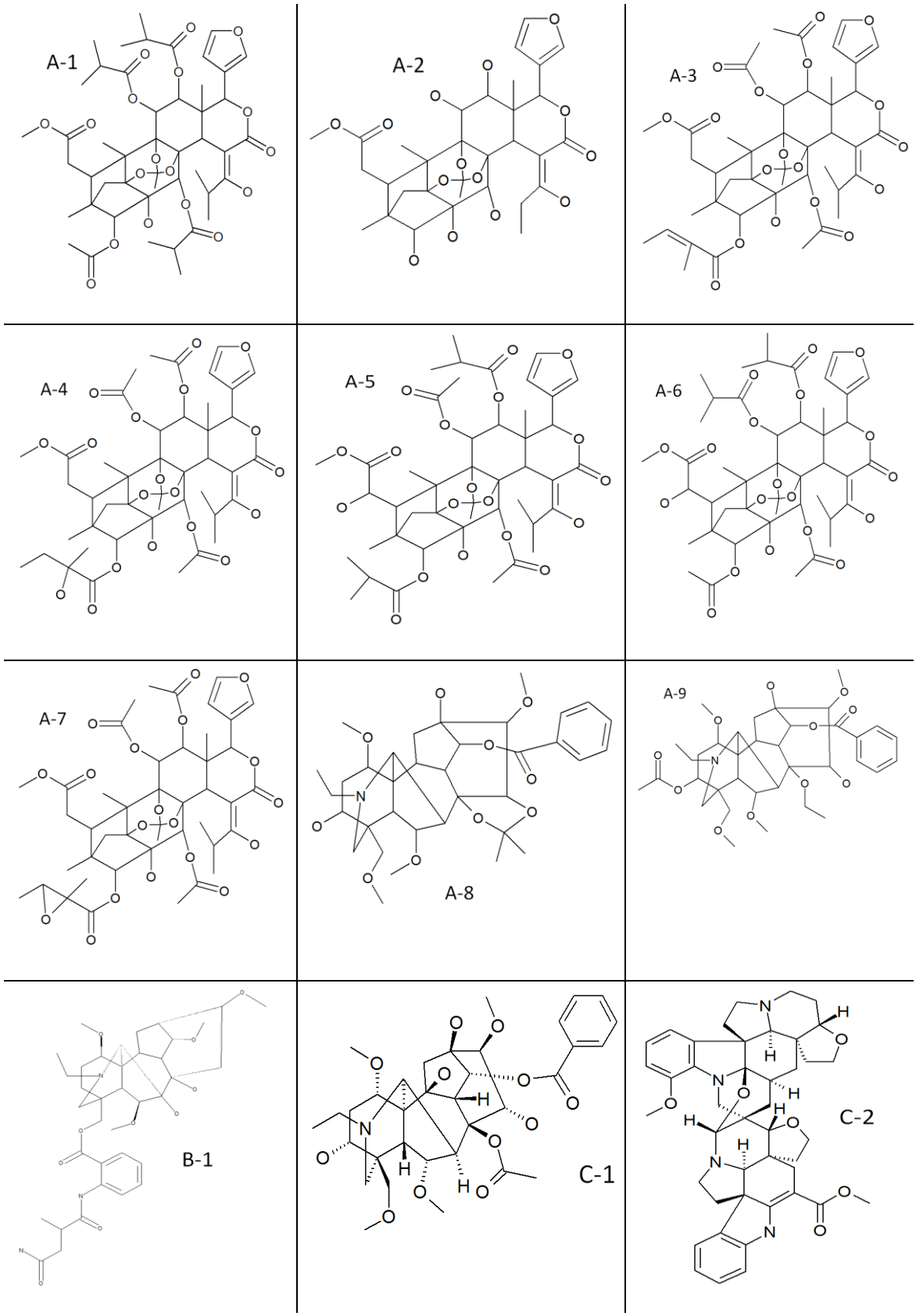


Figure 6.38: These molecules are examples of the cluster no. 0 of the classification shown in Figure 6.37. The molecules A-1 to A-9 originate from the *Chapman & Hall Chemical Database*, the molecule B-1 from the *InterMed Discovery* database and the molecules C-1 and C-2 from the *ChEBI* database.

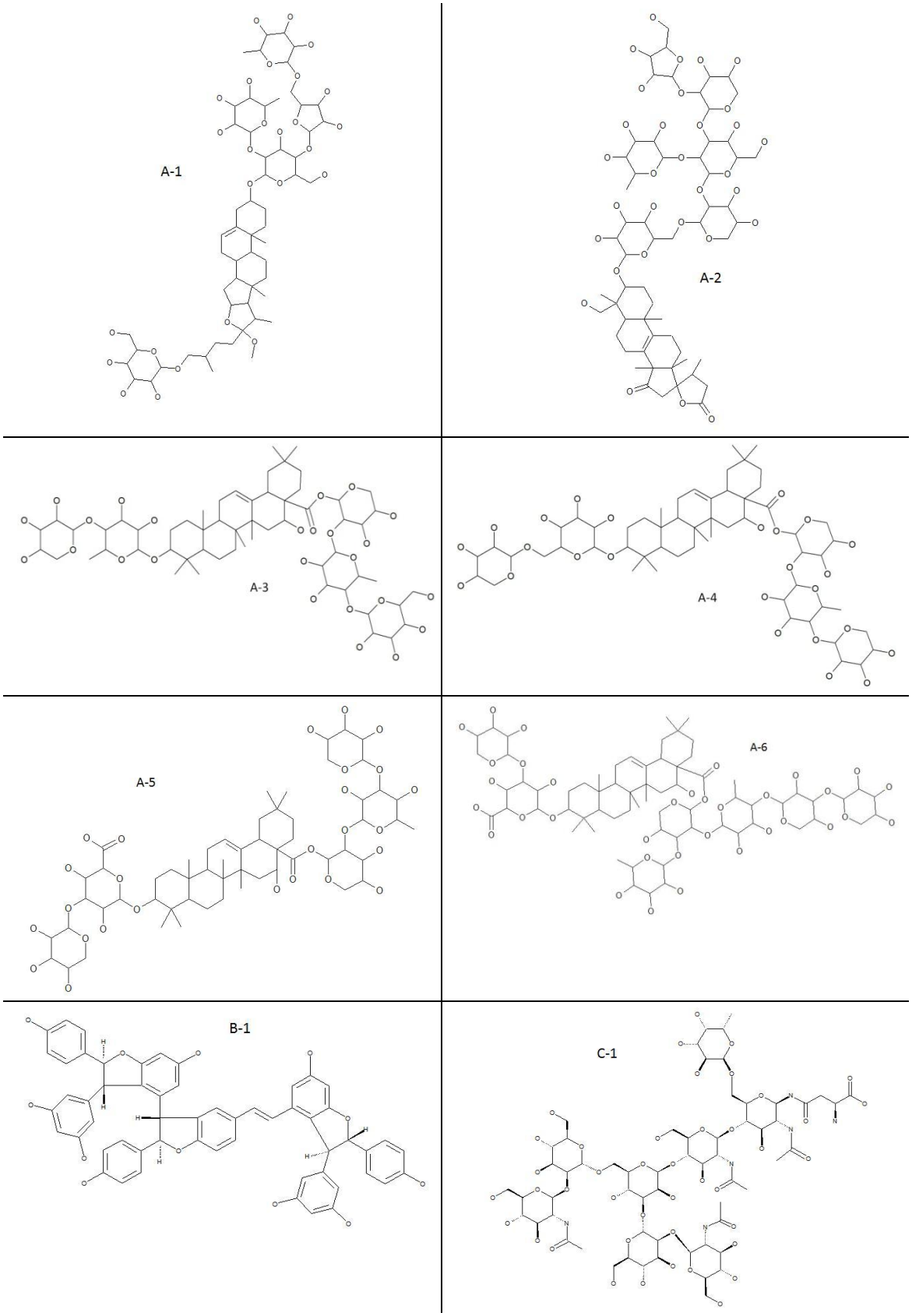


Figure 6.39: These molecules are examples of the cluster no. 1 of the classification shown in Figure 6.37. The molecules A-1 to A6 originates from the *Chapman & Hall Chemical Database*, the molecule B-1 from *the InterMed Discovery* database and the molecule C-1 from the *ChEBI* database and.

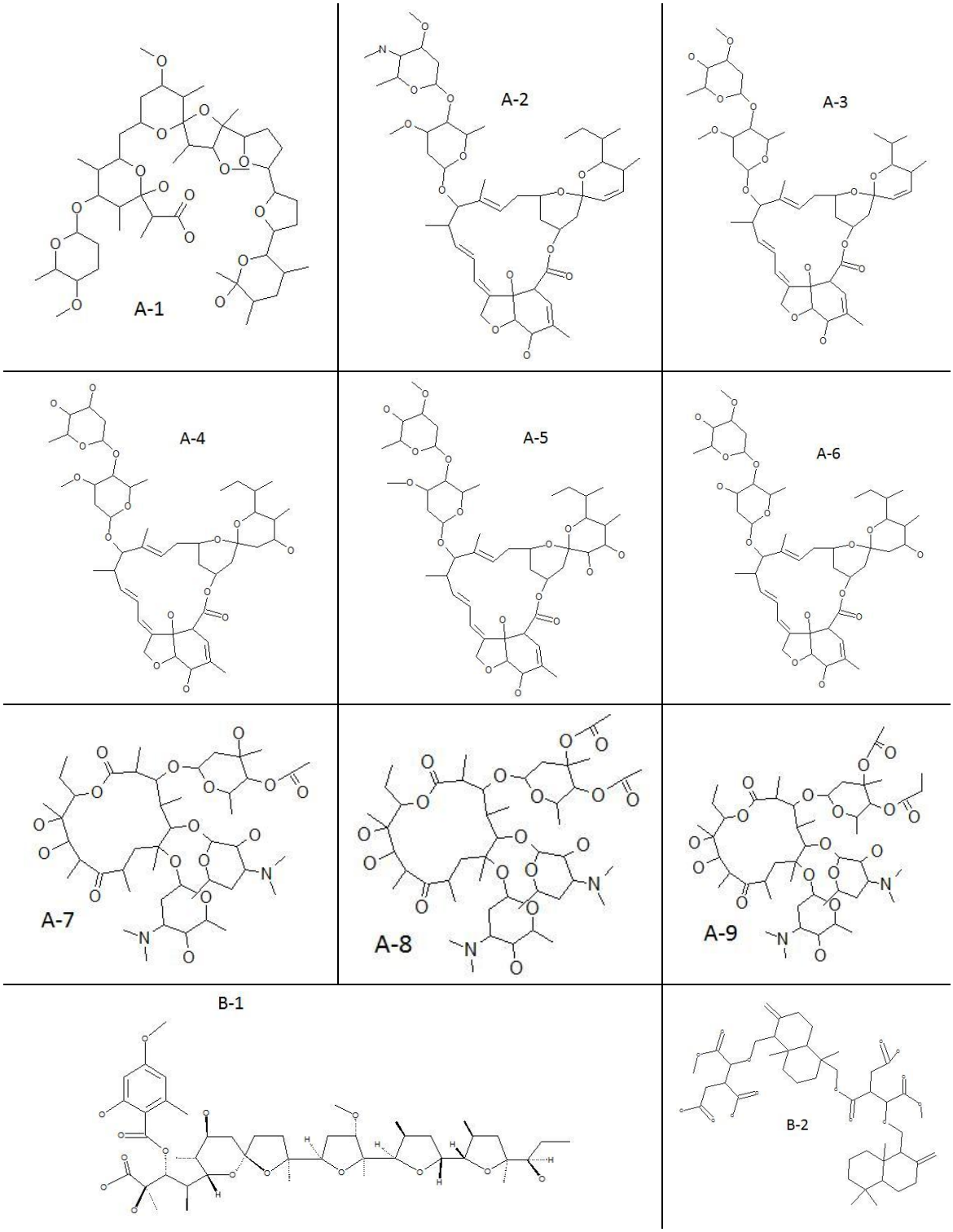


Figure 6.40: These molecules are examples of the cluster no. 3 of the classification shown in Figure 6.37. The molecules A-1 to A-9 originate from the *Chapman & Hall Chemical Database* and the molecules B-1 and B-2 from the *InterMed Discovery* database. This cluster does not contain molecules from the *ChEBI* database.

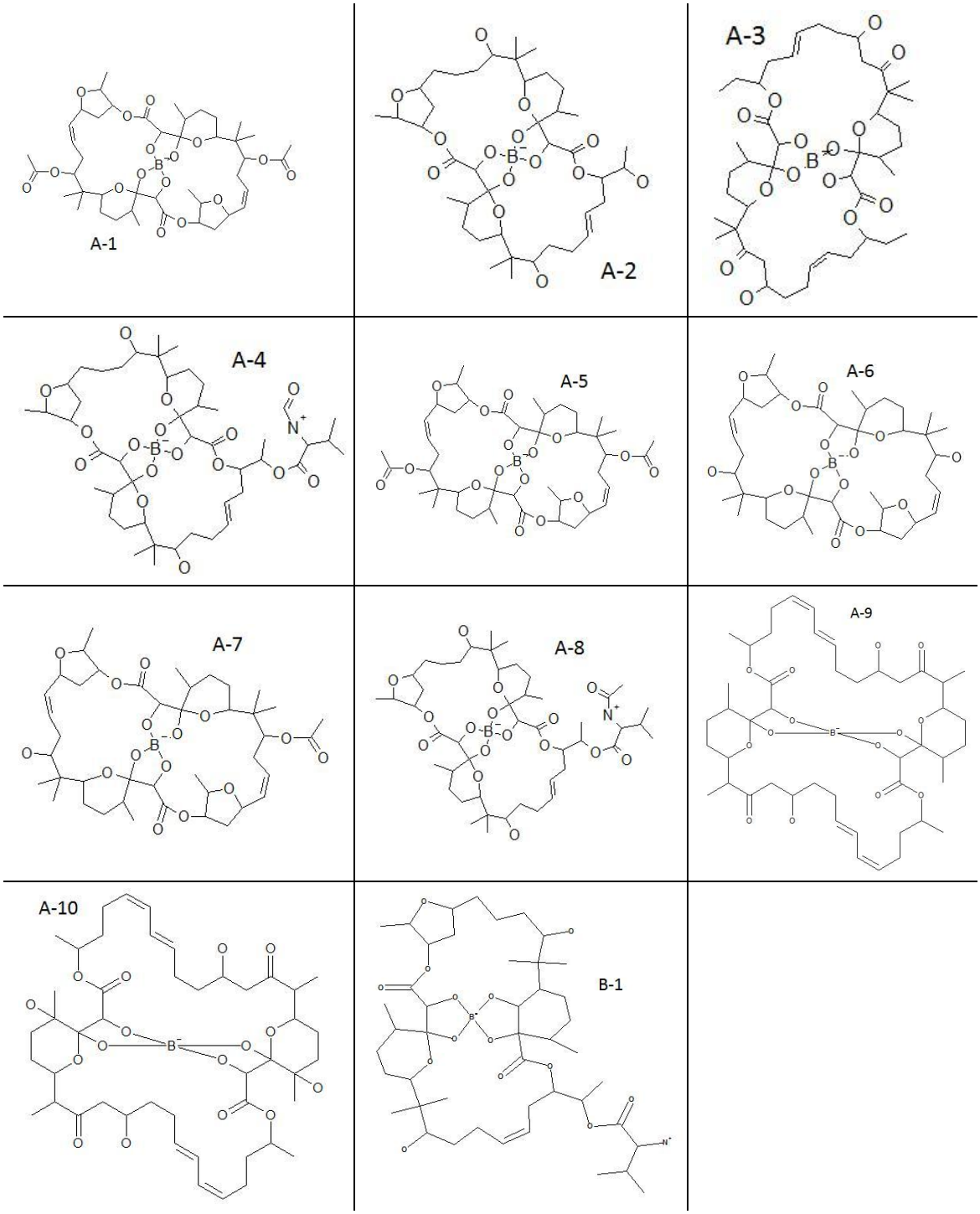


Figure 6.41: These are all molecules of the cluster no. 6 of the classification shown in Figure 6.37. The molecules A-1 to A-10 originate from the *Chapman & Hall Chemical Database* and the molecule B-1 from the *InterMed Discovery* database.

A further clustering of the cluster no. 4 from the first classification (see Figure 6.35) leads to the result shown in Figure 6.42. The original cluster shows a strong coverage of the chemical space from the *ChEBI* database and a weak coverage of the other two databases.

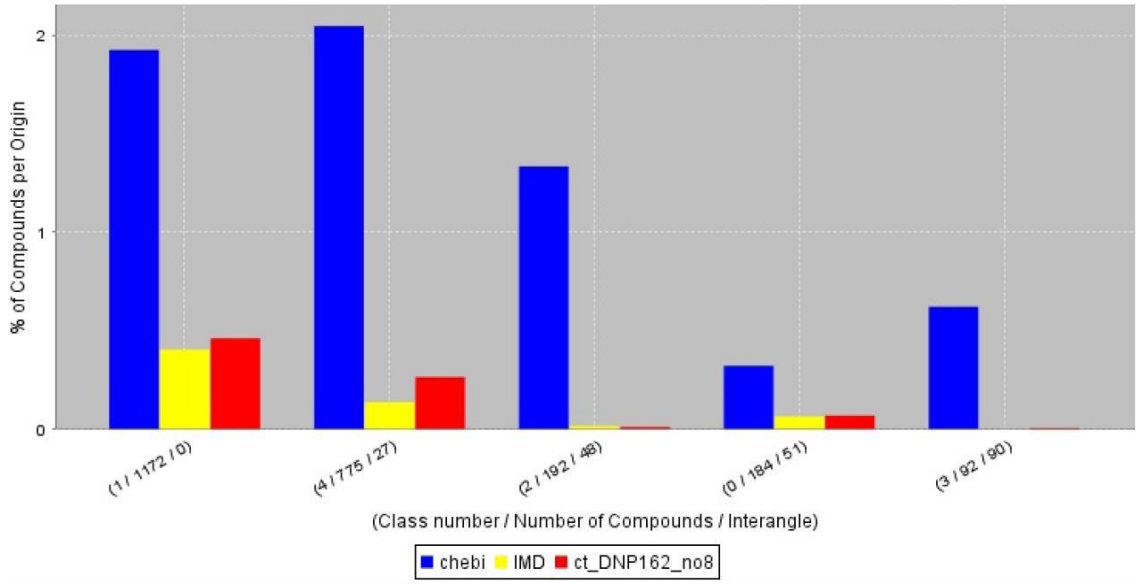


Figure 6.42: The ART 2-A classification of the cluster no. 4 from the classification result shown in Figure 6.35 leads to five different clusters.

Table 9: This table shows the detailed composition of the detected classes from the classification of the cluster no.4 of the results from Figure 6.35

Vigilance Parameter: 0.05
Number of detected classes: 5

Cluster Number	Angle to largest class	Number of Compounds in Cluster	Number of Compounds from IMD	% of IMD	Number of Compounds from ChEBI	% of ChEBI	Number of compounds from ct_DNP162_ no8	% of ct_DNP162_ no8
1	0	1172	67	0.405	238	1.924	867	0.462
4	27	775	23	0.139	253	2.046	499	0.266
2	48	192	3	0.018	165	1.334	24	0.013
0	51	184	11	0.067	40	0.323	133	0.017
3	90	92	0	0.0	77	0.623	15	0.008

This classification indicates a similar coverage of the chemical space from the *Chapman & Hall Chemical Database* and the database from *InterMed Discovery* although the *ChEBI* database has the strongest coverage within this chemical space. The Figure 6.43 shows, for example, molecules from the cluster no. 2 of the classification shown in Figure 6.42. This cluster contains almost exclusively molecules from the *ChEBI* database.

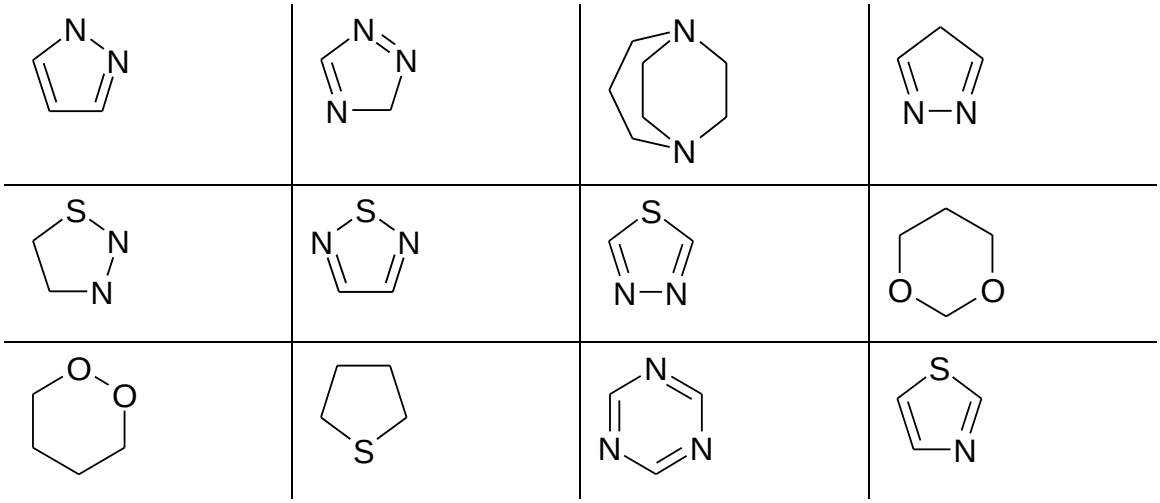


Figure 6.43: These are all molecules of the cluster no. 2 of the classification shown in Figure 6.42. All molecules originate from the *ChEBI* database.

The inspection of molecules of the constitution of the clusters leads to the conclusion that the classification produces reasonable results. The classification result correlates with the expected result because the *Chapman & Hall Chemical Database* and the database from *InterMed Discovery* contain predominantly structures of natural products whereas the *ChEBI* database contains chemical entities of biological interest.

This kind of classification provides useful details about the intellectual impact of a company's database. It shows the sides of strong coverage and, more importantly, also the areas where more analysis could be usefully performed. The Figure 6.39, for example, shows molecules from the cluster no. 1, which contains almost exclusively molecules from the *Chapman & Hall Chemical Database*. Such a white-spot-analysis allows the detection of areas within the chemical space where one database misses its coverage. The classification provides beside the white-sports, details about the coverage of the chemical space, which allows the search for alternatives for a distinct molecule.

A company such as *InterMed Discovery* contains also pharmacological annotation for many of their molecular structure. The combination of chemical-space-analysis providing distinct cluster of molecules and the pharmacological annotation of the molecules could lead to the detection of new therapeutics and be valuable source for further research and modelling. A prove of concept for building a quantitative model with supervised machine learning techniques is shown in section 6.3.

7 Conclusion & Outlook

One may disagree with the conclusions of the article “The End of Theory: The Data Deluge Makes the Scientific Method Obsolete” by *Chris Anderson* (151) in which he proclaims that the massive amount of data of the petabyte age will made the current scientific - hypothesis, model, test - approach obsolete. The massive amount of data will change the way scientists handle the unknown but it will not replace the building of models for hypotheses in order to gain new knowledge. And based on an assumed usable chemical space of 10^{60} compounds, the petabyte age will not provide enough information to fully cover the chemical space without the building of models.

In the course of this work an Open Source workflow engine for cheminformatics research was created. It enables scientists to create, run and share workflows to fit the current demands of their work. The *CDK-Taverna* plug-in provides over 160 different workers from many areas of cheminformatics. These workers allow the flexible creation of workflows and are a basis for further developments.

The availability of large chemical databases within the public domain allows scientist to perform research based on distributed knowledge. The distribution of data and the absence of standards for coding and storing chemical and biological data have lead to a curation challenge. During the work of this thesis a validation of software and data was performed (see chapter 6.1). The CDK was used to determine the atom types of molecules from different databases and the result of this work showed on one side the lack of atom types (especially of heavier atoms within the CDK) and on the other side that different databases uses different coding standards for chemical substructures like the nitro group which leads, for example, to fived bonded nitrogen’s. The next step after the detection of such a curation problem would be the building of workers which transform these molecules based on distinct rules. The transformation to a chemical structure respecting the known chemical laws and preferably chemical coding standards is involved with many different problems such as the generation of 2 or 3D coordinates and needs heavy testing because this data will be the basis for further research.

Besides the work on automated methods for data curation, the creation of a framework for performing chemical data analysis was a major goal of this thesis. For this, a set of worker was developed to support the complete data analysis cycle from the creation of data, through the storing, filtering and pre-processing to the analysis and the comparison of analysis results. The cooperation with *InterMed Discovery*, a natural product lead-discovery company, allowed the application of the developed data analysis framework to a chemical diversity analysis on the base of three chemical databases including over 200,000 molecules (see chapter 6.4). The result of the chemical diversity analysis reveals similar coverage of the chemical space from the *Chapman & Hall Chemical Database* and the proprietary database from *InterMed Discovery* whereas the composition of the *ChEBI* database covers slightly a different chemical space. This result corresponds with the expectations of the analysis because the *Chapman & Hall Chemical Database* and the database from *InterMed Discovery* contain almost exclusively structures of natural products whereas the *ChEBI* database contains different kind of chemical entities of biological interest. The chemical diversity analysis includes the detection of white-spots. These areas within the chemical space miss the coverage of one database and are good starting points for further improvements of each database. The allocation of the different clusters regarding the origin of the data reveals the expected result. Also the visual inspection of structures from a cluster showed a subjectively reasonable constitution of the clusters.

A lead-discovery company, such as *InterMed Discovery*, contains besides the chemical structure also pharmacological annotation for many molecules. The combination of chemical diversity analysis and pharmacological annotations of molecules within a cluster lead to further research and modelling activities.

Supervised machine learning techniques were used as proof of concept for quantitative modelling of adhesive polymer kinetics with the *Mathematica GNWI.CIP* package shown in chapter 6.3. This opens the perspective of an integration of high-level "experimental mathematics" into the *CDK-Taverna* based scientific pipelining.

The implemented data analysis framework could be extended at various points. The support of different pre-processing filters including algorithms for scaling the input vectors or the implementation of other data mining algorithms would be reasonable

tasks for further development. Also implementation of different result (improved) graphical display methods would be quite valuable.

The reaction enumeration workflow implemented during the work for this thesis (see chapter 6.2) is usable for building of small to midsized libraries of chemical structures based on a generic reaction. A virtual screening analysis based on such small libraries reduces the number of synthesised chemical structures and provides a possibility to reduce the cost of the drug development process. The implemented algorithm is currently limited to one generic group per reactant and two reactants per reaction. A reasonable extension of this algorithm would be a removing of the limitations.

All the functionality developed within this project is based on the *Taverna* 1 release chain. The upcoming *Taverna* 2 will be a complete redesign and the adapting of the current implementation to the new *Taverna* 2 design would be another reasonable task to be undertaken.

Besides this, an integration of *Taverna* and the *CDK-Taverna* plug-in in *Bioclipse* (101), an Open Source, visual platform for chem- and bioinformatics problems based on the *Eclipse Rich Client Platform* (RCP) would be a valuable extension to all of these projects. This would create a multifunctional combination of a workflow solution and a life science information management system.

In the course of this work, over 400 classes with more than 36,000 lines of code were developed.

8 Appendix

8.1 Scufl XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<s:scufl xmlns:s="http://org.embl.ebi.escience/xscufl/0.1alpha" version="0.2" log="0">
  <s:workflowdescription lsid="urn:lsid:www.mygrid.org.uk:operation:VI9FMF5HBQ3" author="
Tom Oinn" title="Fetch today's Dilbert comic">Use the local java plugins and some
filtering operations to fetch the comic strip image from http://www.dilbert.com</s:
workflowdescription>
  <s:processor name="dilbertURL">
    <s:stringconstant>http://www.dilbert.com/</s:stringconstant>
  </s:processor>
  <s:processor name="getPage">
    <s:local>org.embl.ebi.escience.scuflworkers.java.WebPageFetcher</s:local>
  </s:processor>
  <s:processor name="getComicStrip">
    <s:local>org.embl.ebi.escience.scuflworkers.java.WebImageFetcher</s:local>
  </s:processor>
  <s:processor name="comicURLRegex">
    <s:stringconstant>.*/archive/images/dilbert.*</s:stringconstant>
  </s:processor>
  <s:processor name="findComicURL">
    <s:local>org.embl.ebi.escience.scuflworkers.java.FilterStringList</s:local>
  </s:processor>
  <s:processor name="getImageLinks">
    <s:local>org.embl.ebi.escience.scuflworkers.java.ExtractImageLinks</s:local>
  </s:processor>
  <s:link source="dilbertURL:value" sink="getPage:url" />
  <s:link source="getPage:contents" sink="getImageLinks:document" />
  <s:link source="getImageLinks:imagelinks" sink="findComicURL:stringlist" />
  <s:link source="comicURLRegex:value" sink="findComicURL:regex" />
  <s:link source="dilbertURL:value" sink="getComicStrip:base" />
  <s:link source="findComicURL:filteredlist" sink="getComicStrip:url" />
  <s:link source="getComicStrip:image" sink="todaysDilbert" />
  <s:sink name="todaysDilbert">
    <s:metadata>
      <s:mimeTypes>
        <s:mimeType>image/*</s:mimeType>
      </s:mimeTypes>
    </s:metadata>
  </s:sink>
</s:scufl>
```

Figure 8.1: This Scufl example shows one of the examples shipped with *Taverna*. It represents a workflow, which loads the today's Dilbert comic from a website and shows the image of the comic the user as result of the workflow

8.2 Adaptive-Resonance-Theory-(ART)-2A-Algorithmus

This chapter contains algorithmic information about the ART 2-A algorithm.

8.2.1 Short Characteristic

The algorithm classifies the *n* vectors $\vec{x}_1,...,\vec{x}_n$, each contains *m* components $x_{i,1},...,x_{i,m}$ into an a priori not defined number of clusters which is defined by a vigilance parameter ρ_{max}.

8.2.2 Initial Point

The initial point contains the *n* vectors $\vec{x}_1,...,\vec{x}_n$, each vector contains *m* components $x_{i,1},...,x_{i,m}$ ($x_{i,j} \geq 0$: only positive, real numbers including Null are allowed). Each vector $\vec{x}_1,...,\vec{x}_n$ characterized one sample and each component $x_{i,1},...,x_{i,m}$ encoded one feature of the corresponding sample. The vectors could be combined to a data matrix $\underline{\underline{X}}$ with *n* rows and *m* columns whereas each row representing a sample vector.

8.2.3 Initializing

- Define the vigilance parameter ρ_{max}:

  $0 < \rho_{max} < 1$

  A ρ_{max} near "0" leads to an unsubtle classification (only few clusters), a ρ_{max} near "1" leads to an subtle classification (many clusters).

- Initialize the $m \cdot c_{cluster,max}$ elements $w_{k,j}$ of the cluster matrix $\underline{\underline{W}}$ (with $c_{cluster,max}$ rows and *m* columns): $w_{k,j} = \frac{1}{\sqrt{m}}$. The parameter $c_{cluster,max}$ is larger than the largest number of expected clusters. Each row of the matrix $\underline{\underline{W}}$ builds a cluster vector $\vec{w}_k$ each with *m* components.

- Define the parameter:

  $0 < \theta < \frac{1}{\sqrt{m}}$ (Threshold for the contrast enhancement)

  $0 < \eta << 1$ (Learning parameter, reasonable is $\eta < 0.5$, default value is 0.1)

$\alpha < \frac{1}{\sqrt{m}}$ (Scaling factor for the modification of the sample vectors)

8.2.4 Training

- Select randomly any sample vector $\vec{x}_i$ from the matrix $\underline{\underline{X}}$. Repeat this process as long as the randomly selected sample vector is not a null vector. Afterwards normalise the vector $\vec{x}_i^0 = \frac{\vec{x}_i}{|\vec{x}_i|}$; $|\vec{x}_i| = \sqrt{\sum_{j=1}^{m} x_{i,j}^2}$
- For the contrast enhancement all components of $\vec{x}_i^0$ are transformed with a non linear threshold function. Than the transformed vector $\vec{y}_i$ gets normalised again.

$$y_{i,j} = \begin{cases} x_{i,j} \text{ für } x_{i,j} > \theta \\ 0 \text{ für } x_{i,j} \le \theta \end{cases}$$

$$\vec{y}_i^0 = \frac{\vec{y}_i}{|\vec{y}_i|}$$

According to amount small components of the vector $\vec{x}_i^0$ are suppressed (noise suppression). Because of this the according to amount small but relevant components should be intensified before the contrast enhancements begin (reasonable a priori component scaling)

- If no cluster exists (1. round $c_{cluster} = 0$) will the vector $\vec{y}_i^0$ gets into the cluster matrix $\underline{\underline{W}}$ and builds the first cluster: $\vec{w}_1 = \vec{y}_i^0$; $c_{cluster}^{new} = 1$. The cluster vector $\vec{w}_1$ represents the first row of the cluster matrix $\underline{\underline{W}}$
- If cluster exists ($c_{cluster} \ge 1$) a maximal ρ_{winner} will be calculated after the following rules:

$$\rho_{winner} = \max(\rho_i)$$

$$\rho_i = \alpha \sum_{j=1}^{m} y_{i,j} \quad \text{as well as} \quad \rho_i = \vec{y}_i^0 \cdot \vec{w}_k \quad \text{mit } k = 1,\ldots, c_{cluster}$$

Keep in mind: $\rho_i = \vec{y}_i^0 \cdot \vec{w}_k = |\vec{y}_i^0| \cdot |\vec{w}_k| \cos(\phi) = \cos(\phi)$ ($\vec{y}_i^0$ and $\vec{w}_k$ are unit vectors and ϕ is the angle between these two vectors). Since all vectors are

located because of $x_{i,j} \geq 0$ within the positive quadrant is $0 \leq \phi \leq 90^\circ \Rightarrow 0 \leq \cos(\phi) \leq 1$ reasonable.

If $\rho_{winner} = \alpha \sum_{j=1}^{m} y_{i,j}$ the number of cluster gets increased by one: $c_{cluster}^{new} = c_{cluster}^{old} + 1$. The vector $\vec{y}_i^{\,0}$ gets included into the new cluster vector $\vec{w}_{c_{cluster}^{new}}$: $\vec{w}_{c_{cluster}^{new}} = \vec{y}_i^{\,0}$.

A new cluster is chosen if no assignment to an existing cluster could persuade, especially if $\rho_{max} = 0$.

$$\alpha \sum_{j=1}^{m} y_{i,j} = \sum_{j=1}^{m} y_{i,j} \cdot \alpha \approx \sum_{j=1}^{m} y_{i,j} \cdot w_{k,j}^{initial} \quad \text{with} \quad w_{k,j}^{initial} = \frac{1}{\sqrt{m}}$$

If $\rho_{winner} = \vec{y}_i^{\,0} \cdot \vec{w}_{k_{winner}}$ will ρ_{winner} be compared with ρ_{max}: for $\rho_{winner} < \rho_{max}$ gets the number of cluster increased by one: $c_{cluster}^{new} = c_{cluster}^{old} + 1$. The vector $\vec{y}_i^{\,0}$ gets included into the new cluster vector $\vec{w}_{c_{cluster}^{new}}$: $\vec{w}_{c_{cluster}^{new}} = \vec{y}_i^{\,0}$. For $\rho_{winner} \geq \rho_{max}$ the number of cluster remain constant and the "winning" cluster vector $\vec{w}_{k_{winner}}$ gets modified as follows:

$$\vec{w}_{k_{winner}}^{new} = \vec{s}$$

$$\vec{s} = \frac{\vec{t}}{|\vec{t}|}$$

$$\vec{t} = \vec{u} + (1 - \eta)\vec{w}_{k_{winner}}^{old}$$

$$\vec{u} = \eta \frac{\vec{v}}{|\vec{v}|}$$

$$v_j = \begin{cases} y_{i,j}^0 \text{ für } w_{k_{winner},j}^{old} > \theta \\ 0 \text{ für } w_{k_{winner},j}^{old} \leq \theta \end{cases}$$

The learning parameter η defines the incremental learning. For $\eta = 0$ remains $\vec{w}_k$ always constant. No incremental learning takes places. For $\eta = 1$ the $\vec{w}_k$ will be completely forgotten and only the new vector $\vec{y}_i^{\,0}$ will be learned. The learning parameter η interferes between the two extremes.

An attribute never gets learned again if this attribute once drops under the threshold θ. This is the responsibility of the threshold vector $\vec{v}$ ("stabilizing").

- All training steps are repeated until the cluster matrix $\underline{\underline{W}}$ shows no significant change after one epoch (this means after all *n* sample vectors $\vec{x}_i$ in a random order once goes through the training steps). This means that the single cluster vectors $\vec{w}_k$ "practically" remain constant. Alternatively all training steps could be repeated until the assignment of the sample vectors $\vec{x}_i$ to their respective clusters remains unchanged between two epochs.

8.2.5 Clustering

- Select a sample vector $\vec{x}_i$ from the matrix $\underline{\underline{X}}$. If it is a null vector adjust it to the null cluster and select another sample vector, otherwise normalise it: $\vec{x}_i^{\,0} = \frac{\vec{x}_i}{|\vec{x}_i|}$
- Transform $\vec{x}_i^{\,0}$ with a non-linear threshold function and normalise the transformed vector $\vec{y}_i$ again.

$$y_{i,j} = \begin{cases} x_{i,j} \text{ für } x_{i,j} > \theta \\ 0 \text{ für } x_{i,j} \le \theta \end{cases}$$

$$\vec{y}_i^{\,0} = \frac{\vec{y}_i}{|\vec{y}_i|}$$

- Assign the maximal ρ_{winner} using the following rules:

  $\rho_{winner} = \max(\rho_i)$

  $\rho_i = \vec{y}_i^{\,0} \cdot \vec{w}_k$ with $k = 1,..., c_{cluster}$
- The sample vector $\vec{x}_i$ belongs to the cluster no. k_{winner} which is defined through $\rho_{winner} = \vec{y}_i^{\,0} \cdot \vec{w}_{k_{winner}}$.

8.2.6 Comparison

Compared with the "kth-nearest-neighbour"-clustering the ART 2-A algorithm is computationally less demanding.

- The "k[th]-nearest-neighbour"-clustering requires $\frac{(n^2 - n)}{2}$ pair wise vector comparisons (each vector is compared with every other vector)
- The ART 2-A algorithm needs only $n \cdot c_{cluster}$ pair wise vector comparisons, whereas the number of cluster has the dimension $c_{cluster} = O(10)$.

For a clustering of 500 vectors this result in 125000 pair wise vector comparisons for the "k[th]-nearest-neighbour" but only in about 5000 for the ART 2-A algorithm (reduction of factor 25).

8.3 CDK-Taverna Plug-in Configuration files

```
<plugins>
  <plugin>cdk-taverna-0.3.0.1.xml</plugin>
  <plugin>cdk-taverna-0.3.0.2.xml</plugin>
  <plugin>cdk-taverna-0.3.1.xml</plugin>
  <plugin>cdk-taverna-0.3.1.1.xml</plugin>
  <plugin>cdk-taverna-0.4.0.0.xml</plugin>
  <plugin>cdk-taverna-0.4.0.2.xml</plugin>
  <plugin>cdk-taverna-0.4.1.0.xml</plugin>
  <plugin>cdk-taverna-0.5.0.0.xml</plugin>
  <plugin>cdk-taverna-0.5.1.0.xml</plugin>
</plugins>
```

Figure 8.2: The pluginlist.xml file contains a list of available plug-in configuration files.

```
<plugin>
  <name>CDK-Taverna</name>
  <description>This Plug-in provides processors from
        the CDK-Taverna Project.</description>
  <identifier>
    org.openscience.cdk.applications.taverna
  </identifier>
  <version>0.5.1.0</version>
  <provider>CDK-Taverna Release Repository</provider>
  <repositories>
    <repository>http://cdk-taverna.de/m2/release_repository/</repository>
  </repositories>
  <profile>
    <artifact groupId="org.openscience.cdk.applications.taverna"
        artifactId="cdk-taverna" version="0.5.1.0"/>
  </profile>
  <taverna>
    <version>1.7.1</version>
  </taverna>
</plugin>
```

Figure 8.3: The example of a plug-in configuration file contains the available information about the installable plug-in. In this case, it is the configuration file of the *CDK-Taverna* plug-in version 0.5.1.0 for *Taverna* version 1.7.1. The file name of this configuration file is cdk-taverna-0.5.1.0.xml.

8.4 Worker of the CDK-Taverna Plug-in

Table 10: List of all workers of the *CDK-Taverna* plug-in grouped by their functionality

Workername / Group	**Description**
Basics	
PerceiveMM2AtomTypes	Perceive MM2 atom types
PerceiveMMFF94AtomTypes	Perceive MMFF94 atom types
PerceiveAtomType	Perceive atom types
SubstructureFilter	Substructure Filter
DetectHueckelAromaticity	Hueckel Aromaticity Detector
StructureDiagramGenerator	Structure Diagram Generator
FingerprintCalculator	Calculates Fingerprint for a given CMLChemfile
CreatePDFWith2DStructures	Create PDF File with 2D Structures
CreateJPGWith2DStructures	Create JPG File with 2D Structures
CreatePNGWith2DStructures	Create PNG File with 2D Structures
BioclipseResultViewer	Start Bioclipse
Model3DBuildersWithMM2ForceField	Model3DBuildersWithMM2ForceField
ReactionEnumerator	ReactionEnumerator

Classificator	
ART2AClassificatorWorker	ART2A Classificator
GetART2AClassificationResultAsCSV	Get classification result
GetART2AClassificationResultAsPDF	Get classification result as PDF
GetInterAngleBetweenClassesAsCSV	Get interangle between classes
LeafOneOutIteratorInitializer	Initialise leaf one out iterator
LeafOneOutIteratorGetNextFingerprintItemList	Get next fingerprint item from leaf one out iterator
LeafOneOutIteratorHasNext	Has next from the leaf one out iterator
CompareART2AClassificationResults	Compare ART2A classification results
GetMoleculeIDsForGivenClasses	Get the molecule ids for a given class
GenerateRandomCentroidVectors	Get random centroid vectors for testing
GetART2AClassificationResultWithDifferentOriginsAsPDF	Get classification result consider different origins as PDF
GetART2AClassificationResultWithDifferentOriginsAsCSV	Get classification result consider different origins as CSV
Database	
InsertMoleculeIntoDB	Insert Molecule into database
GetMoleculesFromDB	Get Molecules from database
InsertQSARResultsIntoDB	Insert QSAR Results into database
UpdateQSARResultsIntoDB	Update QSAR Results within the database
GetQSARVectorFromDB	Get QSAR vector from database
GetMolecularWeightFromDB	Get Molecular Weight from database
GetMoleculesFromDB	Get molecules from database
GetMoleculeIDsAndOriginFromDB	Get molecule ID's and origin from database
GetMoleculesFromDBForSubstructure	Substructure Search of molecules on the database
GetSelectedMoleculesFromDB	Get selected molecules from database for given molecule ids
Database-Iterativ	
IterativeGetMoleculeFromDB	Get Molecule from database
IterativeHasNextMoleculeFromDB	Has Next Molecule from database
IterativeMoleculeFromDBReader	Iterative Molecule from database reader
Database-IO	
ReadMDLSDFileAsDatabaseInput	Read MDL SD File as DB input
InChITools	
InChIParser	Parse InChI
InChIGeneratorWorker	Generate InChI
IO	
ConvertToCMLString	Convert to CML String
WriteToCMLFile	Write CMLChemFile to File
WriteToMDLMolFile	Write MDL Mol File to File
ReadMDLMolFile	Read MDL Mol File
ReadMDLSDFile	Read MDL SD File

FileReader	File Reader
TextFileReader	Textfile Reader
ReadMDLRXNFile	Read MDL RXN File
ReadMDLRXNV3000File	Read MDL RXN V3000 File
ReadSMILESFromFile	Read SMILES from File
ConvertGZFilesToXMLFiles	Converts GZ files to XML files
ConvertXMLFilesToGZFiles	Converts XML files to GZ files
FileWriter	Write CSV / TXT Files
IO-Iterative	
IteratingFileReader	IteratingFileReader
IteratingFileReaderGetContent	GetContent
IteratingFileReaderHasNext	HasNext
IterativeFileWriter	Iterative File Writer
QSAR	
QSARDescriptor	QSAR worker
VectorGenerator	Vector Generator
CSVGenerator	Generates a CSV(Comma Separated Value)
VectorGeneratorForDBInput	Vector Generator For Database Input
QSAR_Atomic	
AtomDegree	AtomDegree-Descriptor
AtomHybridization	AtomHybridization-Descriptor
AtomHybridizationVSEPR	AtomHybridizationVSEPR-Descriptor
AtomValence	AtomValence-Descriptor
BondsToAtom	BondsToAtom-Descriptor
CovalentRadius	CovalentRadius-Descriptor
DistanceToAtom	DistanceToAtom-Descriptor
EffectiveAtomPolarizability	EffectiveAtomPolarizability-Descriptor
InductiveAtomicHardness	InductiveAtomicHardness-Descriptor
InductiveAtomicSoftness	InductiveAtomicSoftness-Descriptor
IPAtomicHOSE	IPAtomicHOSE-Descriptor
IPAtomicLearning	IPAtomicLearning-Descriptor
IsProtonInAromaticSystem	IsProtonInAromaticSystem-Descriptor
IsProtonInConjugatedPiSystem	IsProtonInConjugatedPiSystem-Descriptor
PartialPiCharge	PartialPiCharge-Descriptor
PartialSigmaCharge	PartialSigmaCharge-Descriptor
PartialTChargeMMFF94	PartialTChargeMMFF94-Descriptor
PartialTChargePEOE	PartialTChargePEOE-Descriptor
PeriodicTablePosition	PeriodicTablePosition-Descriptor
PiElectronegativity	PiElectronegativity-Descript
ProtonTotalPartialCharge	ProtonTotalPartialCharge-Descriptor
RDFProton_G3R	RDFProton_G3R-Descriptor
RDFProton_GDR	RDFProton_GDR-Descriptor
RDFProton_GHR_topol	RDFProton_GHR_topol-Descriptor

RDFProton_GHR	RDFProton_GHR-Descriptor
RDFProton_GSR	RDFProton_GSR-Descriptor
SigmaElectronegativity	SigmaElectronegativity-Descriptor
VdWRadius	VdWRadius-Descriptor
QSAR_AtomPair	
PiContactDetection	PiContactDetection-Descriptor
QSAR_Bond	
BondPartialPiCharge	BondPartialPiCharge-Descriptor
BondPartialSigmaCharge	BondPartialSigmaCharge-Descriptor
BondPartialTCharge	BondPartialTCharge-Descriptor
BondSigmaElectronegativity	BondSigmaElectronegativity-Descriptor
IPBondLearning	IPBond-Descriptor
MassNumberDifference	MassNumberDifference-Descriptor
QSAR_Model_Weka	
SimpleKMeansWorker	Simple KMeans Clusterer
EMClusterWorker	EM Clusterer
ExtractClusterResultWorker	Extract Cluster Result Worker
QSAR_Molecular	
AminoAcidCount	AminoAcidCount-Descriptor
APol	APol-Descriptor
AromaticAtomsCount	AromaticAtomsCount-Descriptor
AromaticBondsCount	AromaticBondsCount-Descriptor
AtomCount	AtomCount-Descriptor
Autocorrelation_Charge	Autocorrelation_Charge-Descriptor
Autocorrelation_Mass	Autocorrelation_Mass-Descriptor
Autocorrelation_Polarizability	Autocorrelation_Polarizability-Descriptor
BCUT	BCUT-Descriptor
BondCount	BondCount-Descriptor
BPol	BPol-Descriptor
ChiChain	ChiChain-Descriptor
ChiCluster	ChiCluster-Descriptor
ChiPath	ChiPath-Descriptor
ChiPathCluster	ChiPathCluster-Descriptor
CPSA	CPSA-Descriptor
EccentricConnectivityIndex	EccentricConnectivityIndex-Descriptor
FragmentComplexity	FragmentComplexity-Descriptor
GravitationalIndex	GravitationalIndex-Descriptor
HBondAcceptorCount	HBondAcceptorCount-Descriptor
HBondDonorCount	HBondDonorCount-Descriptor
IPMolecularLearning	IPMolecularLearning-Descriptor
KappaShapeIndices	KappaShapeIndices-Descriptor
LargestChain	LargestChain-Descriptor
LargestPiSystem	LargestPiSystem-Descriptor

LengthOverBreadth	LengthOverBreadth-Descriptor
LongestAliphaticChain	LongestAliphaticChain-Descriptor
MDE	MDE-Descriptor
MomentOfInertia	MomentOfInertia-Descriptor
PetitjeanNumber	Calculate PetitjeanNumber-Descriptor
PetitjeanShapeIndex	PetitjeanShapeIndex-Descriptor
RotatableBondsCount	RotatableBondsCount-Descriptor
RuleOfFiveFilter	Rule-of-Five Filter
TPSA	TPSA-Descriptor
VAdjMa	VAdjMa-Descriptor
Weight	Weight-Descriptor
WeightedPath	WeightedPath-Descriptor
WHIM	WHIM-Descriptor
WienerNumbers	WienerNumbers-Descriptor
XLogP	XLogP-Descriptor
ZagrebIndex	ZagrebIndex-Descriptor
qsardescriptorsproteinTaeAminoAcid	TaeAminoAcid-Descriptor
SmilesTools	
SMILESGenerator	Generates SMILES
SMILESParser	Parse SMILES
Tools	
RemoveMOLFilesWithoutStructureFromList	Remove MOL Files without Structures from List
ExtractIDFromMolFile	Extract org IDs from MOL File (EXNO)
ExtractIntermedIDFromMolFile	Extract org. IMD IDs from MOL File (IMD-No)
ImplicitHydrogenAdder	Add the implicit hydrogens to each molecule
TagMolecules	Tag molecules
ExtractDatabaseIDFromMolecule	Extract the database ID from the molecule
GetMolecularWeightDistribution	Get a molecular weight distribution
ConvertFingerprintItemListToCSV	Convert Fingerprint item list to csv
ConvertFingerprintItemListToMathematicaDataMatrix	Convert Fingerprint item list to Mathematica data vector
ScaleFingerprintItemArrayBetweenZeroAndOne	Scale Fingerprint item list to values between 0 and 1
Tools-Chebi	
chebiExtractDataFromChebiStructureTSV	Extract Data from Chebi Structure TSV
chebiRemoveAllNonMolFilesFromChebiStructureExtraction	Remove all non MDL Mol entries from list

8.5 Descriptors Used For Data Analysis

The following list contains the descriptors and the number of values of each descriptor used for ART 2-A classification. The descriptors listed here are those that remain after the pre-processing of the descriptor values. Thus is not a complete list of the calculated descriptor values because not all descriptors could be calculated for all molecules.

Table 11: This table shows the descriptors and the number of values of each descriptor used for the classification within the work of this project.

Descriptors	**Number of Values**
Amino Acids Count	19
Aromatic Atoms Count	1
Aromatic Bonds Count	1
Atom Count	1
Auto Correlation Mass	5
Auto Correlation Polarizability	5
Bond Count	1
Eccentric Connectivity Index	1
h-Bond Acceptors	1
h-Bond Donors	1
Largest Chain	1
Largest Pi System	1
Longest Aliphatic Chain	1
MDE	18
Nila Complexity	1
Petitjean Number	1
Rotatable Bonds Count	1
vAdjMa	1
Weighted Path	5
Wiener Numbers	2
xlogP	1
Zagreb Index	1

.

9 References

1. **Marris, Emma.** Chemistry society goes head to head with NIH in fight over public database. *Nature.* 09. 06 2005, Bd. 435, 7043, S. 718-719.

2. **Irwin, J. J. und Shoichet, B. K.** ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *Journal of Chemical Information and Modeling.* 45, 2005, 1, S. 177-182.

3. **Degtyarenko, Kirill, et al.** ChEBI: a database and ontology for chemical entities of biological interest. *Nucl. Acids Res.* 36, 2008, 1, S. 344-350.

4. **Chen, Jonathan, et al.** ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinformatics.* 2005, Bd. 21, 22, S. 4133-4139.

5. **DeLano, Warren L.** The case for open-source software in drug discovery. 10, 1. 2 2005, 3, S. 213-217.

6. **Guha, R., et al.** The Blue Obelisk-Interoperability in Chemical Informatics. *Journal of Chemical Information and Modeling.* 46, 2006, 3, S. 991-998.

7. **Schreiber, Stuart L.** Target-Oriented and Diversity-Oriented Organic Synthesis in Drug Discovery. *Science.* 2000, Bd. 287, 5460, S. 1964-1969.

8. **Berman, Helen M., et al.** The Protein Data Bank. *Nucl. Acids Res.* 2000, Bd. 28, 1, S. 235-242.

9. **Ballester, P. J.** Ultrafast shape recognition to search compound databases for similar molecular shapes. *Journal of computational chemistry.* 10, 2007, Bd. 28, 10, S. 1711-1723.

10. **Murray-Rust, P. und Rzepa, H. S.** Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles. *Journal of Chemical Information and Computer Sciences.* 1999, Bd. 39, 6, S. 928-942.

11. **Coalition, Workflow Management.** [Online] [Zitat vom: 13. 09 2008.] http://www.wfmc.org.

12. **Leymann, Frank und Roller, Dieter.** *Production workflow: Concepts and techniques.* Upper Saddle River, NJ : Prentice Hall, 2000.

13. **Taylor, Ian J., et al.** *Workflows for e-Science, Scientific Workflows for Grids.* London : Springer-11645 /Dig. Serial], 2007.

14. **Stallman, Richard.** http://www.gnu.org/. *Why "Open Source" misses the point of Free Software.* [Online] 2007. [Zitat vom: 17. 09 2008.] http://www.gnu.org/philosophy/open-source-misses-the-point.html.

15. **Initiative, Open Source.** Open Source Initiative. [Online] 2008. [Zitat vom: 17. 09 2008.] http://opensource.org/docs/osd.

16. **Raymond, Eric S.** *The cathedral and the bazaar.* Beijing, Köln : O'Reilly, 2000.

17. *The OpenScience Project.* [Online] [Zitat vom: 18. 09 2008.] http://www.openscience.org.

18. **Heise Online.** Heise Onlie. [Online] 01. 09 2008. [Zitat vom: 10. 10 2008.] http://www.heise.de/newsticker/Erste-Open-Source-Professur-ausgeschrieben--/meldung/115276.

19. **Munos, Bernard.** Can open-source R&D reinvigorate drug research? *Nat Rev Drug Discov.* 5, 2006, 9, S. 723-729.

20. **Drews, Jürgen.** *In quest of tomorrow's medicines.* New York, NY : Springer, 1999.

21. **Bain & Company.** HAS THE PHARMACEUTICAL BLOCKBUSTER MODEL GONE BUST? [Online] 08. 12 2003. [Zitat vom: 20. 09 2008.] http://www.bain.com/bainweb/publications/printer_ready.asp?id=14243.

22. Tufts Center for the Study of Drug Development. [Online] [Zitat vom: 20. 09 2008.] http://csdd.tufts.edu.

23. **Hope, Janet.** Pharmaforschung mit Open-Source-Methoden. [Buchverf.] B. Lutterbeck, M. Bärwolff und R. A. Gehring. *Open Source Jahrbuch 2007.* s.l. : Lehmanns Media - LOB.de, 2007, S. 73-85.

24. Medicines for Malaria Venture (MMV). [Online] [Zitat vom: 20. 09 2008.] http://www.mmv.org.

25. International HapMap Project. [Online] [Zitat vom: 21. 09 2008.] http://hapmap.org.

26. The Tropical Disease Initiative. [Online] [Zitat vom: 21. 09 2008.] http://tropicaldisease.org/.

27. **EMBL's European Bioinformatics Institute.** Open access to large-scale drug discovery data. [Online] 23. 07 2008. [Zitat vom: 21. 09 2008.] http://www.ebi.ac.uk/Information/News/pdf/Press23July08.pdf.

28. **Harding, C.** The Open Group. *Definition of SOA.* [Online] 02. 06 2006. [Zitat vom: 27. 09 2008.] http://opengroup.org/projects/soa/doc.tpl?gdid=10632.

29. **OASIS.** [Online] [Zitat vom: 10. 10 2008.] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

30. **He, Hao.** XML.com. *What Is Service-Oriented Architecture.* [Online] 30. 09 2003. [Zitat vom: 10. 10 2008.] http://webservices.xml.com/lpt/a/1292.

31. **Erl, Thomas.** *Soa: Principles of Service Design.* s.l. : Prentice Hall, 2007.

32. —. SOA Principles . [Online] [Zitat vom: 28. 09 2008.] http://www.soaprinciples.com/.

33. **Dijkstra, Edsger W.** *Selected writings on computing.* New York u.a. : Springer, 1982.

34. **Sourceforge.** Sourceforge.net. [Online] [Zitat vom: 02. 12 2008.] http://sourceforge.net.

35. **Warr, Wendy A.** *Integration, Analysis and Collaboration. An Update on Workflow and Pipelining in Cheminformatics.* s.l. : Wendy Warr & Associates , 2007.

36. [Online] [Zitat vom: 29. 09 2008.] http://accelrys.com/.

37. [Online] [Zitat vom: 29. 09 2008.] http://www.inforsense.com/.

38. **ChemAxon.** [Online] [Zitat vom: 28. 09 2008.] http://www.chemaxon.com/.

39. **Daylight Chemical Information Systems, Inc.** [Online] http://www.daylight.com/.

40. **Symyx.** [Online] [Zitat vom: 28. 09 2008.] http://www.mdli.com/.

41. **Molecular Networks.** [Online] [Zitat vom: 28. 09 2008.] http://www.molecular-networks.com/.

42. **Tripos.** [Online] [Zitat vom: 28. 09 2008.] http://www.tripos.com/.

43. **KNIME.** [Online] [Zitat vom: 29. 11 2008.] http://knime.org/.

44. **Eclipse.** [Online] [Zitat vom: 30. 09 2008.] http://www.eclipse.org/.

45. **Schrödinger.** [Online] [Zitat vom: 30. 09 2008.] http://www.schrodinger.com/.

46. **Kepler.** [Online] [Zitat vom: 01. 10 2008.] http://www.kepler-project.org.

47. **Taverna.** [Online] [Zitat vom: 03. 10 2008.] http://taverna.sourceforge.net/.

48. **Oinn, Tom, et al.** Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics.* 2004, Bd. 20, 17, S. 3045-3054.

49. **Hull, Duncan, et al.** Taverna: a tool for building and running workflows of services. *Nucl. Acids Res.* 2006, 34, S. W729-732.

50. **Stevens, Robert, et al.** Building ontologies in DAML + OIL. *Comparative and Functional Genomics.* 4, 2003, 1, S. 133-141.

51. **Oinn, Tom, et al.** Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience.* 10, 2006, 18, S. 1067-1100.

52. **Wolstencroft, K., et al.** Panoply of Utilities in Taverna. *e-Science and Grid Computing, International Conference on.* 2005, S. 156-162.

53. **Kuhn, Thomas.** myExperiment. [Online] 29. 08 2008. [Zitat vom: 10. 10 2008.] http://www.myexperiment.org/workflows/385.

54. *myExperiment: social networking for workflow-using e-scientists.* **Goble, Carole Anne und Roure, David Charles De.** New York, NY, USA : ACM, 2007. WORKS '07: Proceedings of the 2nd workshop on Workflows in support of large-scale science.

55. **myexperiment.** myexperiment.org. [Online] http://www.myexperiment.org/.

56. **Roure, David de, Goble, Carole und Stevens, Robert.** The design and realisation of the Virtual Research Environment for social sharing of workflows. *Future Generation Computer Systems.* 2008, In Press, Corrected Proof.

57. **O'Reilly.** What Is Web 2.0. [Online] 30. 09 2005. [Zitat vom: 09. 10 2008.] http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

58. **Steinbeck, C., et al.** The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. *Journal of Chemical Information and Computer Sciences.* 2, 2003, 43, S. 493-500.

59. **Steinbeck, Christoph, et al.** Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics. *Current Pharmaceutical Design.* 2006, 12, S. 2111-2120(10).

60. **Krause, S., Willighagen, E. L. und Steinbeck, C.** JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures. *Molecules.* 2000, 5, S. 93-98.

61. **Helson, H.E.** Structure Diagram Generation. *Rev. Comput. Chem.* 1999, 13, S. 313-395.

62. **Steinbeck, C.** SENECA: A Platform-Independent, Distributed, and Parallel System for Computer-Assisted Structure Elucidation in Organic Chemistry. *Journal of Chemical Information and Computer Sciences.* 6, 2001, 41, S. 1500-1507.

63. **Faulon, J. -L.** Stochastic Generator of Chemical Structure. 2. Using Simulated Annealing To Search the Space of Constitutional Isomers. *Journal of Chemical Information and Computer Sciences.* 4, 1996, 36, S. 731-740.

64. **Figueras, J.** Ring Perception Using Breadth-First Search. *Journal of Chemical Information and Computer Sciences.* 5, 1996, 36, S. 986-991.

65. **Hanser, T., Jauffret, P. und Kaufmann, G.** A New Algorithm for Exhaustive Ring Perception in a Molecular Graph. *Journal of Chemical Information and Computer Sciences.* 6, 1996, 36, S. 1146-1152.

66. **Symyx.** Symyx. *Description of several chemical structure file formatis used by computer programs developed at MDL.* [Online] [Zitat vom: 14. 10 2008.] http://www.mdl.com/support/knowledgebase/faqs/faq_ib_27.jsp.

67. **Worldwide Protein Databank.** Atomic Coordinate Entry Format Description. [Online] [Zitat vom: 14. 10 2008.] http://www.wwpdb.org/documentation/format32/v3.2.html.

68. **Weininger, David.** SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences.* 1, 1988, 28, S. 31-36.

69. —. SMILES 2. Algorithm for Generation of Unique SMILES Notation. *Journal of Chemical Information and Computer Science.* 2, 1989, 29, S. 97-101.

70. *An Open Standard for Chemical Structure Representation - The IUPAC Chemical Identifier.* **Stein, Stephen E., Heller, Stephen R. und Tchekhovskoi, Dmitrii.** Nîmes, France, 19 - 22 October 2003 : s.n., 2003. Nimes International Chemical Information Conference Proceedings. S. 131-143. 187369993X.

71. **Daylight Chemical Information Systems, Inc.** Daylight Theory Manual. *6. Fingerprints - Screening and Similarity.* [Online] [Zitat vom: 14. 10 2008.] http://www.daylight.com/dayhtml/doc/theory/theory.finger.html.

72. **Hall, L. H. und Kier, L. B.** Electrotopological State Indices for Atom Types: A Novel Combination of Electronic, Topological, and Valence State Information. *Journal of Chemical Information and Computer Science.* 1995, 35, S. 1039-1045.

73. **QSAR Project.** [Online] [Zitat vom: 18. 10 2008.] http://qsar.sourceforge.net/.

74. **Meiler, Jens.** PROSHIFT: Protein chemical shift prediction using artificial neural networks. *Journal of Biomolecular NMR.* 1, 2003, 26, S. 25-37.

75. **Aires-de-Sousa, J., Hemmer, M. C. und Gasteiger, J.** Prediction of 1H NMR Chemical Shifts Using Neural Networks. *Analytical Chemistry.* 1, 2002, 74, S. 80-90.

76. **Lipinsk, C.A, et al.** Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews.* 1997, 23, S. 3-25.

77. **Wang, R., Fu, Y. und Lai, L.** A New Atom-Additive Method for Calculating Partition Coefficients. *Journal of Chemical Information and Computer Sciences.* 3, 1997, 37, S. 615-621.

78. **Kier, Lemont B und Hall, Lowell H.** *Molecular connectivity in structure-activity analysis.* New York : Wiley, 1986. 0471909831.

79. **Kier, Lemont B. und Hall, Lowell H.** Molecular connectivity VII: Specific treatment of heteroatoms. *Journal of Pharmaceutical Sciences.* 12, 1976, 65, S. 1806-1809.

80. **Kier, Lemont B., et al.** Molecular connectivity I: Relationship to nonspecific local anesthesia. *Journal of Pharmaceutical Sciences.* 12, 1975, 64, S. 1971-1974.

81. **Wiener, H.** Correlation of Heat of Isomerization and Difference in Heat of Vaporization of Isomers Among Paraffin Hydocarbons. *J. Am. Chem. Soc.* 1947, 67, S. 17-20.

82. **Gutman, I., et al.** Graph theory and molecular orbitals. XII. Acyclic polyenes. *The Journal of Chemical Physics.* 9, 1975, 62, S. 3399-3405.

83. **Petitjean, Michel.** Applications of the radius-diameter diagram to the classification of topological and geometrical shapes of chemical compounds. *Journal of Chemical Information and Computer Sciences.* 4, 1992, 32, S. 331-337.

84. **Kier, Lemont B.** A Shape Index from Molecular Graphs. *Quantitative Structure-Activity Relationships.* 3, 1985, 4, S. 109-116.

85. —. Shape Indexes of Orders One and Three from Molecular Graphs. *Quantitative Structure-Activity Relationships.* 1, 1986, 5, S. 1-7.

86. —. Distinguishing Atom Differences in a Molecular Graph Shape Index. *Quantitative Structure-Activity Relationships.* 1, 1986, 5, S. 7-12.

87. **Katritzky, A. R., et al.** Correlation of Boiling Points with Molecular Structure. 1. A Training Set of 298 Diverse Organics and a Test Set of 9 Simple Inorganics. *J. Phys. Chem.* 24, 1996, 100, S. 10400-10407.

88. **Goldstein, H.** *Classical Mechanics.* Reading, MA : Addison Wesley, 1950.

89. **Pearlman, R. S. und Smith, K. M.** Metric Validation and the Receptor-Relevant Subspace Concept. *J. Chem. Inf. Comput. Sci.* 1, 1999, 39, S. 28-35.

90. **Pearlman, Robert S. und Smith, K. M.** Novel software tools for chemical diversity. *Perspectives in Drug Discovery and Design.* 1998, 9, S. 339-353.

91. **Todeschini, R. und Grammatica, P.** 3D Modelling and Prediction by WHIM Descriptors. Part 5. Theory, Development and Chemical Meaning of WHIM Descriptors. *Quant. Struct. Act. Relat.* 1997, 16, S. 113-119.

92. **Todeschini, R., Marengo, E. und Lasagni, M.** New molecular descriptors for 2D and 3D structures. Theory. *Journal of Chemometrics.* 4, 1994, 8, S. 263-272.

93. **Ertl, P., Rohde, B. und Selzer, P.** Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* 20, 2000, 43, S. 3714-3717.

94. **Bremser, W.** HOSE: a novel substructure code. *Analytica Chimica Acta.* 1978, 103, S. 355-65.

95. —. Expectation ranges of 13-C NMR chemical shifts. *Magnetic Resonance in Chemistry.* 4, 1985, 23, S. 271-275.

96. **PostgreSQL.** [Online] [Zitat vom: 21. 10 2008.] http://www.postgresql.org/.

97. **Schmid, Ernst-Georg.** Pgchem::tigress. [Online] [Zitat vom: 22. 10 2008.] http://pgfoundry.org/projects/pgchem/.

98. **Haider, N.** Checkmol/Matchmol. [Online] [Zitat vom: 22. 10 2008.] http://merian.pch.univie.ac.at/~nhaider/cheminf/cmmm.html.

99. **The Open Babel Package.** Open Babel. [Online] [Zitat vom: 22. 10 2008.] http://openbabel.sourceforge.net/.

100. **Witten, Ian H.** *Data mining.* 2. ed., 4. print. Amsterdam u.a. : Elsevier, 2007.

101. **Spjuth, Ola, et al.** Bioclipse: An open source workbench for chemo- and bioinformatics. *BMC Bioinformatics.* 1, 2007, 8.

102. **Moore, G. E.** Cramming More Components onto Integrated Circuits. *Electronics.* 8, 1965, 38, S. 114-117.

103. **Intel.** Moore's Law. [Online] [Zitat vom: 23. 10 2008.] http://www.intel.com/museum/archives/history_docs/mooreslaw.htm.

104. **Swearingen, Kirsten.** HOW MUCH INFORMATION? [Online] http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/.

105. **GenBank.** Growth of GenBank. [Online] [Zitat vom: 23. 10 2008.] http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html.

106. **Popper, Karl R und Keuth, Herbert.** *Logik der Forschung.* Tübingen : Mohr Siebeck, 2005. 316148410X.

107. **Mitchell, Tom Michael.** *Machine learning.* New York : McGraw-Hill, 1997.

108. **Manning, Christopher D. und Schütze, Hinrich.** *Foundations of statistical natural language processing.* MIT Press : Cambridge, Mass., 2005.

109. **Sears, Andrew.** *The human-computer interaction handbook, Fundamentals evolving technologies and emerging applications.* 2. ed.. New York [u.a.] : Lawrence Erlbaum, 2008.

110. **Gasteiger, Johann und Engel, Thomas.** *Chemoinformatics, A textbook.* Weinheim : Wiley-VCH, 2003.

111. **Kotsiantis, S. B.** Supervised Machine Learning A Review of Classification Techniques. *Informatica.* 2007, Vol. 31, No. 3, pp. 249-268.

112. **Mcculloch, W. and Pitts, W.** A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics.* 1943, Vol. 5, pp. 115-133.

113. **Engelbrecht, Andries P.** *Computational intelligence, An introduction.* Chichester England , Hoboken N.J. : J. Wiley & Sons, 2002.

114. **Hopfield, J. J.** Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America.* 1982, Bd. 79, S. 2554-2558.

115. **Elman, Jeffrey L.** Finding structure in time. *Cognitive Science.* 1990, Vol. 14, 2, pp. 179-211.

116. **Kohonen, Teuvo.** *Self-organizing maps.* 3. ed. Berlin : Springer, 2001.

117. **Moody, J. E. and Darken, C.** Fast learning in networks of locally-tuned processing units. *Neural Computation.* 1989, 1, pp. 281-294.

118. **Lindenmair, Wolfgang und Kranzinger, Franz.** *Neuronale Netze.* Stuttgart : Klett-Schulbuchverl., 1995.

119. **Zupan, Jure und Gasteiger, Johann.** *Neural networks in chemistry and drug design.* 2nd. ed. Weinheim : Wiley-VCH, 1999.

120. **Carpenter, Gail A. und Grossberg, Stephen.** Adaptive resonance theory (ART). *The handbook of brain theory and neural networks.* 1998, S. 79-82.

121. **Grossberg, Stephen.** Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science.* 1, 1997, 11, S. 23-63.

122. **Carpenter, G. A. und Grossberg, S.** ART 2: self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.* 1987, 26, S. 4919-4930.

123. **Carpenter, G.A., Grossberg, S. und Rosen, D.** ART 2-A: an adaptive resonance algorithm for rapid categorylearning and recognition. *Neural networks.* 4, 1991, 4, S. 493-504.

124. **Carpenter, G.A. und Grossberg, S.** ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks.* 1990, 3, S. 129-152.

125. **Carpenter, Gail A., Grossberg, Stephen und Rosen, David B.** Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Netw.* 6, 1991, 4, S. 759-771.

126. **Carpenter, Gail A., Grossberg, Stephen und Reynolds, John H.** ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Netw.* 5, 1991, 4, S. 565-588.

127. **Carpenter, G.A., et al.** Fuzzy ARTMAP: an adaptive resonance architecture for incrementallearning of analog maps. *Neural Networks.* 1992, 3, S. 309-314.

128. Taverna 1.7.1 Manual . [Online] [Zitat vom: 30. 10 2008.] http://www.mygrid.org.uk/usermanual1.7/spi.html.

129. **Project, Apache Maven.** [Online] [Zitat vom: 31. 10 2008.] http://maven.apache.org.

130. **Kuhn, Thomas.** myExperiment. [Online] [Zitat vom: 03. 11 2008.] http://www.myexperiment.org/workflows/551.

131. **Gesellschaft für natruwissenschaftliche Informatik mbH.** GNWI. [Online] [Zitat vom: 25. 11 2008.] www.gnwi.de.

132. **Wolfram Research.** Woflram Research. [Online] [Zitat vom: 25. 11 2008.] www.wolfram.com.

133. **Sun.** Sun. [Online] [Zitat vom: 25. 11 2008.] www.sun.com.

134. **Microsoft.** Microsoft. [Online] [Zitat vom: 25. 11 2008.] www.microsoft.com.

135. *Optical structure recognition application.* **Filippov, Igor V. und Nicklaus, Marc C.** Philadelphia, PA, United States : s.n., 2008. Abstracts of Papers, 236th ACS National Meeting.

136. **Kuhn, Thomas.** myExperiment. [Online] [Zitat vom: 05. 11 2008.] http://www.myexperiment.org/workflows/552.

137. —. myExperiment. [Online] [Zitat vom: 05. 11 2008.] http://www.myexperiment.org/workflows/385.

138. —. myExperiment. [Online] [Zitat vom: 05. 11 2008.] http://www.myexperiment.org/workflows/553.

139. —. myExperiment. [Online] [Zitat vom: 06. 11 2008.] http://www.myexperiment.org/workflows/556.

140. —. myExperiment. [Online] [Zitat vom: 06. 11 2008.] http://www.myexperiment.org/workflows/554.

141. —. myExperiment. [Online] [Zitat vom: 07. 11 2008.] http://www.myexperiment.org/workflows/558.

142. —. myExperiment. [Online] [Zitat vom: 21. 11 2008.] http://www.myexperiment.org/workflows/567.

143. **Koch, K. U. und Zielesny, A.** Neuronale Netze verkürzen die Klebstoffentwicklung. *Adhäsion.* 2004, S. 32-37.

144. **GmbH, InterMed Discovery.** InterMed Discovery. [Online] [Zitat vom: 29. 11 2008.] www.intermed-discovery.com.

145. **Kuhn, Thomas.** myExperiment. [Online] [Zitat vom: 06. 11 2008.] http://www.myexperiment.org/workflows/555.

146. —. myExperiment. [Online] [Zitat vom: 07. 11 2008.] http://www.myexperiment.org/workflows/557.

147. —. myExperiment. [Online] [Zitat vom: 15. 11 2008.] http://www.myexperiment.org/workflows/563.

148. —. myexperiment. [Online] [Zitat vom: 16. 11 2008.] http://www.myexperiment.org/workflows/559.

149. —. myExperiment. [Online] [Zitat vom: 17. 11 2008.] http://www.myexperiment.org/workflows/564.

150. —. myExperiment. [Online] [Zitat vom: 17. 11 2008.] http://www.myexperiment.org/workflows/565.

151. **Anderson, Chris.** The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. *Wired.* 2008.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit − einschließlich Tabellen, Karten und Abbildungen −, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie − abgesehen von unten angegebenen Teilpublikationen − noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. D. Schomburg betreut worden.

Köln, den 17.02.2009

Thomas Kuhn