DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

# "Structure-based pharmacophore models for prediction of developmental toxicity"

verfasst von / submitted by

Vanja Mitrovic

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magistra der Pharmazie (Mag.pharm.)

Wien, 2019 / Vienna, 2019

| | |
|---|---|
| Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet: | A 449 |
| Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet: | Diplomstudium Pharmazie |
| Betreut von / Supervisor: | Univ.Prof. Mag. Dr. Gerhard Ecker |
| Mitbetreut von / Co-Supervisor: | Dr. Riccardo Martini |

# ACKNOWLEDGEMENT

First, I would like to express my sincere gratitude to my Supervisor Univ. Prof. Mag. Dr. Gerhard Ecker for giving me a chance to write my diploma thesis, his professional guidance and for letting me be a part of his research group and to learn from them.

Additionally, I am very grateful to my Cosupervisor Dr. Riccardo Martini for his immense support and guidance during my research and afterwards. I would also like to thank the Dr. Ecker`s research group for their contribution and motivation.

# ABSTRACT

The objective of this thesis was to evaluate whether the final output of structure-based pharmacophore virtual screening can be a valid input for machine learning methods in order to obtain predictions for a compound to induce DART. This thesis started with a list of proteins, that are supposed to participate in the molecular initiating event of a DART adverse outcome pathway, taken from the Crackit DART challenge. We downloaded all the PDBs associated with the UniprotIDs to generate structure-based pharmacophore models. The pharmacophore models were subsequently used for screening. The virtual screening output was used to create a model whose prediction was compared with the prediction of a model made using a set of fingerprints calculated based on the dataset.

# ZUSAMMENFASSUNG

Das Ziel dieser Diplomarbeit war es zu bestimmen, ob die Ergebnisse einer Pharmakophor–Datenbankdurchsuchung zum Erstellen von Computermodelle durch maschinelles Lernen verwendet werden können, um zu bestimmen welche chemischen Verbindungen zu DART führen können. Diese Diplomarbeit basiert auf einer Liste von Proteinen, die vermutlich am DART Signalweg beteiligt sind. Die Liste wurde von Crackit DART challenge erstellt. Zuerst wurden alle PDBs mit dazugehörigen UniprotIDs heruntergeladen, um die strukturbasierte Pharmakophormodelle zu erstellen, die dann für die Datenbankduchsuchung verwendet wurden. Abschließend wir haben die Modelle erstellt, deren Voraussagen verglichen wurden.

# TABLE OF CONTENTS

# 1 GENERAL BACKGROUND

## *1.1* Developmental toxicity

Developmental toxicity is a field that is receiving an increasing attention nowadays. It associates any adverse toxic effect to the embryo development or fetus. One of the causes of developmental toxicity are chemicals, that could affect embryo development in different ways. On the one side the chemicals could act directly on the cells of the embryo or fetus and cause cell damages or cell death, leading to the abnormal development. On the other side, the chemicals can cause a mutation in the parent's germ cells, which can be then transmitted to the fertilized ovum. The mutation in fertilized ovum can lead to abnormal embryo development. In the developmental toxicity studies there are two important dose descriptors to which the toxicity level is referred. The lowest observed adverse effect level (LOAEL) is referred as a measurement of how toxic the chemicals can be. It is the dose minimum at which observed adverse effect on the organism are recognized [1]. Therefore, Organization for economic Co-operation and development (OECD) have developed the testing guidelines for research on the test animals in the field of developmental and reproductive toxicity to determine the toxic potential of different chemicals [2].

The 3 basic types of developmental toxicity are:

- Embryolethality: Failure to conceive, spontaneous abortion
- Embryotoxicity: Growth retardation or delayed growth of specific organ
- Teratogenicity: irreversible conditions that leave permanent birth defects in live offspring

## *1.2* Embryonal development

The first stage of embryonal development starts with the formation of zygote as a product of the fusion of male and female pronuclei. As the Zygote is transported to the uterus, its cells undergo mitotic division, forming so called blastomeres. Blastomeres continuously divide themselves and produce the inner and outer cells, forming a cavity which is known as a blastocyst. Due to the further cell division the inner cells will form the embryo and the outer cell layer will form the trophoblast, that is essential for intrauterine mammalian development.

The second stage known as implantation is achieved when the blastocyst reaches the uterine wall and implants inside. However, if the endometrium is not fully developed, the blastocyst cannot embed itself into endometrium and the implantation fails. If the implantation succeeds the throphoblast cells fuse with each other, forming the syncytiotrophoblast, that secures blastocyst to the endometrium. In this stage the trophoblast starts with production of human chorionic gonadotropin (hCG), that is important for the development of the embryo [3].

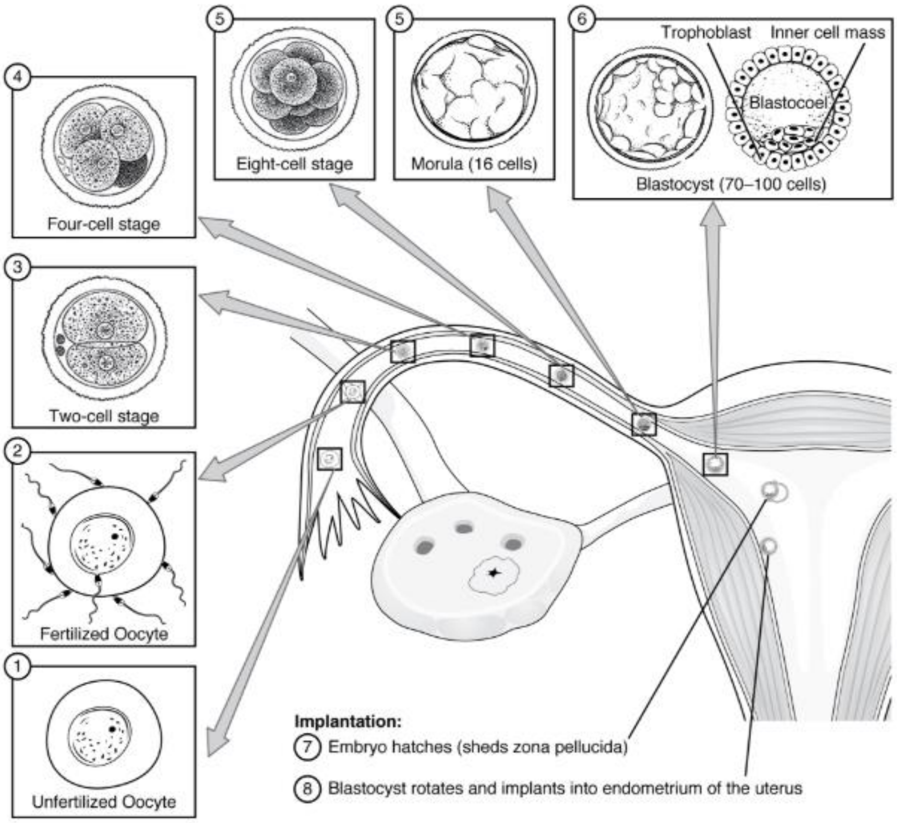


*Figure 1. The stages of embryonal development taken from https://opentextbc.ca/anatomyandphysiology/chapter/28-2-embryonic-development/*

Additionally, different signaling pathways are responsible for developmental stages of embryo and determine to high extent the later development. Therefore any mutation or disruption in those pathways can lead to severe malformation or disturbed development [4].

The crucial morphogenetic process occurs during the blastogenesis, which extends throughout the first 4 weeks of embryonal development. In this stage the embryo is more susceptible to different factors that could affect it and lead to a variety of different developmental abnormalities such as growth restriction, miscarriage or later on fetal death [5].

## *1.3* Animal studies

Most of the studies done in this field are animal-based studies. However, in vivo studies are mostly time consuming, expensive [6] and can result in the lack of scientific certainty [7] which  includes the challenges in extrapolation of findings from animals to humans [8]. Thus, alternative methods to predict the developmental toxicity which include model organism such as Drosophila [9], as well as zebrafish embryo [10] or mammalian embryo culture [11] have been proposed. The data generated in these model organisms and human cells facilitates the translation to the effects in humans [12]. Therefore, the Crackit Challenge 26 - based on the principles of the 3R`s - is trying to use cell based methods and computational approaches to predict DART effects. The assumption behind it is, that the molecular initiating event caused by molecular reaction at a molecular level could induce an adverse effect in the organism (AOP).

## *1.4* NC3R

NC3R`s is the scientific organization in UK which is trying to provide a framework for more appropriate animal research.  The principles of the organization are based on 3R`s (Replacement, Reduction, Refinement) in animal research.

The Replacement is considered as a method to avoid or replace the use of animals in research. The aim is to accelerate the development of new research approaches to obtain reliable results. There are two categories: full and partial Replacement.

- The full Replacement is considered as a restraint to use the animals in the research.
- The partial Replacement method is based on the use of animals, which are considered as uncapable of suffering, according to current scientific thinking.

The second principle, Reduction, refers to the methods that reduce the number of animals used in the experiments. However, it also includes the maximization of data through usage of other methods to obtain replicable results.

The third principle, Refinement, refers to the methods, that avoid pain, distress and harm of the animals, in all stages of animal use.

## *1.5* Crackit Challenge

CRACKIT is the project, that is trying to deliver the new technologies with 3R benefit. It has been developed in order to facilitate the collaboration between pharmaceutical, chemical and academical branches in order to accelerate the development and availability of different 3R approaches.

A Crackit project is divided into two parts: Crackit Challenges and Crackit Solutions, to maximize scientific and commercial benefits of the new technologies.

A Crackit Challenge is based on funding different collaborations between pharmaceutical or chemical industries and academics to solve scientific and business challenges involving animals in research.

A Crackit Solution is a technology hub, developed to accelerate the development of different methods with 3R`s impact and its scientific application  in order to get better commercial benefits [13].

One of the Crackit Challenges, on which my thesis is based on, is the DART Challenge (Developmental and reproductive toxicity).

### *1.5.1 DART Challenge*

Developmental and reproductive toxicity testing are focusing on estimating the effect of chemicals on adult fertility and sexual behavior, implantation and the development of the embryo. However, DART toxicity studies are animal-based studies, which are often time consuming and the application to the humans is not clear.  Therefore, researchers are trying to use cell-based or computational methods in order to discover different chemicals with DART effect. The DART pathway is based on the concept of adverse outcome pathway (AOP), that links the molecular initiating event caused by chemical interaction on the molecular level with the adverse effect on the organism. Considering the fact that many mechanisms could be involved in a DART AOP, scientists are trying to use the data obtained in human cells and non-mammalian model organism in a DART AOP in order to improve the translation of the data obtained in these systems to effects in humans.

The DART Challenge is sponsored by Shell and Syngenta with the aim

- to develop the data strategy on how to properly relate the data between a compound and effect, or between specific gene and specific physiology for model organisms, such as human, mouse, rat, etc
- to properly match the relationship between genes and physiology in order to apply it to humans

Considering this, the Crackit project a provided a list of proteins that are supposed to be involved in the DART pathway [12].

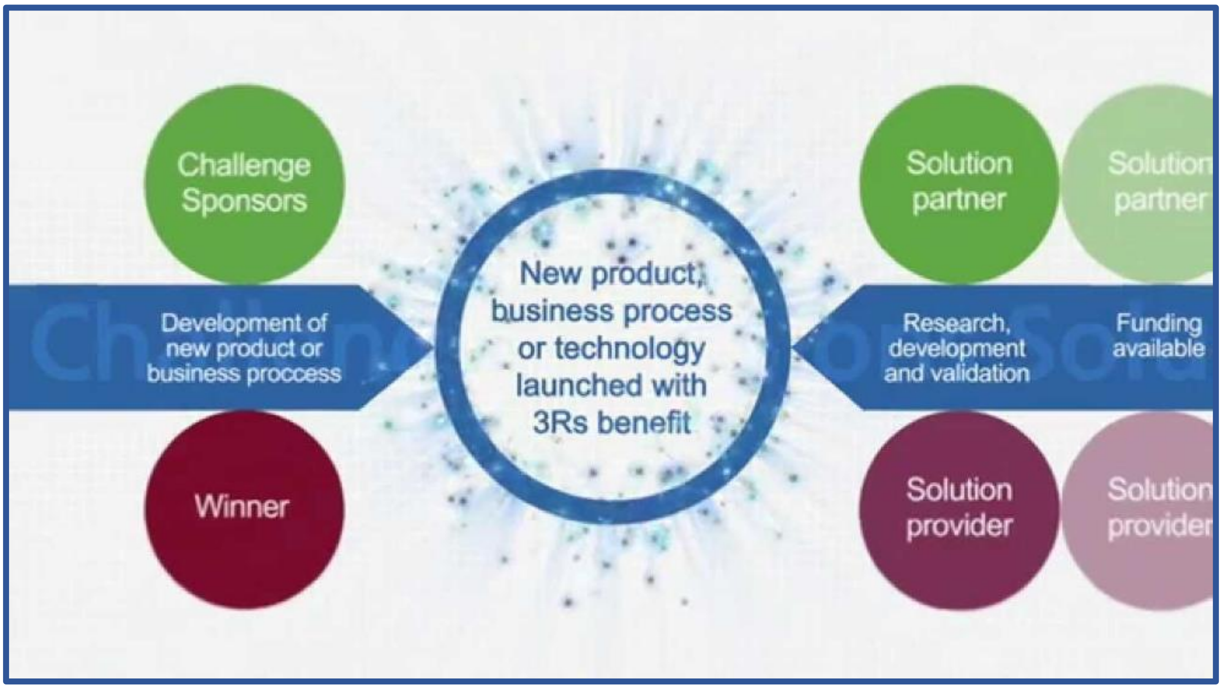

Challenge Sponsors
Development of new product or business proccess
Winner
New product, business process or technology launched with 3Rs benefit
Solution partner
Solution partner
Research, development and validation
Funding available
Solution provider
Solution provider

*Figure 2. Short overview of Crackit Project taken from https://crackit.org.uk*

# 2 AIM OF THE RESEARCH

The aim of this thesis was to evaluate whether the final output of structure-based pharmacophore virtual screening can be a valid input for machine learning methods in order to obtain predictions for a compound to induce DART.

This thesis started with a list of proteins, that are supposed to participate in the molecular initiating event of a DART adverse outcome pathway, taken from the Crackit DART challenge. The first step was to download all the PDBs associated with the UniprotIDs, in order to generate structure-based pharmacophore models. The pharmacophore models were subsequently used for screening of the Toxref database. The virtual screening output was used to create a model whose prediction was compared with the prediction of a model made using a set of fingerprints calculated based on the dataset.

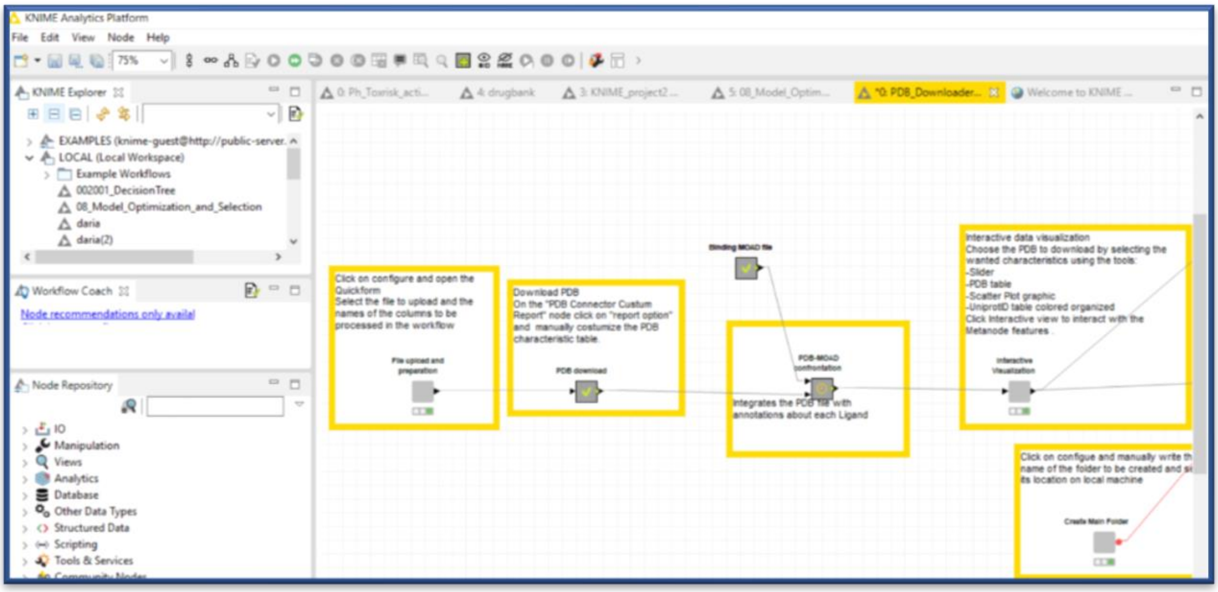
# 3 COMPUTATIONAL METHODS

## 3.1 KNIME - Konstanz Information Miner

Knime is an open source software, which provides a platform for enabling data visualization and interactive execution of a data pipeline. Throughout Knime, it is possible to process data from different research areas as part of one workflow. For data science, a Knime workflow has a crucial significance, because it enables the documentation of a large amount of data, and it makes reproducible science easier.

A Knime workflow contains an extensive catalogue of different nodes from different research areas to create cross-domain workflows. A node is the smallest processing unit in Knime that is executed to perform a specific task. To create a workflow the nodes are connected between each other, as each node has an input and an output port in order to transfer the data [14].

One of the important Knime strengths are the data analysis and machine learning functions, that were used in this thesis [14].



Click on configure and open the Quickform
Select the file to upload and the names of the columns to be processed in the workflow

Download PDB
On the "PDB Connector Custum Report" node click on "report option" and manually costumize the PDB characteristic table.

Integrates the PDB file with annotations about each Ligand

Interactive data visualization
Choose the PDB to download by selecting the wanted characteristics using the tools:
-Slider
-PDB table
-Scatter Plot graphic
-UniprotID table colored organized
Click Interactive view to interact with the Metanode features .

Click on configue and manually write th
name of the folder to be created and s
its location on local machine

*Figure 3. A part of the Knime workflow for the developmental toxicity*

## *3.2* Pharmacophore modelling

The pharmacophore concept was first introduced in 1909 by Ehrlich [15], who described a pharmacophore as a molecular framework that carries special drug features, responsible for drug`s biological activity. The recent definitions define pharmacophore as a three dimensional arrangement of chemical features necessary for the ligand molecule to interact with proteins in a specific binding mode [16]. The 3D arrangement of chemical features represents the chemical functionalities of active small molecules such as hydrogen bond acceptor (HBA), hydrogen bond donor (HBD), hydrophobic areas, positively or negatively charged areas and metal coordinating areas [17]. The three dimensional chemical feature allows easy interpretability and efficient implementation of high-throughput virtual screening methods [18].

There are two types of pharmacophore models:

- Ligand-based pharmacophore model, which are based on a set of known ligands and their activity at a given protein. It uses chemical features from the 3D structure of a set of known ligands that are representative for protein-ligand interactions. The first step of this approach is a creation of a conformational space for the ligands to represent the conformational flexibility of the ligands. The second step is to align the multiple ligands and to determine the essential chemical features to construct the pharmacophore model [15].
- Structure-based pharmacophore model uses the 3D structure of the protein-ligand complex. It includes the analysis of the chemical features in the binding site and their spatial bonds. This approach is only applied when the 3D structure of the protein-ligand complex is known. However, a frequently encountered problem is the presence of too many chemical features, as pharmacophore models with more than 7 chemical features are too selective for the virtual screening of different databases [15].

## *3.3* LigandScout

LigandScout is the software used in this thesis to perform the structure-based pharmacophore modelling of the crystal structures derived from Protein Data Bank. The software uses different algorithms to perform alignments and to represent ligand-protein interactions [19].

The pharmacophore interactions are represented through different interactions, such as hydrogen bond acceptors and donors, charge and lipophilic interactions [20]. LigandScout uses all available chemical features in order to analyze the ligand-macromolecule interaction [17], whereas different features are characterized differently (Figure 4):

- Yellow spheres represent lipophilic areas
- Red arrows represent hydrogen bond acceptor
- Green arrows represent hydrogen bond donor
- Blue circles represent aromatic structures
- Grey spheres represent the exclusion volume coat
- Red spheres represent negative ionizable areas
- Blue spheres represent positive ionizable areas
- Blue cones represent metal binding feature

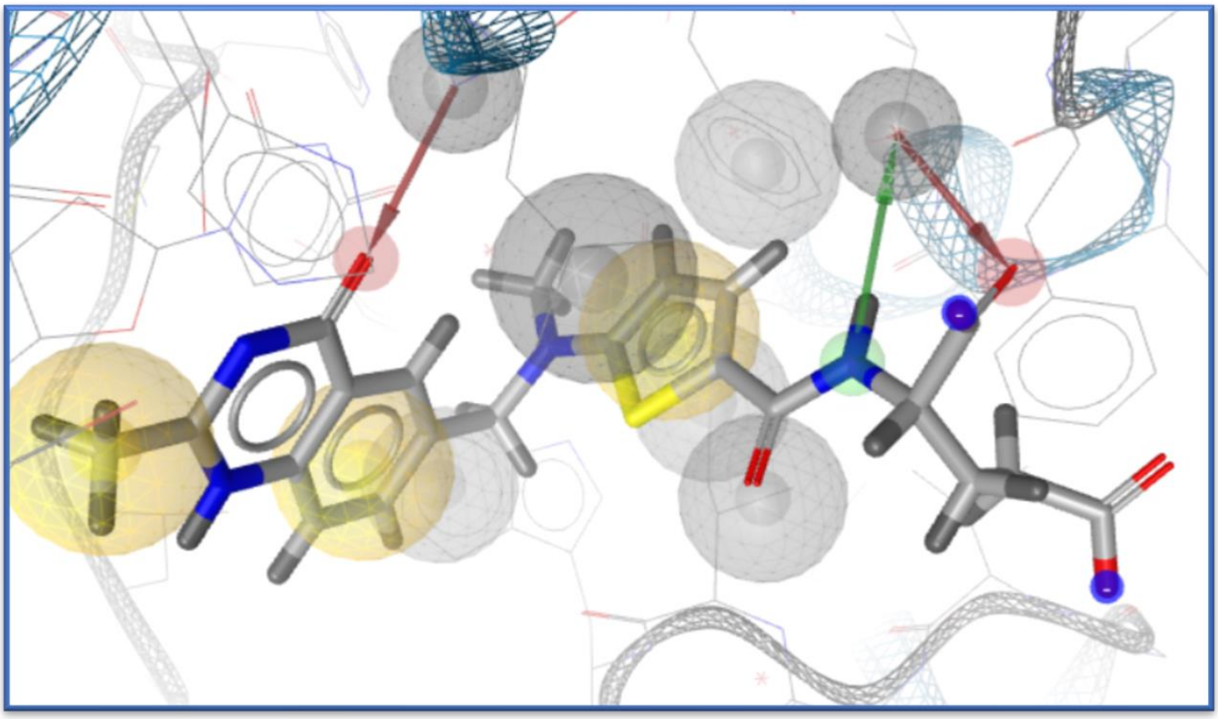


*Figure 4. The pharmacophore of the PDB (1IOO) in LigandScout*

The hydrogen bond distance is set to be in the range between 2.5 and 3.8 Å by default, and the angles between donor and acceptor are stated to be ideal of 180∘.

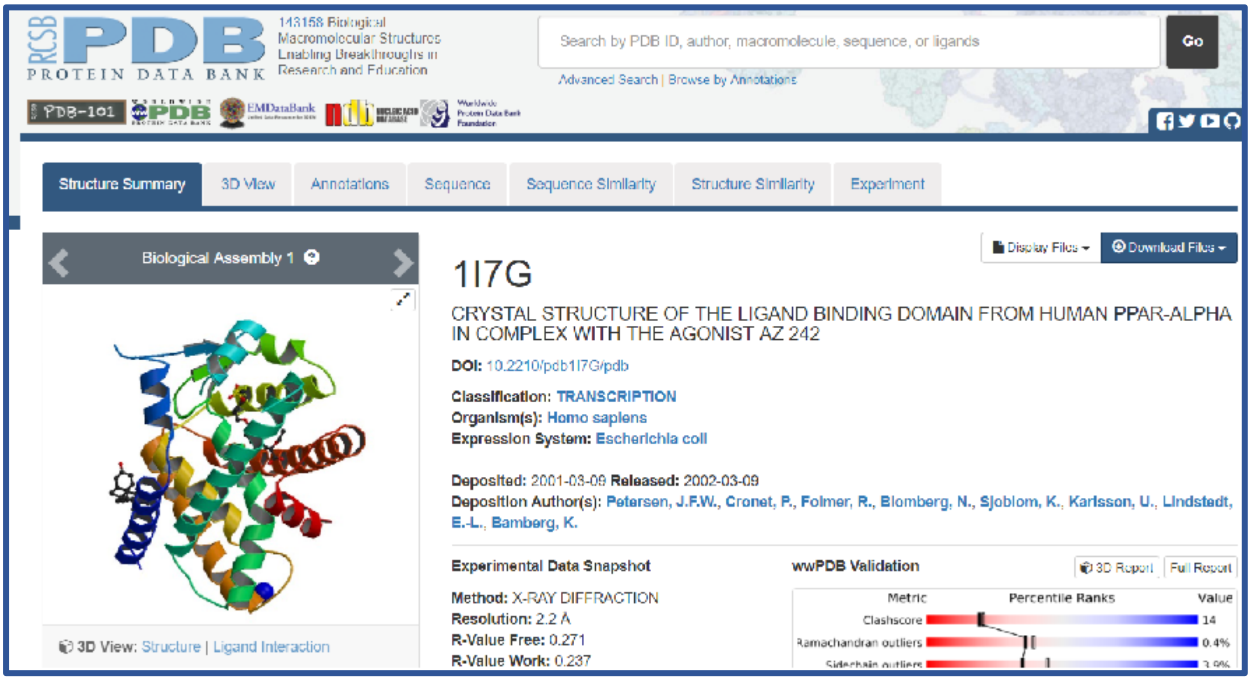
Hydrophobic areas are represented as spheres. However, the hydrophobic spheres are placed only if there are hydrophobic areas within a distance range of 1-5 Å at the macromolecule side.

Additionally, if the program recognizes for all atoms or groups which are protonated or deprotonated at physiological pH an interaction partner in the distance range within a 1.5-5.6 Å, it adds the feature or sphere to the charged group [17].

As an additional chemical feature, the program includes the excluded volume spheres regarding areas which are inaccessible for the ligands in order to reflect possible steric restrictions [21].

## *3.4* The Protein Data Bank

The Protein Data Bank is a worldwide archive for experimentally determined, atomic-level three dimensional structures of biological macromolecules [22], which was established in 1971 [23]. It contains approximately 130,000 protein structures (May 2017) from multiple species [24]. The PDB includes a wide range of macromolecules including enzymes, membrane proteins, protein bound to DNA and some viruses [25]. Most of the atomic structures of proteins in the PDB were determined by X-Ray crystallography, some of them with NMR spectroscopy and cryo-electron microscopy. Each PDB entry is characterized by a 4-character PDB identifier [26] as it is presented in Figure 5. All PDB entries include atomic structure information, experimental procedures, data about small molecules, and structure determination data.


RCSB PDB
PROTEIN DATA BANK
143158 Biological Macromolecular Structures Enabling Breakthroughs in Research and Education
Search by PDB ID, author, macromolecule, sequence, or ligands
Go
Advanced Search | Browse by Annotations
PDB-101
wwPDB
EMDataBank
NDB
Worldwide Protein Data Bank Foundation
Structure Summary
3D View
Annotations
Sequence
Sequence Similarity
Structure Similarity
Experiment
Biological Assembly 1
Display Files
Download Files
3D View: Structure | Ligand Interaction
3D Report
Full Report

1I7G

CRYSTAL STRUCTURE OF THE LIGAND BINDING DOMAIN FROM HUMAN PPAR-ALPHA IN COMPLEX WITH THE AGONIST AZ 242

**DOI:** 10.2210/pdb1I7G/pdb

**Classification:** TRANSCRIPTION
**Organism(s):** Homo sapiens
**Expression System:** Escherichia coli

**Deposited:** 2001-03-09 **Released:** 2002-03-09
**Deposition Author(s):** Petersen, J.F.W., Cronet, P., Folmer, R., Blomberg, N., Sjoblom, K., Karlsson, U., Lindstedt, E.-L., Bamberg, K.

**Experimental Data Snapshot**

**Method:** X-RAY DIFFRACTION
**Resolution:** 2.2 Å
**R-Value Free:** 0.271
**R-Value Work:** 0.237

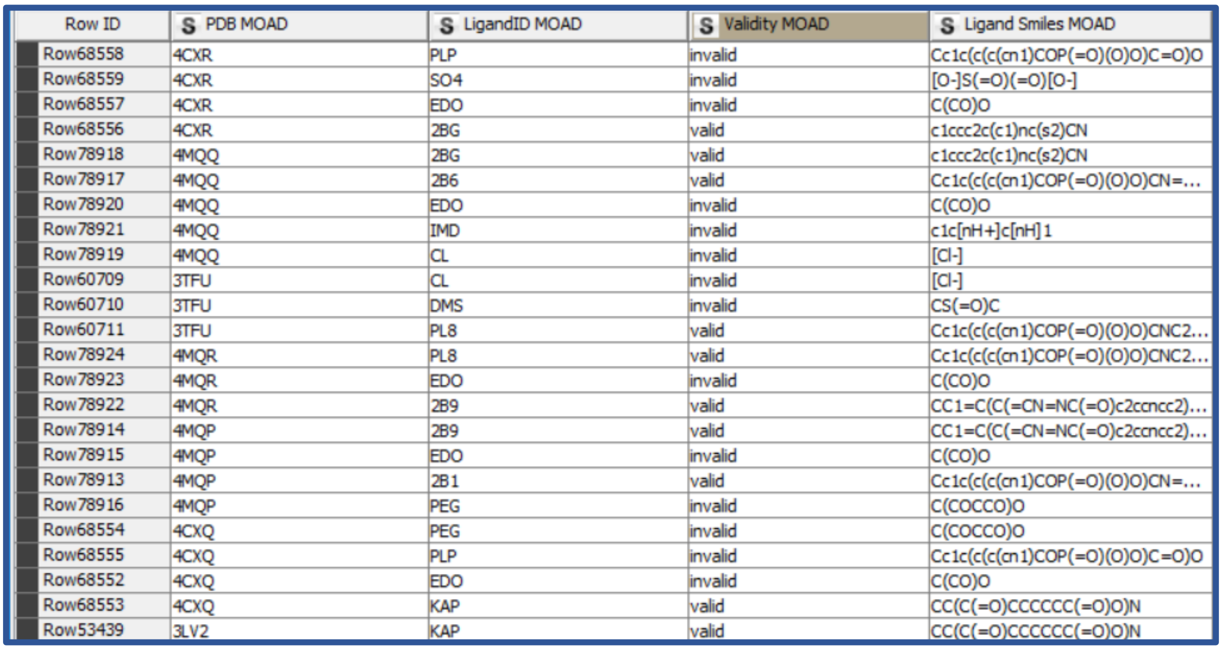
**wwPDB Validation**

| Metric | Percentile Ranks | Value |
|---|---|---|
| Clashscore | | 14 |
| Ramachandran outliers | | 0.4% |
| Sidechain outliers | | 3.9% |

*Figure 5. One of the Protein Data Bank entries*

## *3.5* BINDING MOAD – Mother of all Database

Binding MOAD is a large collection of high resolution structures from PDB with ligand annotation (valid/invalid) and protein classification (enzyme/non enzyme) [27], updated till 2014 [28]. It includes all entries of the Protein data bank, but it excludes structures which are inappropriate. It retains only those structures whose resolution is better than 2.5 A. The Binding MOAD distinguishes between small molecules which are considered as a part of the crystallization matrix or an artifact of the protein and therefore are stated as invalid ligands. On the other hand it considers small biological molecules like agonists, antagonists, inhibitors, cofactors as valid ligands [29]. The focus lies on small molecules bound to the protein, so peptides containing more than 10 amino acids or chains containing more than 4 nucleic acids are not considered as being relevant. This makes MOAD much more appropriate in categorizing ligands, when compared with the Protein Data Bank, which does not distinguish between valid and invalid ligands.



| Row ID | S PDB MOAD | S LigandID MOAD | S Validity MOAD | S Ligand Smiles MOAD |
|---|---|---|---|---|
| Row68558 | 4CXR | PLP | invalid | Cc1c(c(c(cn1)COP(=O)(O)O)C=O)O |
| Row68559 | 4CXR | SO4 | invalid | [O-]S(=O)(=O)[O-] |
| Row68557 | 4CXR | EDO | invalid | C(CO)O |
| Row68556 | 4CXR | 2BG | valid | c1ccc2c(c1)nc(s2)CN |
| Row78918 | 4MQQ | 2BG | valid | c1ccc2c(c1)nc(s2)CN |
| Row78917 | 4MQQ | 2B6 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CN=... |
| Row78920 | 4MQQ | EDO | invalid | C(CO)O |
| Row78921 | 4MQQ | IMD | invalid | c1c[nH+]c[nH]1 |
| Row78919 | 4MQQ | CL | invalid | [Cl-] |
| Row60709 | 3TFU | CL | invalid | [Cl-] |
| Row60710 | 3TFU | DMS | invalid | CS(=O)C |
| Row60711 | 3TFU | PL8 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CNC2... |
| Row78924 | 4MQR | PL8 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CNC2... |
| Row78923 | 4MQR | EDO | invalid | C(CO)O |
| Row78922 | 4MQR | 2B9 | valid | CC1=C(C(=CN=NC(=O)c2ccncc2)... |
| Row78914 | 4MQP | 2B9 | valid | CC1=C(C(=CN=NC(=O)c2ccncc2)... |
| Row78915 | 4MQP | EDO | invalid | C(CO)O |
| Row78913 | 4MQP | 2B1 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CN=... |
| Row78916 | 4MQP | PEG | invalid | C(COCCO)O |
| Row68554 | 4CXQ | PEG | invalid | C(COCCO)O |
| Row68555 | 4CXQ | PLP | invalid | Cc1c(c(c(cn1)COP(=O)(O)O)C=O)O |
| Row68552 | 4CXQ | EDO | invalid | C(CO)O |
| Row68553 | 4CXQ | KAP | valid | CC(C(=O)CCCCCC(=O)O)N |
| Row53439 | 3LV2 | KAP | valid | CC(C(=O)CCCCCC(=O)O)N |

*Figure 6. Binding MOAD (small section)*

## *3.6* Toxicity Reference Database (ToxRef)

The Toxicity Reference Database was created to present the data from the guideline in vivo toxicity studies. It contains the review of the submitted toxicity studies, known as data evaluation records for roughly 400 chemicals from the U.S. EPA's Office of Pesticide Programs (OPP). The data included five types of studies from a variety of species: developmental in rat and rabbit, subchronic in rat and mouse, reproductive in rat, chronic or cancer in rat or mouse. The doses are given in part per million or in mg/kg, based on the body weight and food consumption. Moreover, the observed effects were described on the all dose levels. The critical level at which a distinct effect was observed is described as LOAEL, whereas the level where there was no effect observed  is termed NOAEL [30].

# 4 COMPUTATIONAL APPROACH

In this thesis we used the Knime software to process the large amount of data obtained in this thesis and to develop a workflow which enables the interactive data visualization.

## *4.1* Knime workflow

This thesis was based on the list of proteins, that are supposed to participate in development of developmental toxicity (Appendix) [12]. Each protein sequence can be characterized by a specific UniprotID, therefore the corresponding UniprotIDs are usually attributed to each protein. Starting from a list containing 182 UniprotIDs we developed a Knime workflow to download all crystal structures available in the Protein Data Bank which correspond to the UniprotIDs present on the list.

## *4.2* PDB Download

First and foremost, we made a table that contains UniprotIDs and the protein name as columns. In order to download the PDBs, the first part of the Knime workflow was developed. The first node used was the *Loop start node* to use each UniprotID iteration to download the PDB. To download all the PDBs the query was defined and used as input for the *node PDB Connector Custom Report*. Through this node we retrieve the PDBs and it enables also the selection of specific properties, such as

- Structure summary: resolution, release date, classification
- Ligand details: Ligand smiles, InChI Key, LigandID, Ligand Name
- Binding affinity: Kd, EC50, IC50
- Biological details: source, plasmid name, taxonomy ID

To close a loop, the *Loop end* node was used to obtain the necessary PDBs as it was shown in Figure 7.

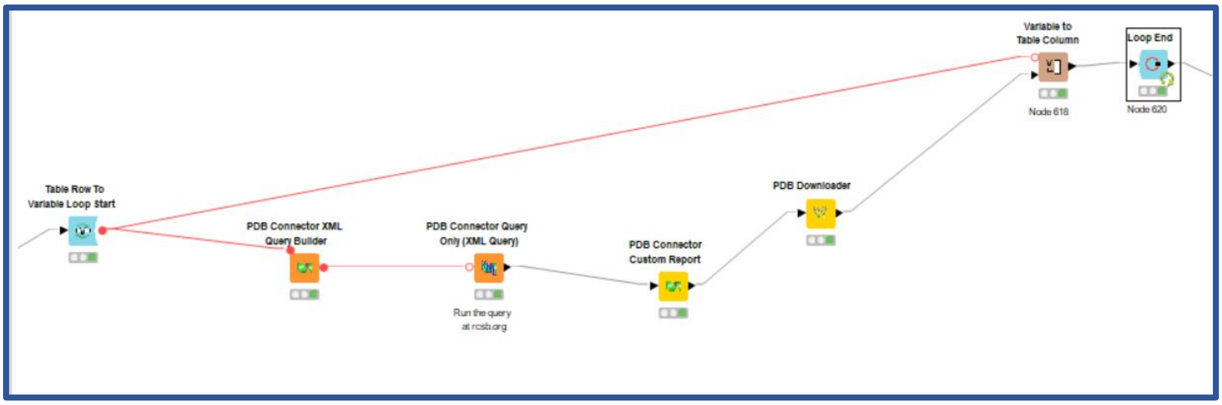
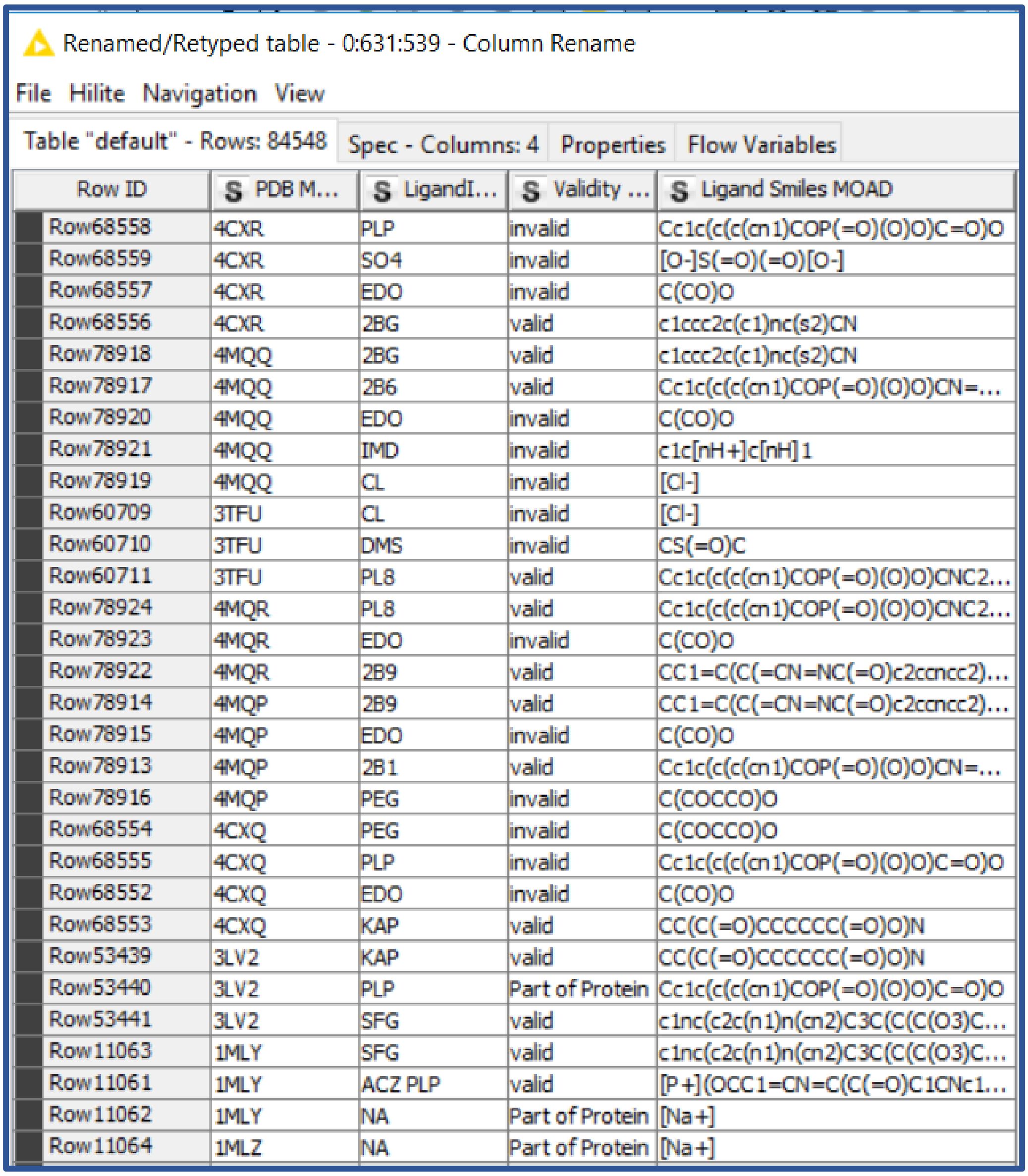


*Figure 7. Knime pathway to download PDB from Protein Data Bank based on the UniprotID*

## *4.3* Binding MOAD Integration

The biggest advantage of the MOAD is the clear separation between valid and invalid Ligands, such as artifacts or solvents. To clearly state which ligands from downloaded PDB are valid for this thesis, the information regarding the ligands that are present in the Binding MOAD were integrated in the second part of the Knime workflow. First, a csv file containing Binding MOAD data was downloaded from the website. Some manual editing was necessary to transform it into the desired format.

The resulting table containing the desired characteristics was then created (Figure 8).


Renamed/Retyped table - 0:631:539 - Column Rename
File Hilite Navigation View
Table "default" - Rows: 84548
Spec - Columns: 4
Properties
Flow Variables

| Row ID | S PDB M... | S LigandI... | S Validity ... | S Ligand Smiles MOAD |
|---|---|---|---|---|
| Row68558 | 4CXR | PLP | invalid | Cc1c(c(c(cn1)COP(=O)(O)O)C=O)O |
| Row68559 | 4CXR | SO4 | invalid | [O-]S(=O)(=O)[O-] |
| Row68557 | 4CXR | EDO | invalid | C(CO)O |
| Row68556 | 4CXR | 2BG | valid | c1ccc2c(c1)nc(s2)CN |
| Row78918 | 4MQQ | 2BG | valid | c1ccc2c(c1)nc(s2)CN |
| Row78917 | 4MQQ | 2B6 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CN=... |
| Row78920 | 4MQQ | EDO | invalid | C(CO)O |
| Row78921 | 4MQQ | IMD | invalid | c1c[nH+]c[nH]1 |
| Row78919 | 4MQQ | CL | invalid | [Cl-] |
| Row60709 | 3TFU | CL | invalid | [Cl-] |
| Row60710 | 3TFU | DMS | invalid | CS(=O)C |
| Row60711 | 3TFU | PL8 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CNC2... |
| Row78924 | 4MQR | PL8 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CNC2... |
| Row78923 | 4MQR | EDO | invalid | C(CO)O |
| Row78922 | 4MQR | 2B9 | valid | CC1=C(C(=CN=NC(=O)c2ccncc2)... |
| Row78914 | 4MQP | 2B9 | valid | CC1=C(C(=CN=NC(=O)c2ccncc2)... |
| Row78915 | 4MQP | EDO | invalid | C(CO)O |
| Row78913 | 4MQP | 2B1 | valid | Cc1c(c(c(cn1)COP(=O)(O)O)CN=... |
| Row78916 | 4MQP | PEG | invalid | C(COCCO)O |
| Row68554 | 4CXQ | PEG | invalid | C(COCCO)O |
| Row68555 | 4CXQ | PLP | invalid | Cc1c(c(c(cn1)COP(=O)(O)O)C=O)O |
| Row68552 | 4CXQ | EDO | invalid | C(CO)O |
| Row68553 | 4CXQ | KAP | valid | CC(C(=O)CCCCCC(=O)O)N |
| Row53439 | 3LV2 | KAP | valid | CC(C(=O)CCCCCC(=O)O)N |
| Row53440 | 3LV2 | PLP | Part of Protein | Cc1c(c(c(cn1)COP(=O)(O)O)C=O)O |
| Row53441 | 3LV2 | SFG | valid | c1nc(c2c(n1)n(cn2)C3C(C(C(O3)C... |
| Row11063 | 1MLY | SFG | valid | c1nc(c2c(n1)n(cn2)C3C(C(C(O3)C... |
| Row11061 | 1MLY | ACZ PLP | valid | [P+](OCC1=CN=C(C(=O)C1CNc1... |
| Row11062 | 1MLY | NA | Part of Protein | [Na+] |
| Row11064 | 1MLZ | NA | Part of Protein | [Na+] |

*Figure 8. shows small section from Binding MOAD*

Inside the metanode named PDB-MOAD Confrontation (Figure 9) the Protein Data Bank Ligands were confronted with MOAD Ligands. First, the *Group Loop Start* was executed to group all the PDB-IDs from the Protein Data Bank, having the same UniprotID as a criterion.

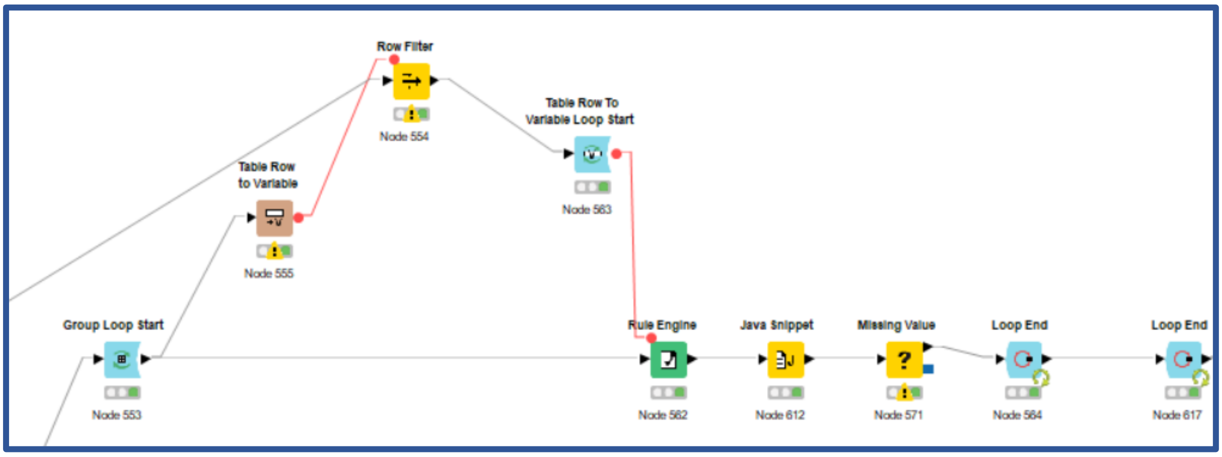
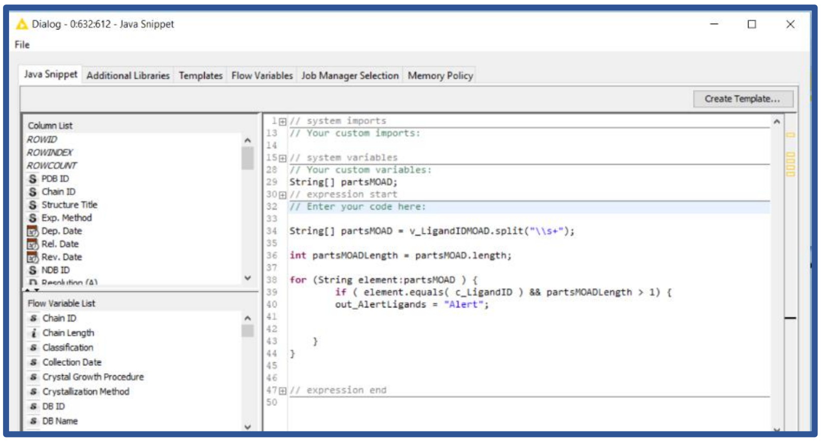


Figure 9. MOAD and PDB confrontation

Using the *Rule Engine node*, the validity of the Ligand was checked for the entries downloaded from Protein Data Bank, thanks to the comparison with MOAD. If the Ligand is present in the PDB database but not in MOAD, an additional column is added to the table with the statement “Ligand is not part of the MOAD”.

However, the MOAD database contains multimeric ligands which are not correctly read in the PDB database. Therefore, additional descriptions of the ligands were added to the table with *Java Snippet node*. The multimeric ligands could this way be compared between PDB and MOAD. An alert was added if the LigandID downloaded from the Protein Data Bank and Ligand ID from MOAD match, but do not correspond completely as it is presented in Figure 10.



```
// system imports
// Your custom imports:

// system variables
// Your custom variables:
String[] partsMOAD;
// expression start
// Enter your code here:

String[] partsMOAD = v_LigandIDMOAD.split("\\s+");

int partsMOADLength = partsMOAD.length;

for (String element:partsMOAD ) {
        if ( element.equals( c_LigandID ) && partsMOADLength > 1) {
        out_AlertLigands = "Alert";


    }
}


// expression end

```

Figure 10. Java Snippet node configure window

However, considering that the column LigandID contained missing values, the *Missing value node* was integrated into the workflow (Figure 9), in order to replace the missing values with the expression “Ligand not present”. To close the loop, *End Loop node* was executed with the obtained results.

## *4.4* Interactive Visualization

After the results were obtained by combining the PDB and MOAD Databases, the next step was to represent the collected data in tables and a sunburst chart. In order to achieve this, the *metanode Interactive Visualization* was used. First, as a best way to represent the data in a Sunburst chart, the *Color Manager node* was used to add a color for each UniprotID. Afterwards, the *node Date Field Extractor* was used to extract the release date and the resolution of the structure. These two data were then combined in an interactive scatter plot having the value “Resolution” on Y axis and the value “Release Date” on the X axis (Figure 11).

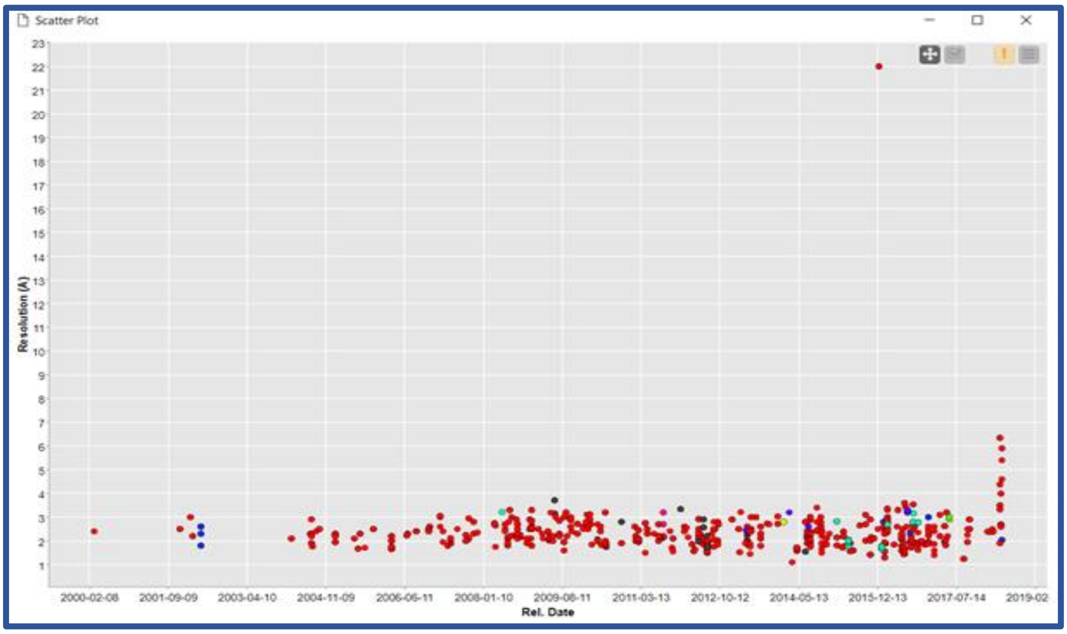
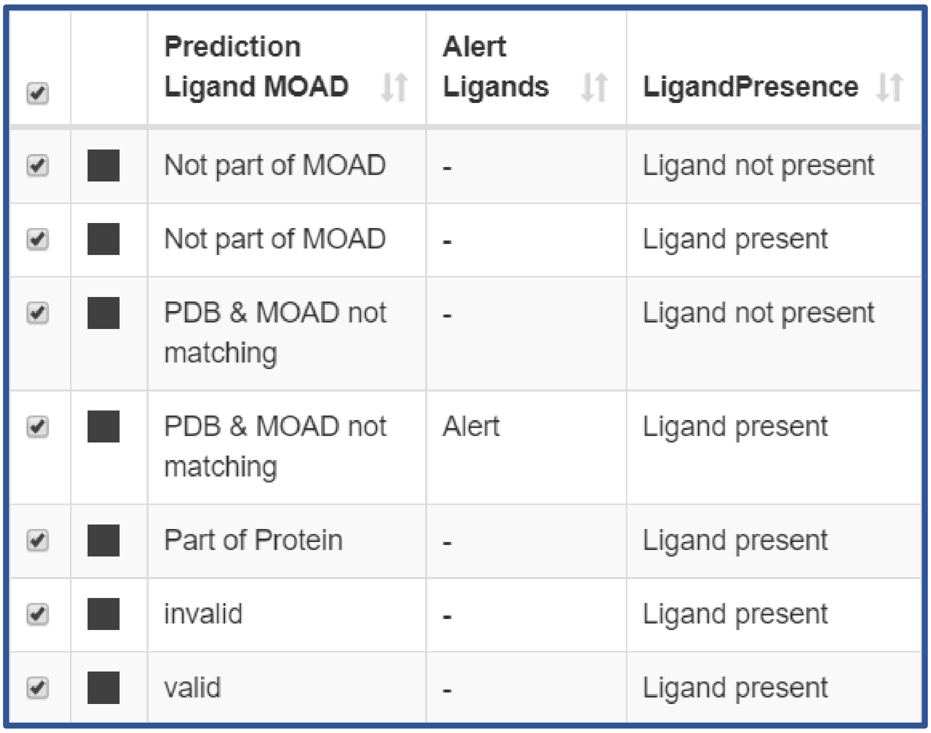


*Figure 11. Scatter plot includes Resolution value on the Y axis and Release Date on the X axis*

Additionally, to have a better overview of the collected data, the additional ligand information was added to the table. The *Rule Engine node* was used to state the ligand presence in the PDB. Furthermore, the *Missing Value Nodes* state whether the

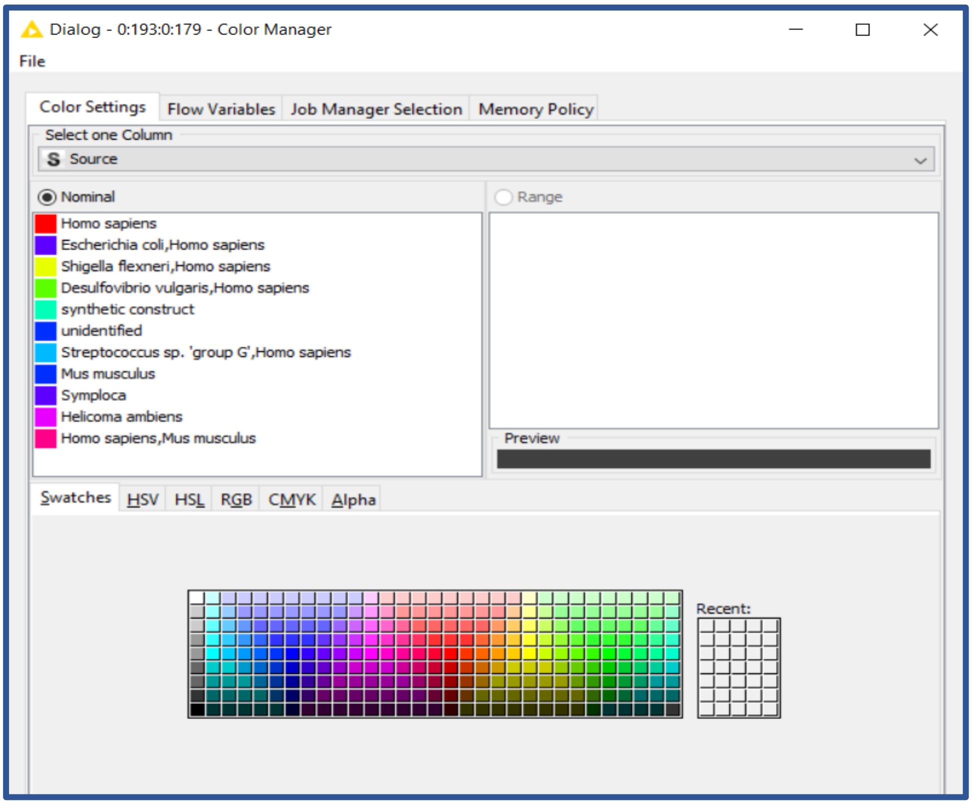
LigandID in the MOAD Database matches the LigandID in the PDB Database. The resulting table allows us to choose the characteristics of the ligands (Figure 12).



| ☑ | | Prediction Ligand MOAD | Alert Ligands | LigandPresence |
|---|---|---|---|---|
| ☑ | ■ | Not part of MOAD | - | Ligand not present |
| ☑ | ■ | Not part of MOAD | - | Ligand present |
| ☑ | ■ | PDB & MOAD not matching | - | Ligand not present |
| ☑ | ■ | PDB & MOAD not matching | Alert | Ligand present |
| ☑ | ■ | Part of Protein | - | Ligand present |
| ☑ | ■ | invalid | - | Ligand present |
| ☑ | ■ | valid | - | Ligand present |

*Figure 12. Additional information about ligands*

Considering the large amount of collected data, The *Color Manager node* was used to add a color for each Source entry from PDB to have a better overview about the proteins as it can be seen in Figure 13.



Select one Column: Source

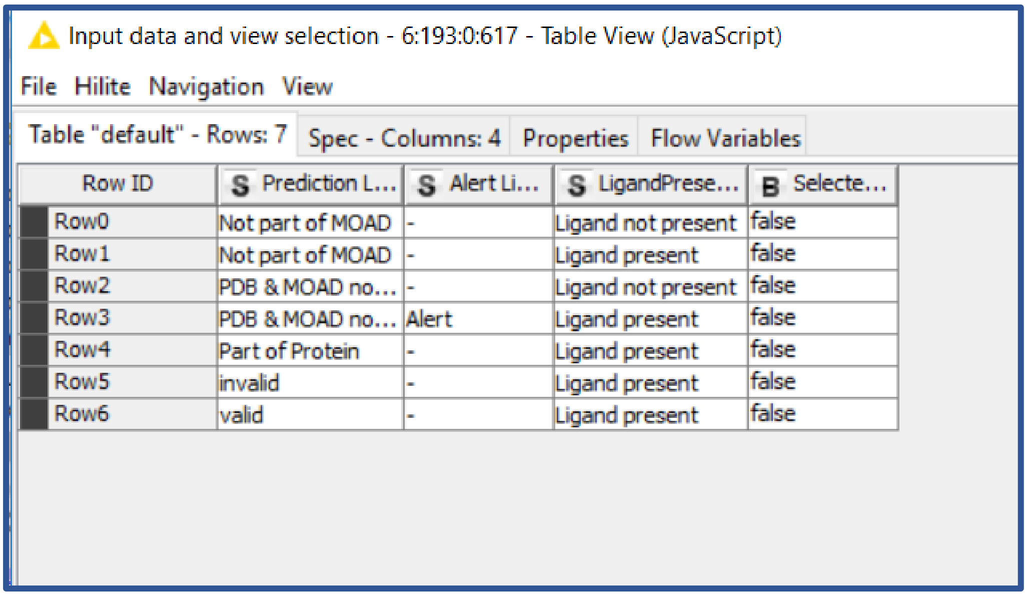
● Nominal ○ Range

- 🟥 Homo sapiens
- 🟪 Escherichia coli,Homo sapiens
- 🟨 Shigella flexneri,Homo sapiens
- 🟩 Desulfovibrio vulgaris,Homo sapiens
- 🟩 synthetic construct
- 🟦 unidentified
- 🟦 Streptococcus sp. 'group G',Homo sapiens
- 🟦 Mus musculus
- 🟪 Symploca
- 🟪 Helicoma ambiens
- 🟥 Homo sapiens,Mus musculus

Preview

*Figure 13. Color Manager node shows the color attributed to the source*

As additional information, the *metanode Interactive Visualization* contains LigandID information of different PDBs. The view allows us to have a better overview whether the Ligand is present or not, and if the structure downloaded from PDB is also present in MOAD or not. In the case when the Ligand is present, additional information is stated such as whether the Ligand is valid or not, or parts of the protein in the PDB and MOAD do not match.



Input data and view selection - 6:193:0:617 - Table View (JavaScript)
File
Hilite
Navigation
View
Table "default" - Rows: 7
Spec - Columns: 4
Properties
Flow Variables

| Row ID | S Prediction L... | S Alert Li... | S LigandPrese... | B Selecte... |
|---|---|---|---|---|
| Row0 | Not part of MOAD | - | Ligand not present | false |
| Row1 | Not part of MOAD | - | Ligand present | false |
| Row2 | PDB & MOAD no... | - | Ligand not present | false |
| Row3 | PDB & MOAD no... | Alert | Ligand present | false |
| Row4 | Part of Protein | - | Ligand present | false |
| Row5 | invalid | - | Ligand present | false |
| Row6 | valid | - | Ligand present | false |

*Figure 14. represents the table where additional information is stated for Ligands of PDB and MOAD*

## *4.5* PDB Saver

Afterwards, to download the selected PDBs we used the *Group Loop start node*, in order to group the PDBs using UniprotID as a criterion. As a next step the *Group by node* was executed to group all PDBs and PDB structures and simultaneously avoid all the duplicates due to different ligands for every PDB chain. However, *Chunk Loop start node* was executed afterwards, because *PDB Saver node* needs single PDBs as an input to download the selected one. In order to download it, the *Java Edit Variable node* was executed to define the location of the subfolders and the name of the subfolders, in this case the UniprotID and the name of the protein (i.e. P10275 - Androgen receptor). The *Create Folder node* was executed to create subfolders containing PDBs inside of the main folder. The *PDB Saver node* was executed to save the PDB copy on the local computer. Using the *Loop End node* each PDB was downloaded and saved in specific subfolders.

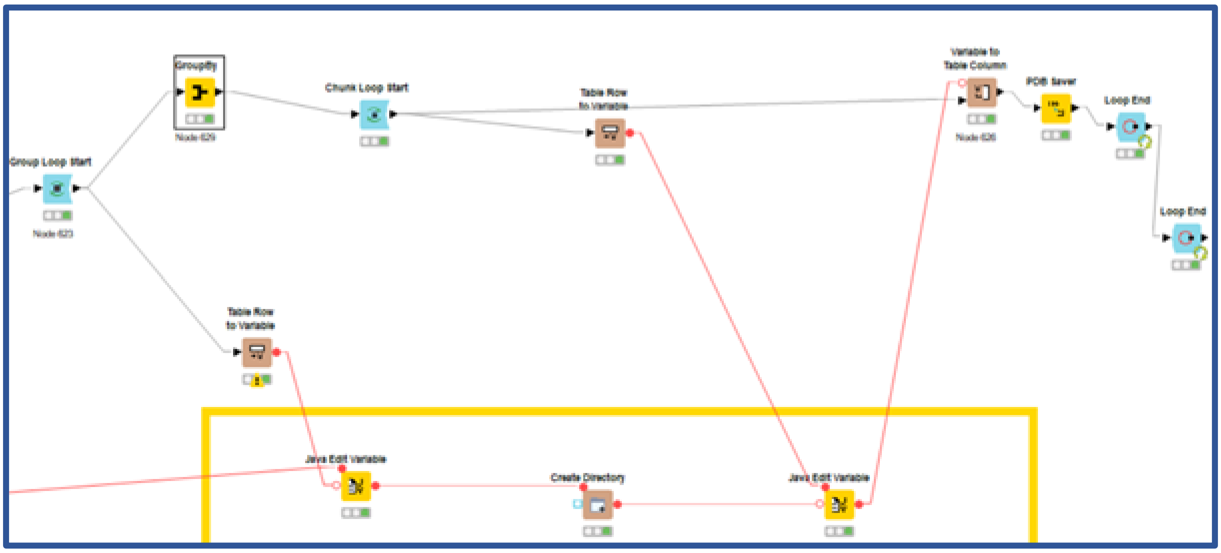


*Figure 15. Knime workflow to save the downloaded PDBs on the local computer*

## *4.6* Ligand extraction

Considering the large amount of downloaded data, we decided to focus only on UniprotIDs derived from humans. However, even after choosing to focus only on human PDBs we still had a large amount of data for structure-based modelling. The next step was to search the literature to determine which of the ligands are involved in causing developmental toxicity. In order to extract all the ligands from data file containing all downloaded data, the *Column Filter node* was used to filter the columns containing LigandID, Ligand Smiles, UniProtID, PDB-ID and InChI Key. The *GroupBy node* grouped all the columns except the UniprotID column (Figure 16).

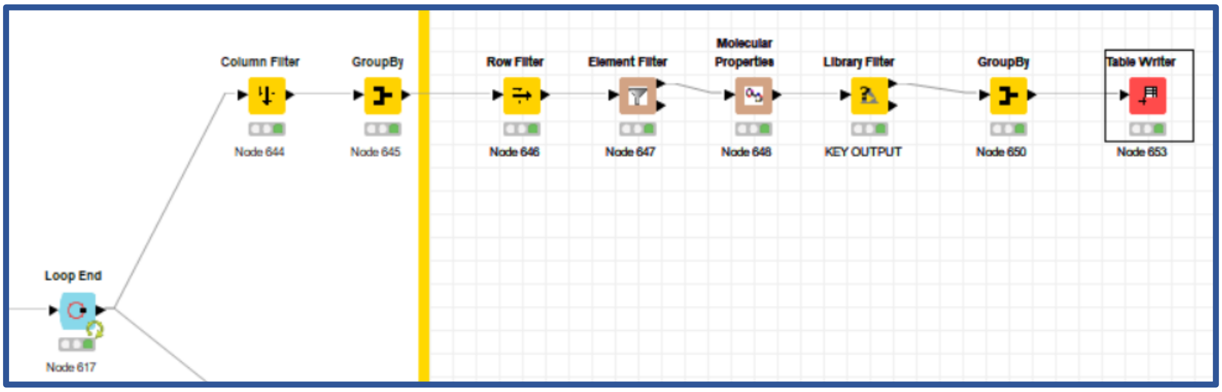
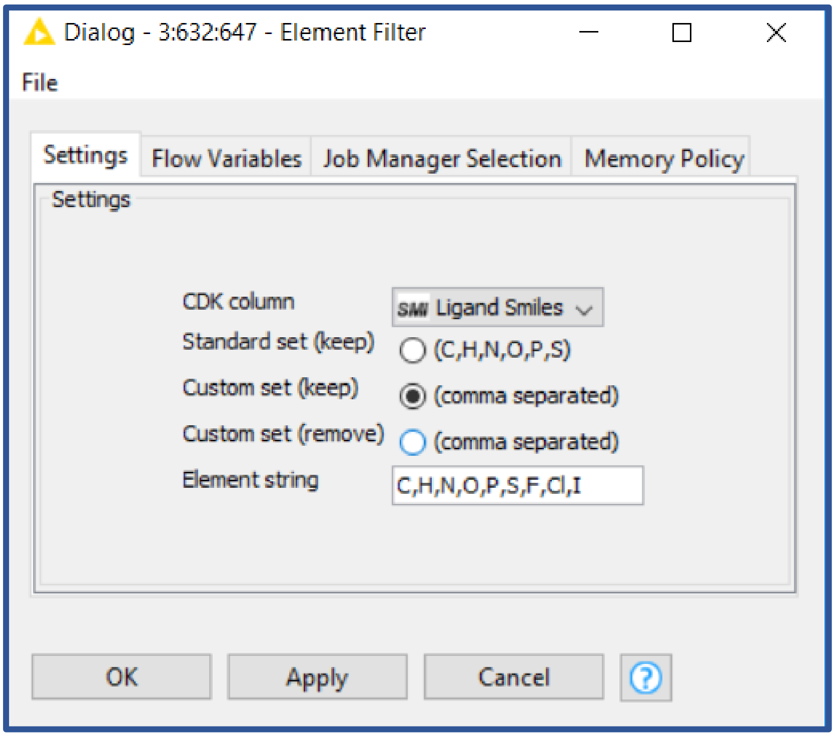


*Figure 16. Knime workflow to extract the ligands*

However, considering that ions or salts are recognized as ligands, the *Element filter node* enables to filter the molecules based on elements. In this case, we decided to keep specific elements in column Ligand Smiles (Figure 17).



CDK column: SMI Ligand Smiles

Standard set (keep) ○ (C,H,N,O,P,S)

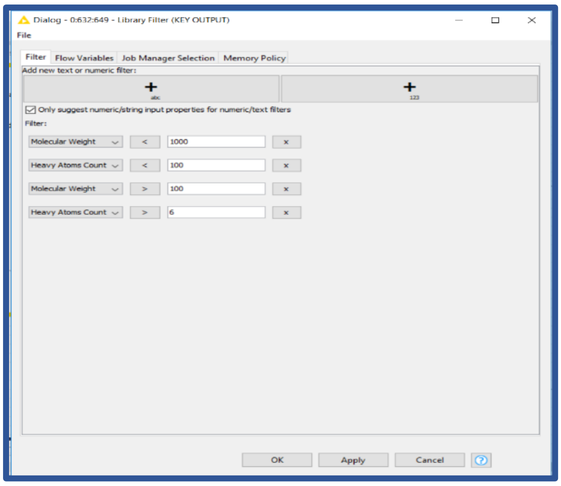
Custom set (keep) ● (comma separated)

Custom set (remove) ○ (comma separated)

Element string: C,H,N,O,P,S,F,Cl,I

*Figure 17. Element Filter node Configure window*

Additionally, with the *node Molecular properties* additional molecular properties were calculated and added to the table, in this case number of heavy atoms and molecular weight. These properties were used as an input for the *Library filter node* in which we filtered the molecules with specific molecular weight and number of heavy atoms to obtain a more appropriate list of the ligands (Figure 18).



Add new text or numeric filter:

☑ Only suggest numeric/string input properties for numeric/text filters

Filter:

```
Molecular Weight     <   1000   x
Heavy Atoms Count    <   100    x
Molecular Weight     >   100    x
Heavy Atoms Count    >   6      x
```

*Figure 18. Represents the characteristics which were chosen to filter the ligands*

## *4.7* Creating structure-based pharmacophore models

127 PDB were retrieved from Protein Data Bank. For each of those a structure-based pharmacophore model was made using LigandScout. After the PDB was retrieved in LigandScout, the pharmacophore features were created by choosing the option *Create a pharmacophore* in LigandScout. After creating the pharmacophores, exclusion volume coat was added to the pharmacophore to increase the selectivity of the pharmacophore. The exclusion volume matches the positions that are sterically claimed by macromolecular environment. Therefore, we had to check whether we could change some exclusion spheres to get more selective models and to possibly increase the enrichment. After adding exclusion volume, the next step was to look whether some distinct feature vectors could be changed, considering the binding site amino acids, distance range of the bond, etc. The hydrogen bond vectors were changed by choosing the option *Convert the selected vector features* if there was the possibility to interact with 2 or more protein atoms and generate in this way too many features.

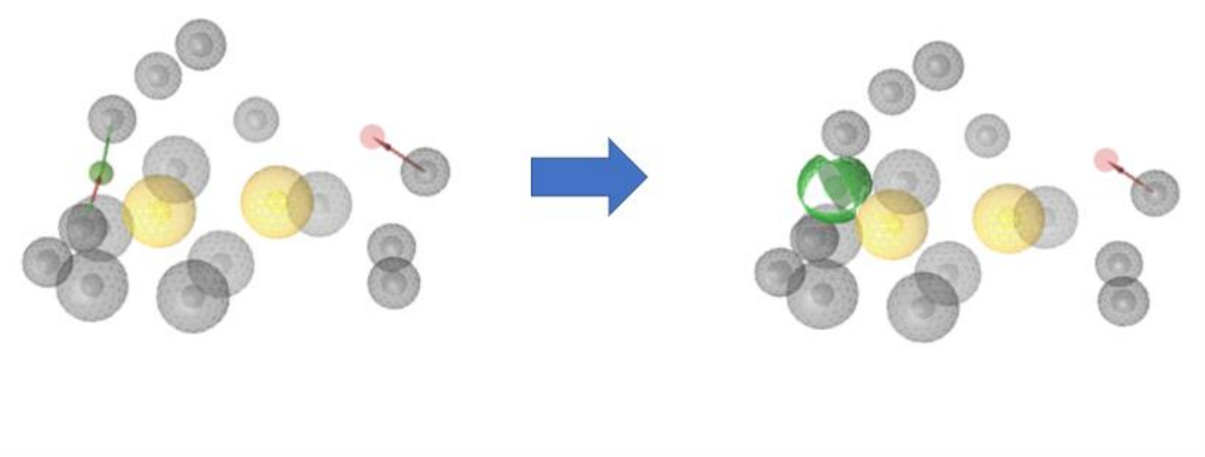


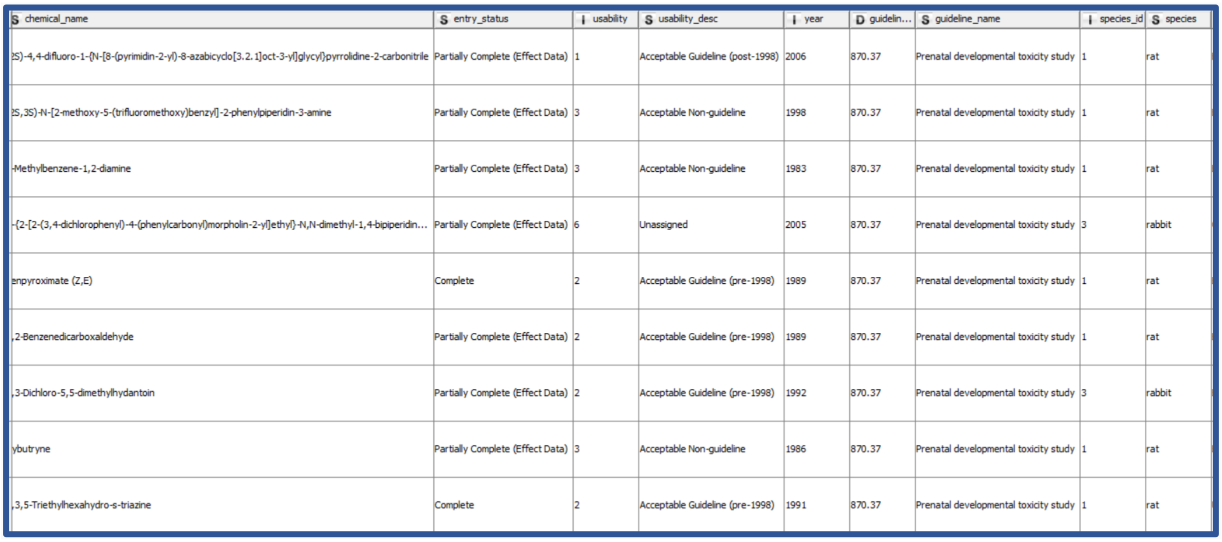
Figure 19. PDB 1T65 before changing pharmacophore features and after

## 4.8 Database selection

In this thesis the following databases were considered:

- *Toxrefdb_nel_lel_noael_loael_summary_AUG2014_FOR_PUBLIC_RELEASE* [31]: The database was taken from Toxcast website, in which all chemicals were tested in vivo whether they cause or not developmental toxicity. This database includes 11.815 compounds. The Toxref file contains columns with information such as chemical name, Loael, Nel, the source of the data, and guidelines. Moreover, the chemicals in this database were divided into categories based on the effect that they were causing, such as maternal, reproductive, developmental, etc. In addition, for each chemical it was stated on which species the experiment was done, what was the highest and what was the lowest used dose. In the LOAEL column, the effect was described with numbers (0,1,2,3) with zero meaning no effect. However, this datafile lacks explanation about the methods how the researcher determined whether the effect of the chemicals was the strongest one or not. For this reasons, we decided not to proceed with it.
- *Toxrefdb_study_tg_effect_endpoint_AUG2014_FOR_PUBLIC_RELEASE* [31]: This database contains the chemicals that definitely caused the lowest observed effect. However, the chemicals were divided into two groups

characterized with 0 and -1. In the zero group the researcher tested chemicals for which they expected to observe LOAEL. In the -1 group the researcher tested chemicals for which they were not sure whether the chemicals could induce developmental toxicity. Moreover, this file does not contain a SMILES column or an InchI Key, but contains the CAS number of the substance, which will be used in the further Knime workflow. This file contains 17.098 compounds (Figure 21). The Toxref dataset was used as active compounds in our virtual screening.



| chemical_name | entry_status | usability | usability_desc | year | guidelin... | guideline_name | species_id | species |
|---|---|---|---|---|---|---|---|---|
| 2S)-4,4-difluoro-1-{N-[8-(pyrimidin-2-yl)-8-azabicyclo[3.2.1]oct-3-yl]glycyl}pyrrolidine-2-carbonitrile | Partially Complete (Effect Data) | 1 | Acceptable Guideline (post-1998) | 2006 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| 2S,3S)-N-[2-methoxy-5-(trifluoromethoxy)benzyl]-2-phenylpiperidin-3-amine | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1998 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| -Methylbenzene-1,2-diamine | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1983 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| -{2-[2-(3,4-dichlorophenyl)-4-(phenylcarbonyl)morpholin-2-yl]ethyl}-N,N-dimethyl-1,4-bipiperidin... | Partially Complete (Effect Data) | 6 | Unassigned | 2005 | 870.37 | Prenatal developmental toxicity study | 3 | rabbit |
| enpyroximate (Z,E) | Complete | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| ,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| ,3-Dichloro-5,5-dimethylhydantoin | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1992 | 870.37 | Prenatal developmental toxicity study | 3 | rabbit |
| ybutryne | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1986 | 870.37 | Prenatal developmental toxicity study | 1 | rat |
| ,3,5-Triethylhexahydro-s-triazine | Complete | 2 | Acceptable Guideline (pre-1998) | 1991 | 870.37 | Prenatal developmental toxicity study | 1 | rat |

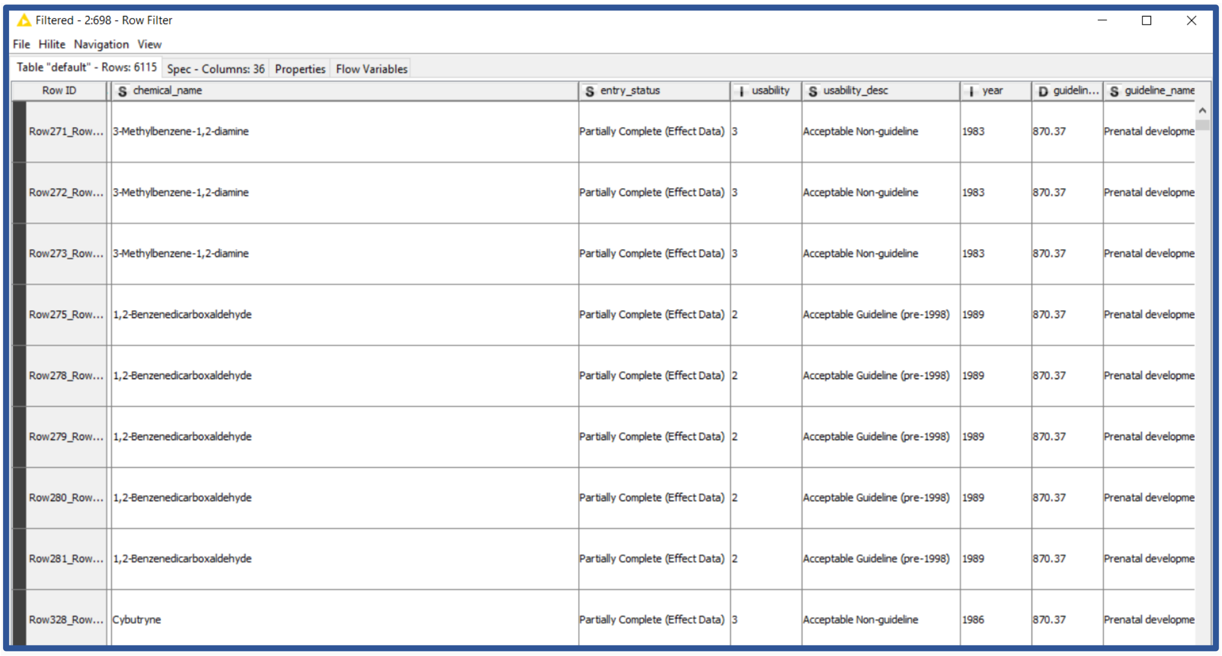
*Figure 20. short overview of the Toxref table*

- *The Drugbank database*: The drugbank compounds were used as inactive compounds for the virtual screening. This database consists of 2.141 compounds.

Furthermore, as it was previously mentioned in 4.6 *Ligand Extraction*, we used the literature to determine a list of ligands that could cause developmental toxicity. The ligands which are involved in causing developmental toxicity were added to the Toxref list under the assumption that they are active compounds.

## *4.9* Dataset preparation

In this thesis the screening was performed with the Toxref dataset which contained 17.098 entries. As it was previously mentioned that this database file does not include the Smiles column that is important for further steps. Therefore, the Toxref file was merged to the Toxcast release file (SDF file), which contains all chemicals which were released from Toxcast or Tox21 database. Tox21 (Toxicology in the 21st century) is a collaboration between several federal agencies with the aim to develop better assessment methods to determine whether certain chemicals can cause negative health effects [32]. The Toxref file and the SDF file were joined together with the *Joiner node*. In the configure window of the *Joiner node*, the option Inner joiner was chosen based on the CAS number of the substance. As a result, we obtained 14.935 chemicals with the *Joiner node*, with the necessary chemical properties of the molecules. This file contains all chemicals that are found to cause different toxicities such as maternal, reproductive etc. With the *Row filter node* all other effect categories were removed from the table except developmental toxicity. At this point the database contained 6.115 chemicals. However, considering that the same chemicals were used in different experiments and different results were obtained, this dataset contains several duplicates as it can be seen in Figure 21.



| Row ID | S chemical_name | S entry_status | I usability | S usability_desc | I year | D guidelin... | S guideline_name |
|---|---|---|---|---|---|---|---|
| Row271_Row... | 3-Methylbenzene-1,2-diamine | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1983 | 870.37 | Prenatal developme |
| Row272_Row... | 3-Methylbenzene-1,2-diamine | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1983 | 870.37 | Prenatal developme |
| Row273_Row... | 3-Methylbenzene-1,2-diamine | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1983 | 870.37 | Prenatal developme |
| Row275_Row... | 1,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developme |
| Row278_Row... | 1,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developme |
| Row279_Row... | 1,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developme |
| Row280_Row... | 1,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developme |
| Row281_Row... | 1,2-Benzenedicarboxaldehyde | Partially Complete (Effect Data) | 2 | Acceptable Guideline (pre-1998) | 1989 | 870.37 | Prenatal developme |
| Row328_Row... | Cybutryne | Partially Complete (Effect Data) | 3 | Acceptable Non-guideline | 1986 | 870.37 | Prenatal developme |

*Figure 21. Small section of Toxref (Data duplicates)*

In order to prepare the dataset for screening, the *node Standard properties* was executed. With this node additional information about molecular weight of the substances was calculated and merged to the table. This output of the *Standard properties node* was used as an input for the *Row filter node* to filter the substances based on their molecular weight. As output 6.025 chemicals were obtained. However, the compounds contained salts as a part of their structure and to remove it the *RDKit salt stripper* was executed.

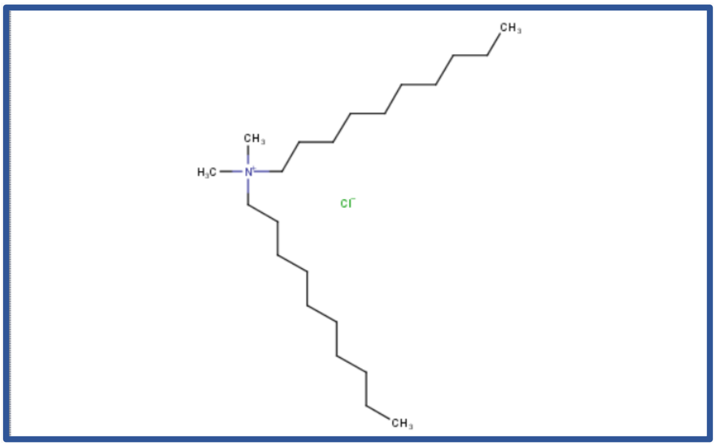
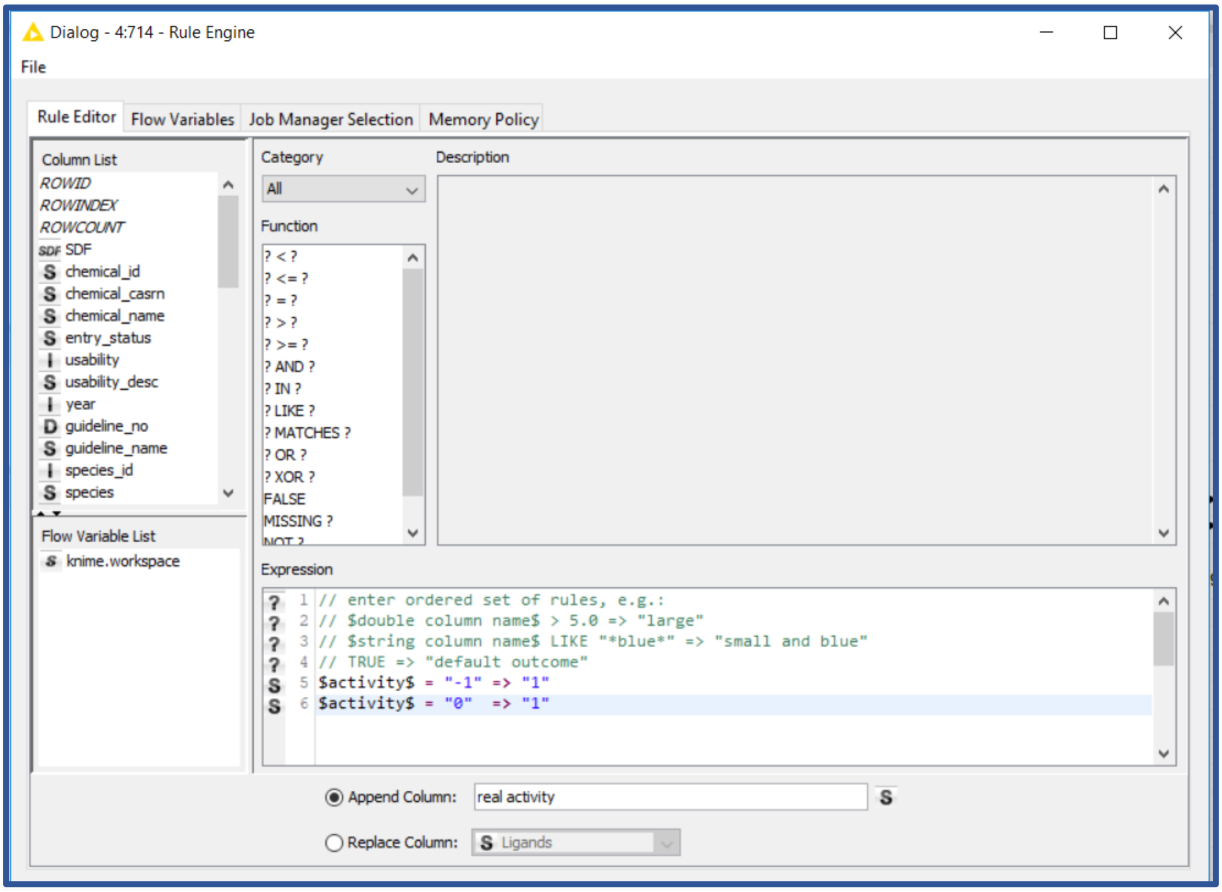


*Figure 22. shows one compound containing salt which was removed with the RDKit salt stripper node*

Considering the potential presence of duplicates, the *node Duplicate remover* was used. The final output of this node contained 468 compounds.

At this point the list of ligands was prepared also for the virtual screening. First, the *CDK to Molecule node* was used to convert the CDK molecules into SDF molecules. Moreover, the *Table Merger node* was executed to merge the Toxref dataset with the Ligand list. Considering both datasets as active compounds the *Rule Engine node* was used to classify all the compounds as active ones (Figure 23).



```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
$activity$ = "-1" => "1"
$activity$ = "0"  => "1"
```

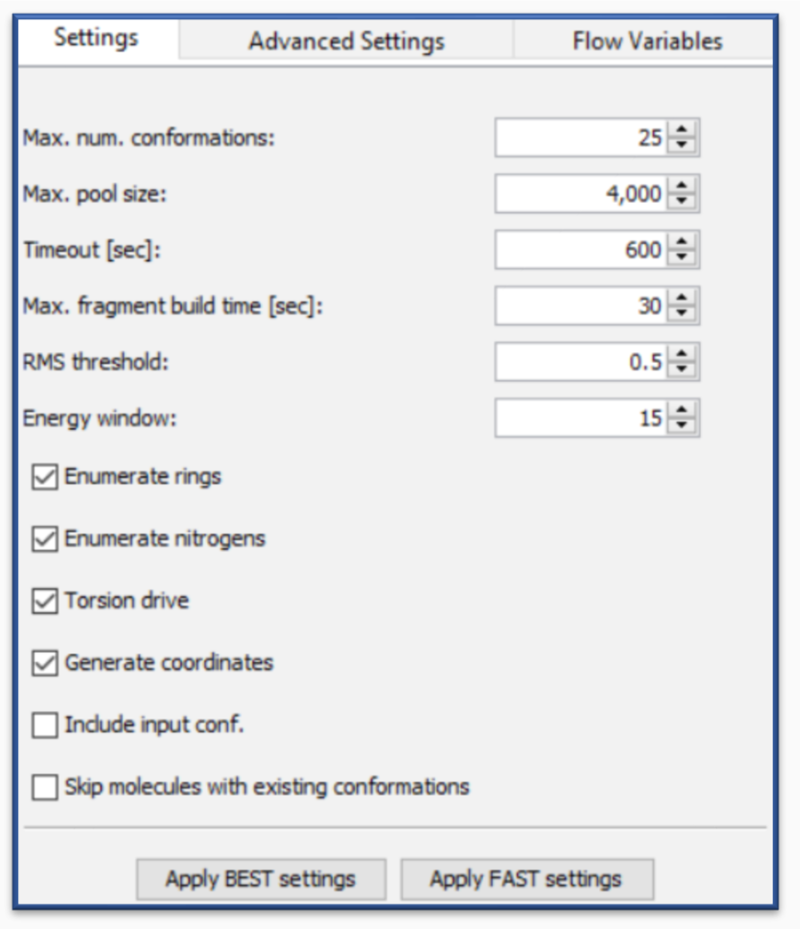
● Append Column: real activity

○ Replace Column: Ligands

*Figure 23. Rule Engine configure window*

Additionally, to avoid duplicates the *node Duplicate Remover* was added to the workflow. The resulting table contains 468 entries.

In order to screen the database with the previously made pharmacophore models, the conformation of the molecules was computed with the *Icon node* as it can be seen in Figure 24.



Max. num. conformations: 25
Max. pool size: 4,000
Timeout [sec]: 600
Max. fragment build time [sec]: 30
RMS threshold: 0.5
Energy window: 15

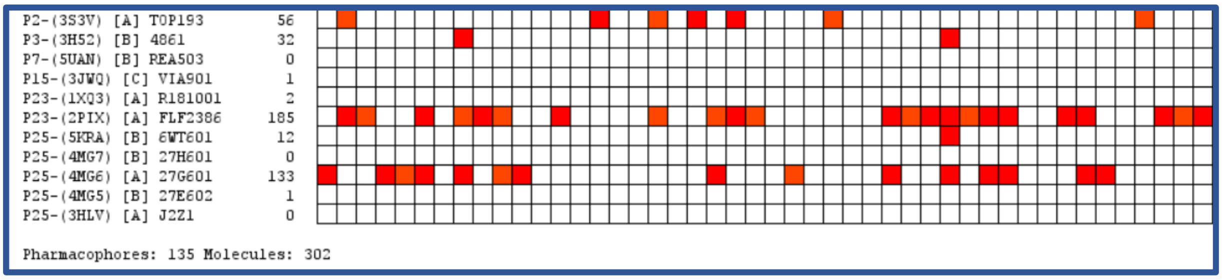
☑ Enumerate rings
☑ Enumerate nitrogens
☑ Torsion drive
☑ Generate coordinates
□ Include input conf.
□ Skip molecules with existing conformations

*Figure 24. Icon configure window*

The output of the Icon node was used for the *LDB Writer node* to create a screening library which can be read in LigandScout (ldb file).

## *4.10* Virtual Screening

Finally, the virtual screening was performed with the Toxref dataset in Knime, as active compounds. As previously mentioned the list with ligands that can cause developmental toxicity according to the literature was added to the Toxref dataset as active compounds. On the other side, the drug bank dataset was used as inactive compounds. In Knime the *node Activity Profiling* was used to screen the database against the pharmacophore models. The result of the screening was displayed in a heat map, where hits were represented with red colors (Figure 25).



```
P2-(3S3V) [A] TOP193      56
P3-(3H52) [B] 4861        32
P7-(5UAN) [B] REA503       0
P15-(3JWQ) [C] VIA901      1
P23-(1XQ3) [A] R181001     2
P23-(2PIX) [A] FLF2386   185
P25-(5KRA) [B] 6WT601     12
P25-(4MG7) [B] 27H601      0
P25-(4MG6) [A] 27G601    133
P25-(4MG5) [B] 27E602      1
P25-(3HLV) [A] J2Z1        0

Pharmacophores: 135 Molecules: 302
```

*Figure 25. Small section of Heat map (active compounds)*

## *4.11* Machine learning methods

Machine learning methods are used widely in order to analyze high-throughput data to solve important biological questions [33]. The structure-based pharmacophore models were made in order to perform virtual screening to get an output data. The virtual screening output was used as an input for different machine learning methods to see if using the SBP information will give models with good predictivity.

First and foremost, the workflow was started with *Column List Loop start node* in order to use each pharmacophore columns in the input table as an iteration. After aggregating all the pharmacophore columns with *Column Aggregator node* and using a function sum to count all the hits, another column with the sum of all hits was added to the table.

## *4.12* Machine learning method 1

The first machine learning method was random forest, through which the sum of the pharmacophore hits was described with a bit vector. Each pharmacophore hit was classified as 1, whereas 0 was used when the pharmacophore did not hit the compound. In this case a bit vector was seen as a sequence of hits (i.e. 0001100010). The Random Forest method uses the sequence of hits bit vector as a pharmacophore description.

In order to perform the validation of the models, the dataset was divided into training (70%) and test set (30%). The random forest method uses decision trees algorithm to build a model out of the data set. The *node Random Forest learner* uses the training set and builds a model based on the cross validation method. The *node Random Forest Predictor* uses the test set to test the model and to make predictions.

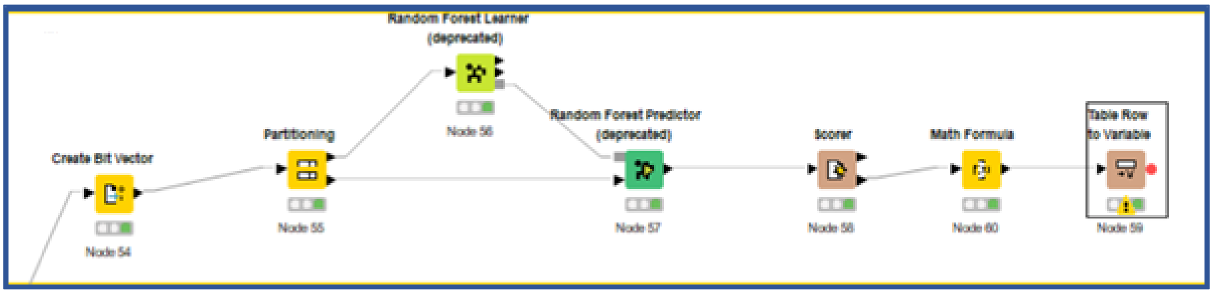


*Figure 26. Machine learning method 1*

Therefore, to evaluate the performance of the model, the parameters sensitivity and specificity were taken into account to test the accuracy of our models [34]. In the random forest algorithm the sensitivity is classified as percentage  of active compounds correctly classified as active, whereas the specificity is referred to as percentage  of inactive  compounds correctly classified as inactive [34]. Specificity and Sensitivity value range from 0 to 1, whereas zero value states that the search did not yield any actives (Sensitivity) or inactive (Specificity). On the other side the value 1 indicates that all actives (Sensitivity) or inactives (Specificity) could be retrieved [18]. Moreover, one important parameter for evaluation of the model performance is balanced accuracy, which in this case was 0,5.

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

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

## *4.13* Machine learning method 2

The second method used as descriptors the output of the pharmacophore screening and, by the means of *Attribute selected classifier node*, tried to identify the best combination of descriptors to build a model. The dataset was divided into training (70%) and test set (30%) with the *Partitioning node*. In order to avoid the column duplicates, the *node Low variance filter* was executed to remove all the column duplicates which could affect the variance. If the variance is too low, the possibility to distract certain machine learning algorithms is much higher. Executing the *node X-Partitioner* the training set was divided into 5 subsets, four used as a training and one as a test set. The training set will be used as an input for the *node Attribute selected classifier*, which makes a model for a training set. To evaluate the model, the *WEKA predictor node* was executed, which uses the test set from *X-Partitioner node* to evaluate the output of *Attribute selected classifier node*. The *Weka predictor node* output serves as an input for the *X-Aggregator* which aggregates cross-validation results and outputs the prediction for the model.

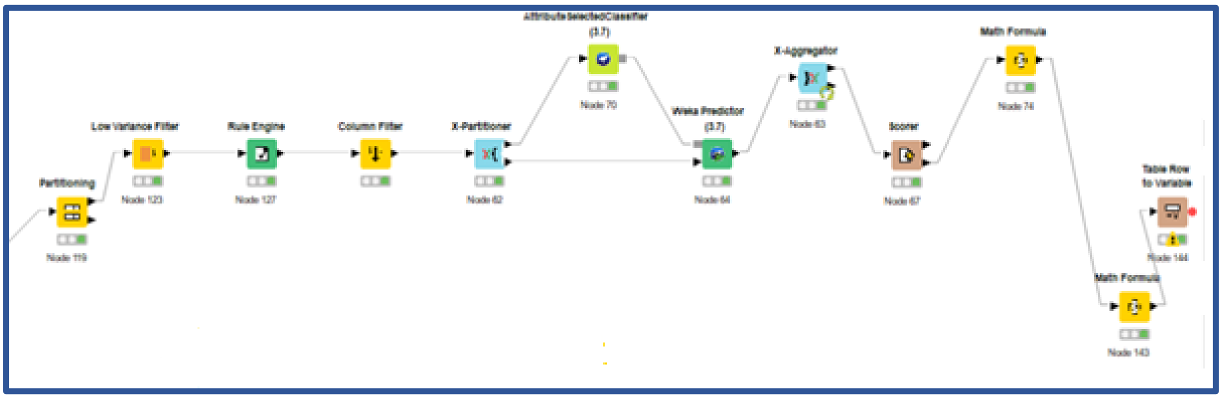


*Figure 27. Machine learning method 2*

## *4.14* Machine learning method 3

In the third method we used a user-defined threshold. The pharmacophore screening results were aggregated under the new column „sum“, that represented how many pharmacophores a compound hits. For example, if the chosen threshold was 4, the molecule was classified as active (1) if it hits 4 or more pharmacophores, otherweise was assigned as inactive (0). The data set was divided into training (70%) and test set (30%) using *Partitioning node*, considering the activity column. The training set was used as an input for *Low variance node*, in order remove double-compatible columns. The execution of the *Reference Column filter* we aggregated the training and test set, whereas the *String to number node* was integrated to convert the string into integer. The output of this node was used for *Scorer node*, that compared two columns by their attribute values and creates confusion matrix.

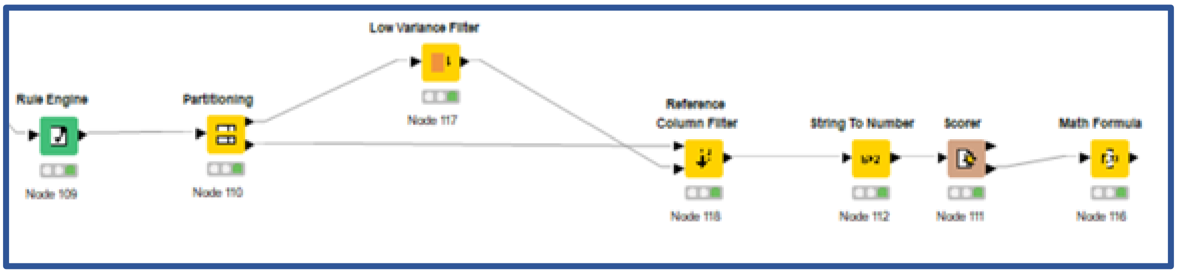


*Figure 27. Machine learning method 3*

## *4.15* Machine learning method 4

In the fourth machine learning method, we calculated the ratio of active and inactive compounds that were retrieved by each pharmacophore, and then we used this value to select only the one with the highest ratio. First the dataset was divided into active and inactive compounds using the activity column. Considering that the table contains duplicate columns, which could affect the machine learning results, the *Low variance node* was executed before dividing the dataset to remove the duplicates. Afterwards, integrating the *Transpose node* allows the swapping of columns and rows. Moreover, for active and inactive compounds the *Column Aggregator node* was executed to sum all the pharmacophores and to use the sum as an input for *Math Formula node* to count the ratio between the actives and the sum. The same workflow was performed for the inactive compounds and both tables were joined with the *Joiner node*. In order to calculate the ratio of active and inactive compounds for each pharmacophore created, the *Math Formula node* was integrated again, and the results were obtained.

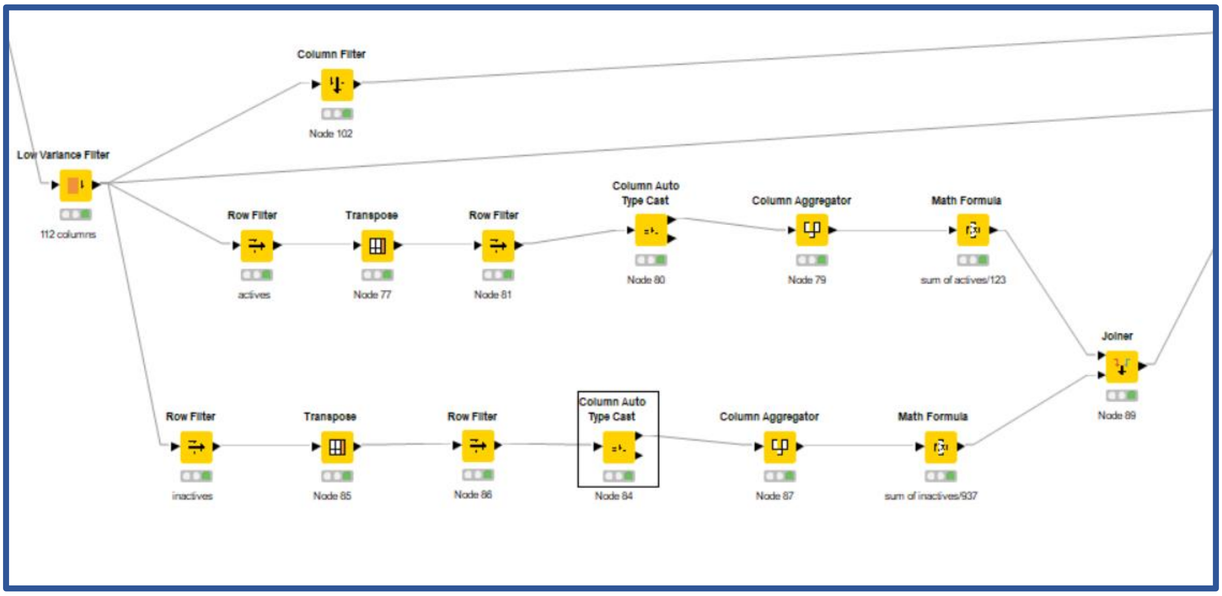


*Figure 28. The analysis of the ratio of actives and inactive*

Furthermore, using the *Row filter node* only those pharmacophores were kept for which the ratio between active and inactive is higher than 1 to see if the balanced accuracy would improve.  Afterwards, the parameters were optimized with the *Parameter optimization Loop Start node* to maximize the parameter to obtain higher balanced accuracy.  The joined dataset was divided into training and test set with the *Partitioning node*. These two sets were used to make a prediction in order to evaluate the pharmacophore model. This part of the workflow was similar to the others before it, however the difference is that in this one the threshold was optimized.

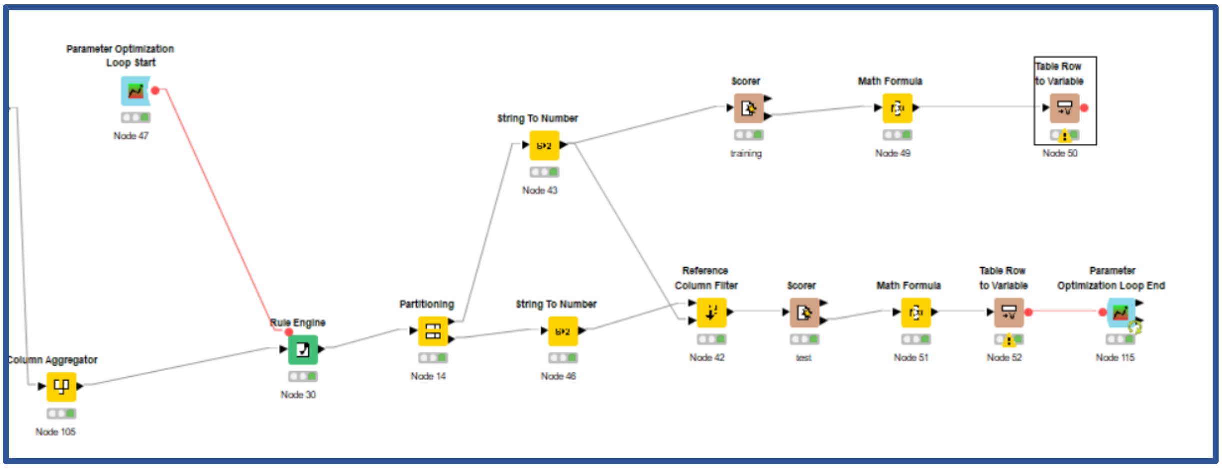


*Figure 29. Optimized threshold*

# 5 DISCUSSION AND RESULTS

First and foremost, this thesis was based on the list of the proteins created by the Crackit Challenge, that could potentially participate in the DART adverse outcome pathway and lead to developmental toxicity. Based on the previously mentioned protein list, using UniprotID as a start point for this research, a Knime workflow was developed to download all PDB-IDs associated with these UniprotIDs. As a result, out of 182 UniprotIDs we obtained 2.170 PDB-IDs. The Protein Data Bank is a large archive of crystal structures, but there are still some crucial disadvantages of PDB. Crystallization products, ions or salts are classified as ligands in PDB, even though they are not. They are all seen as Ligands and therefore it is not possible to differentiate between them.

As a solution for this problem, Binding MOAD was integrated in the Knime workflow to compare it with PDB. The main difference between MOAD and PDB is a strict classification of ligands. MOAD distinguishes between valid and invalid ligands. Therefore, with MOAD invalid ligands such as salts or ions were removed, and only valid ligands were kept.

In the metanode Interactive Visualization we used colors to represent the following characteristics:

- source
- UniprotID and PDB
- ligand presence

Considering that the PDBs obtained with the Knime workflow were not only human, different colors were attributed to different species in order to distinguish them. Furthermore, to each UniprotID specific colors were attributed.

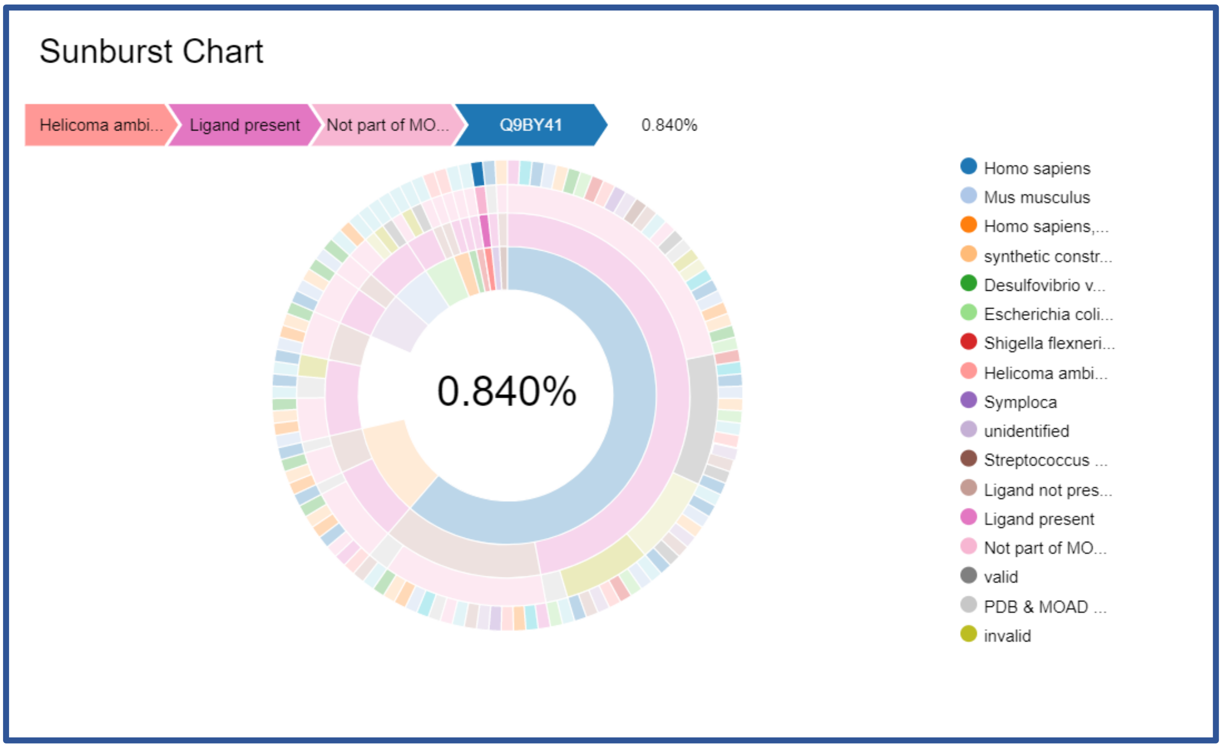


*Figure 30. Sunburst chart represents the source, Ligand presence, whether the Ligand is part of MOAD and UniprotID*

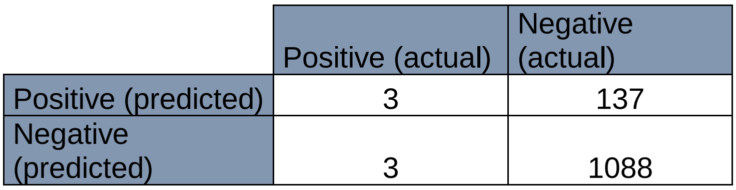
Considering the number of PDBs that were downloaded, we decided that my focus should be only on UniprotIDs that contain at least one human PDB. Therefore, I selected only UniprotIDs which contained at least one human PDB. Finally, I obtained 88 UniprotIDs which corresponded to a total of 2.139 human PDB-IDs. However, considering the still large amount of data, the next step was to determine which of the Ligands in the downloaded PDBs were causing developmental toxicity according to the literature.

The literature used was the Pubmed official site, on which I focused only on in vivo research. However, each research outcome depends on the conditions under which the research was performed. The same drug was used in different experiments and on different species under different conditions. For this reason, it was not possible to clearly label all the compounds. Furthermore, another limitation is in the species differences: sensitivity to the compound or incomplete understanding of the mechanism leading to the AOP between animals and humans [35]. Based on the literature research I obtained 26 UniprotIDs with 127 PDBs with Ligands causing developmental toxicity.

Afterwards, structure-based pharmacophores were made out of 127 PDBs with ligands causing developmental toxicity. Changing the features of the pharmacophores for some of the PDBs was not necessary, as the automatically made pharmacophores in LigandScout were fitting all the criteria previously explained in *Paragraph 4.7*. The pharmacophores were saved and used for screening of a set of actives and inactives. However, the screening results showed that the pharmacophores which were not further elaborated had in general lower level of selectivity.

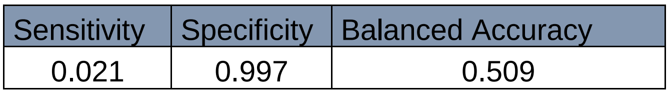
As it was stated before, the screening was performed using compounds from Toxref database as active and those in Drugbank as inactive. The Toxref database contains 468 chemicals tested to determine whether these could lead to developmental toxicity or not. However, the LOAEL is stated with values 1 and 0, whereas 1 stands for the effect observed but not expected and 0 stands for effect was observed and expected. Therefore, all the compounds in this database were classified as active. On the other hand, the Drugbank dataset was used under the assumption that all compounds inside are approved and the possibility to obtain hits from the Drugbank dataset is small. Unfortunately, the obtained screening results showed a different outcome. The screening of the active dataset resulted in 302 active compounds, whereas the screening of the inactive dataset resulted in 2.141 hits.

In order to test the final output of the pharmacophores, different machine learning methods were integrated in the workflow. The pharmacophore models were used as descriptors for each method. One of them was random forest, which used a bit vector based on the hits of the pharmacophores. A bit vector is a sequence of a pharmacophore hits, used as a descriptor. Moreover, it uses training to build a model using different cross-validation algorithms, whereas the test set is used to evaluate the performance of the model. However, the results with this method were not good, considering that the balanced accuracy was 0.502, which is not different from random selection.



| | Positive (actual) | Negative (actual) |
|---|---|---|
| Positive (predicted) | 3 | 137 |
| Negative (predicted) | 3 | 1088 |

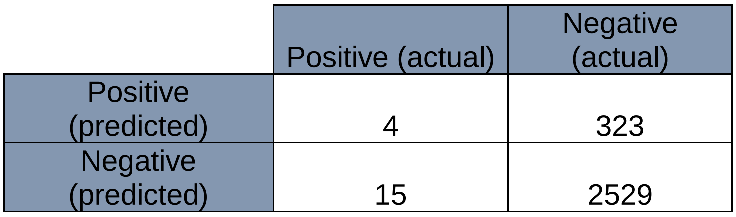
*Figure 31. Confusion matrix (Machine learning method 1)*



| Sensitivity | Specificity | Balanced Accuracy |
|---|---|---|
| 0.021 | 0.997 | 0.509 |

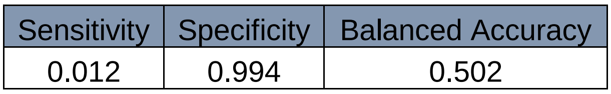
*Figure 32. Sensitivity, Specifity and Balanced accuracy value (Machine learning method 1)*

The second method was a stratified sampling method, in which the training set was split into small subsets, whereas the last subset is seen as test set. This method also uses a training set to build a model, which was evaluated with the test set and resulted in different number of compounds compared to other machine learning methods (Figure 34). However, also the results obtained with this method were not satisfying, the balanced accuracy was 0,502.



| | Positive (actual) | Negative (actual) |
|---|---|---|
| Positive (predicted) | 4 | 323 |
| Negative (predicted) | 15 | 2529 |

*Figure 33. Confusion matrix (Machine learning method 2)*



| Sensitivity | Specificity | Balanced Accuracy |
|---|---|---|
| 0.012 | 0.994 | 0.502 |

*Figure 34. Sensitivity, Specifity and Balanced accuracy values (Machine learning method 2)*

In the third machine learning method the data set was divided into training set and test set to build a model. In order to consider whether the compound is active or inactive the threshold was specified by the user. Even with different threshold values (from 1 to 5), the balanced accuracy did not improve (Figure 36).

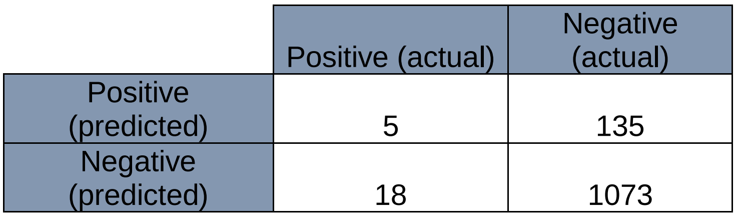
|  | Positive (actual) | Negative (actual) |
|---|---|---|
| Positive (predicted) | 38 | 102 |
| Negative (predicted) | 234 | 857 |

*Figure 35. Confusion matrix (Machine learning method 3)*

| Sensitivity | Specificity | Balanced Accuracy |
|---|---|---|
| 0.271 | 0.786 | 0.528 |

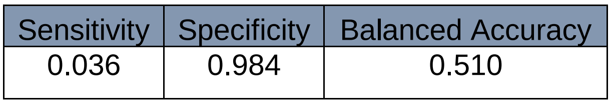
*Figure 36. Sensitivity, Specifity, Balanced accuracy values (Machine learning method 3)*

In the fourth attempt the ratio of the actives and inactives was calculated and only those pharmacophores were kept for which the ratio was higher than 1. The active and inactive compounds were joined again and the dataset was divided into training and test set to build a model. In this method the threshold was automatically optimized. Even though we maximized the parameter to improve the balanced accuracy the results were only slightly different (Figure 39).



| | Positive (actual) | Negative (actual) |
|---|---|---|
| Positive (predicted) | 5 | 135 |
| Negative (predicted) | 18 | 1073 |

*Figure 37. Confusion matrix (Machine learning method 4)*



| Sensitivity | Specificity | Balanced Accuracy |
|---|---|---|
| 0.036 | 0.984 | 0.510 |

*Figure 38. Sensitivity, Specifity and Balanced accuracy (Machine learning method 4)*

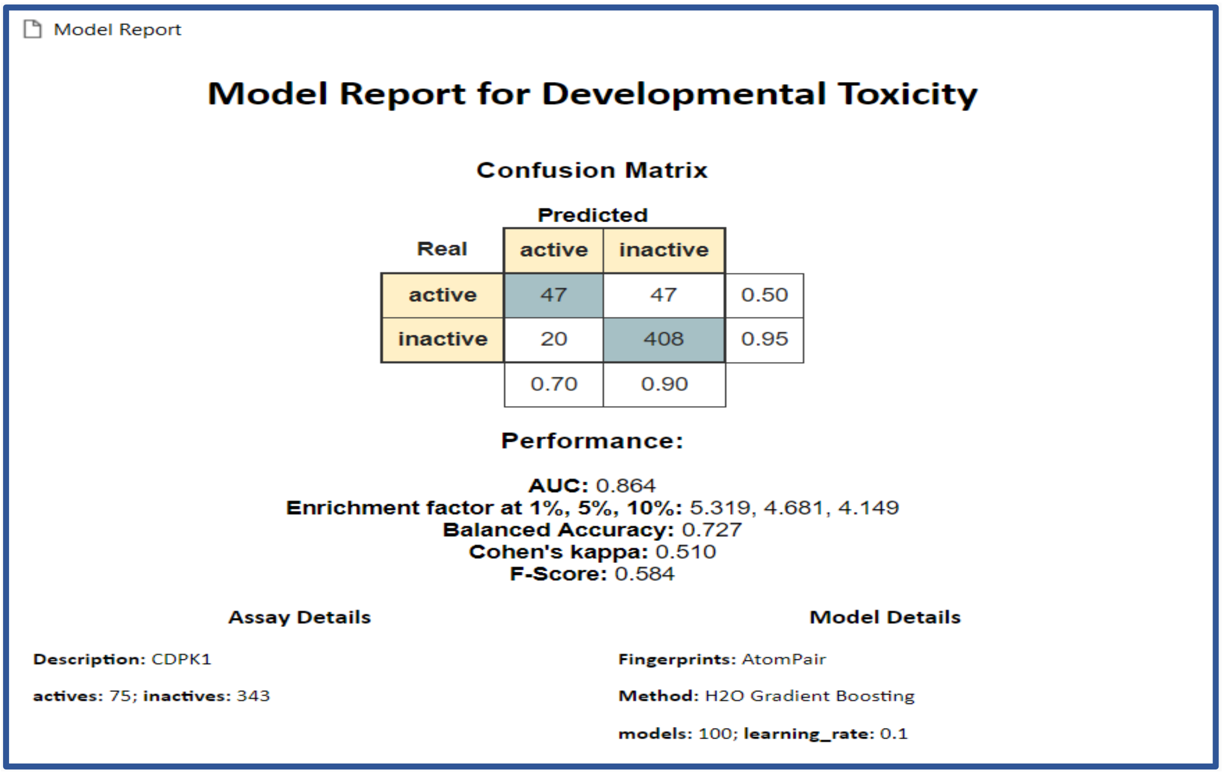
The methods described above used pharmacophore models as descriptors for model making. As it was stated before, the pharmacophore models did not show the necessary selectivity. Moreover, machine learning methods were used to test whether the final output of the virtual screening by pharmacophores would give good predictions. In all four cases the output was not different from random selection.

However, towards the end of the production of this thesis, a workflow for automatic model generation was published by the Knime team. The Knime workflow 04_Analytics/11_Optimization/08_Model_Optimization_and_Selection (Comparison Workflow) was used as a reference. It uses an advanced parameter optimization protocol with four different machine learning methods. Using a dataset prepared for virtual screening, the following fingerprints were calculated and deployed as descriptors for developing models and for evaluation of the model performance.

- ECFC6
- ECFP4
- ECFP6
- AtomPair
- RDKit

The dataset was divided into training (80%) and test set (20%) using node Partitioning. Different metanodes were included into the workflow to optimize the parameters and to build a model for developmental toxicity.

Than we run the before mentioned workflow with our dataset, to compare the results. This allowed us to see if the results obtained with our method would be attributable to the dataset selection, or to the inability of the pharmacophore models to prefer actives over inactives. As it can be seen in Figure 40, the balanced accuracy was 0.72, which is considerably better than the results obtained with the machine learning methods that use the pharmacophore output as an input.



Model Report

**Model Report for Developmental Toxicity**

**Confusion Matrix**

|  | **Predicted** |  |  |
|---|---|---|---|
| **Real** | **active** | **inactive** |  |
| **active** | 47 | 47 | 0.50 |
| **inactive** | 20 | 408 | 0.95 |
|  | 0.70 | 0.90 |  |

**Performance:**

**AUC:** 0.864
**Enrichment factor at 1%, 5%, 10%:** 5.319, 4.681, 4.149
**Balanced Accuracy:** 0.727
**Cohen's kappa:** 0.510
**F-Score:** 0.584

**Assay Details**

**Description:** CDPK1

**actives:** 75; **inactives:** 343

**Model Details**

**Fingerprints:** AtomPair

**Method:** H2O Gradient Boosting

**models:** 100; **learning_rate:** 0.1

*Figure 39. The end result of the Comparison Workflow*

This significantly better result showed that the poor performance obtained with our pharmacophore models was not caused by the compound selection. On the other side, the machine learning methods used in my workflow were using the result of pharmacophore screening as descriptor. This result suggests that the poor performance obtained with our models was not the dataset selection itself. However, there are many reasons which could lead to this result.

1. This thesis was started based on the list of protein targets which are supposed to be involved in developmental toxicity provided by the Crackit Challenge. In most of the cases there is no confirmation that these proteins cause developmental toxicity. Considering the large amount of data we obtained by downloading the PDBs, we focused only on those PDBs having ligands which are toxic according the literature. Therefore, there is a probability of having toxic ligands in other PDBs which were not taken into account, because there are no studies done with the aim to determine if those ligands could lead to developmental toxicity.
2. The pharmacophore models made were based on molecules tested in vivo, not on enzymes. This means that overall it was not possible to have a model validation for the pharmacophores using the dataset.
3. The Drugbank compounds were used as inactive compounds under the assumption that my pharmacophore models will not hit any of the Drugbank molecules.

# 6 CONCLUSION

The developmental toxicity field is getting a lot of attention. Considering the fact that most of the research is performed on animals, which is time consuming and expensive, introducing new computational methods is a reasonable way to try to discover to which extent some chemicals could affect the development of an embryo. Furthermore, keeping in mind the harm that these animals must experience, in order for chemicals to be tested is also a great disadvantage. Therefore, the Crackit Challenge 26 is aiming to reduce the animal-based research in this field in order to get more reliable results and to reduce danger for animals.

Within this diploma thesis my aim was to discover whether the inclusion of structural information in the model generation would lead to models with good predictivity. This was realized by using the output of the virtual screening pharmacophore as an input for machine learning methods. The results obtained with random forest or stratified sampling methods were not different from random selection considering that balanced accuracy was 0,5. To evaluate if the problem was residing in the dataset itself, we run the dataset in a comparison workflow which includes only information from the ligands. The best model generated had a balanced accuracy of 0,72.

Unfortunately, the poor predictivity of the results obtained with our pharmacophore-based methods are the sum of a series of limitations and approximations that had to be made. For this reason, at the moment, this approach is mainly limited by the lack of:

- a validated list of proteins directly involved in Developmental toxicity
- a dataset of the molecules tested on the target and a consequent database of proven inactive compounds
- the number of available crystal structures

When the aforementioned limitations will be overcome, this approach might result to be a valid tool to investigate developmental toxicity from the in-silico perspective.

## *6.1* TABLE OF ABBREVIATIONS

| ***Abbreviation*** | ***Meaning*** |
|---|---|
| Uniprot | Universal Protein Database |
| PDB | Protein Databank |
| Toxref database | Toxicity Reference Database |
| DART | Developmental and reproductive toxicity |
| MOAD | Mother of all databases |
| TP | True positive |
| TN | True negative |
| FP | False positive |
| FN | False negative |
| LOAEL | Lowest observed adverse effect level |
| NOAEL | No observed adverse effect level |
| OECD | Organization for economic Co-operation and development |
| AOP | Adverse observed effect |

## *6.2* TABLE OF FIGURES

## *6.3* BIBLIOGRAPHY

[1] „Reproductive and Developmental Toxicity“. [Online]. Verfügbar unter: https://www.chemsafetypro.com/Topics/CRA/Developmental_and_Reproductive_Toxicity.html. [Accessed: 20-Nov-2018].

[2] „Reproductive & Developmental Toxicity“, *AltTox.org*. [Online]. Verfügbar unter: http://alttox.org/mapp/toxicity-endpoints-tests/reproductive-developmental-toxicity/. [Accessed: 28-März-2018].

[3] „28.2 Embryonic Development – Anatomy and Physiology“. [Online]. Verfügbar unter: https://opentextbc.ca/anatomyandphysiology/chapter/28-2-embryonic-development/. [Accessed: 12-Aug-2018].

[4] „Mammalian Developmental Biology Portal - LifeMap Discovery“. [Online]. Verfügbar unter: https://discovery.lifemapsc.com/in-vivo-development. [Accessed: 25-Juli-2018].

[5] E. Gilbert-Barness, „Teratogenic Causes of Malformations“, *Ann. Clin. Lab. Sci.*, Bd. 40, Nr. 2, S. 99–114, März 2010.

[6] A. B. Raies und V. B. Bajic, „In silico toxicology: computational methods for the prediction of chemical toxicity“, *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, Bd. 6, Nr. 2, S. 147–172, März 2016.

[7] E. Fritsche *u. a.*, „OECD/EFSA workshop on developmental neurotoxicity (DNT): The use of non-animal test methods for regulatory purposes“, *ALTEX - Altern. Anim. Exp.*, Bd. 34, Nr. 2, S. 311–315, Mai 2017.

[8] R. Tsuji und K. M. Crofton, „Developmental neurotoxicity guideline study: Issues with methodology, evaluation and regulation*“, *Congenit. Anom.*, Bd. 52, Nr. 3, S. 122–128, Sep. 2012.

[9] N. S. Sipes *u. a.*, „Predictive Models of Prenatal Developmental Toxicity from ToxCast High-Throughput Screening Data“, *Toxicol. Sci.*, Bd. 124, Nr. 1, S. 109–127, Nov. 2011.

[10] K. C. Brannen, J. M. Panzica-Kelly, T. L. Danberry, und K. A. Augustine-Rauch, „Development of a zebrafish embryo teratogenicity assay and quantitative prediction model“, *Birth Defects Res. B. Dev. Reprod. Toxicol.*, Bd. 89, Nr. 1, S. 66–77, Feb. 2010.

[11] J. Conde-Vancells *u. a.*, „Combining mouse embryonic stem cells and zebrafish embryos to evaluate developmental toxicity of chemical exposure“, *Reprod. Toxicol.*, Bd. 81, S. 220–228, Okt. 2018.

[12] „Challenge 26: DARTpaths | CrackIT“. [Online]. Verfügbar unter: https://crackit.org.uk/challenge-26-dartpaths. [Accessed: 02-Dez-2018].

[13] „Crack IT Challenges | CrackIT“. [Online]. Verfügbar unter: https://crackit.org.uk/crack-it-challenges. [Accessed: 28-Okt-2018].

[14] M. R. Berthold *u. a.*, „KNIME: The Konstanz Information Miner“, in *Data Analysis, Machine Learning and Applications*, C. Preisach, H. Burkhardt, L. Schmidt-Thieme, und R. Decker, Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 319–326.

[15] S.-Y. Yang, „Pharmacophore modeling and applications in drug discovery: challenges and recent advances“, *Drug Discov. Today*, Bd. 15, Nr. 11, S. 444–450, Juni 2010.

[16] D. Schneidman-Duhovny, O. Dror, Y. Inbar, R. Nussinov, und H. J. Wolfson, „Deterministic Pharmacophore Detection via Multiple Flexible Alignment of Drug-Like Molecules“, *J. Comput. Biol.*, Bd. 15, Nr. 7, S. 737–754, Sep. 2008.

[17] A. Vuorinen und D. Schuster, „Methods for generating and applying pharmacophore models as virtual screening filters and for bioactivity profiling“, *Methods*, Bd. 71, S. 113–134, Jan. 2015.

[18] G. Wolber und W. Sippl, „Chapter 21 - Pharmacophore Identification and Pseudo-Receptor Modeling“, in *The Practice of Medicinal Chemistry (Fourth Edition)*, C. G. Wermuth, D. Aldous, P. Raboisson, und D. Rognan, Hrsg. San Diego: Academic Press, 2015, S. 489–510.

[19] „LigandScout User Manual“, S. 143.

[20] G. Wolber und T. Langer, „LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters“, *J. Chem. Inf. Model.*, Bd. 45, Nr. 1, S. 160–169, Jan. 2005.

[21] L. De Luca *u. a.*, „Structure-based screening for the discovery of new carbonic anhydrase VII inhibitors“, *Eur. J. Med. Chem.*, Bd. 71, S. 105–111, Jan. 2014.

[22] P. W. Rose *u. a.*, „The RCSB protein data bank: integrative view of protein, gene and 3D structural information“, *Nucleic Acids Res.*, Bd. 45, Nr. Database issue, S. D271, Jan. 2017.

[23] H. M. Berman, G. J. Kleywegt, H. Nakamura, und J. L. Markley, „The Future of the Protein Data Bank“, *Biopolymers*, Bd. 99, Nr. 3, S. 218–222, März 2013.

[24] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura, und S. Velankar, „Protein Data Bank (PDB): The Single Global Macromolecular Structure Archive“, *Methods Mol. Biol. Clifton NJ*, Bd. 1607, S. 627–641, 2017.

[25] M. W. Parker, „Protein Structure from X-Ray Diffraction“, *J. Biol. Phys.*, Bd. 29, Nr. 4, S. 341–362, 2003.

[26] R. A. Laskowski, „Protein Structure Databases“, in *Data Mining Techniques for the Life Sciences*, Bd. 1415, O. Carugo und F. Eisenhaber, Hrsg. New York, NY: Springer New York, 2016, S. 31–53.

[27] A. Ahmed, R. D. Smith, J. J. Clark, J. B. Dunbar, und H. A. Carlson, „Recent improvements to Binding MOAD: a resource for protein–ligand binding affinities and structures“, *Nucleic Acids Res.*, Bd. 43, Nr. D1, S. D465–D469, Jan. 2015.

[28] R. D. Smith, L. Hu, J. A. Falkner, M. L. Benson, J. P. Nerothin, und H. A. Carlson, „Exploring protein–ligand recognition with Binding MOAD“, *J. Mol. Graph. Model.*, Bd. 24, Nr. 6, S. 414–425, Mai 2006.

[29] M. L. Benson *u. a.*, „Binding MOAD, a high-quality protein ligand database“, *Nucleic Acids Res.*, Bd. 36, Nr. Database, S. D674–D678, Dez. 2007.

[30] M. T. Martin, R. S. Judson, D. M. Reif, R. J. Kavlock, und D. J. Dix, „Profiling Chemicals Based on Chronic Toxicity Results from the U.S. EPA ToxRef Database“, *Environ. Health Perspect.*, Bd. 117, Nr. 3, S. 392–399, März 2009.

[31] O. US EPA, „Exploring ToxCast Data: Downloadable Data“, *US EPA*, 01-Nov-2017. [Online]. Verfügbar unter: https://www.epa.gov/chemical-research/exploring-toxcast-data-downloadable-data. [Accessed: 02-Dez-2018].

[32] O. US EPA, „Toxicology Testing in the 21st Century (Tox21)“, *US EPA*, 31-Aug-2015. [Online]. Verfügbar unter: https://www.epa.gov/chemical-research/toxicology-testing-21st-century-tox21. [Accessed: 03-Dez-2018].

[33] B. F. F. Huang und P. C. Boutros, „The parameter sensitivity of random forests“, *BMC Bioinformatics*, Bd. 17, Nr. 1, Sep. 2016.

[34] N. Triballeau, F. Acher, I. Brabet, J.-P. Pin, und H.-O. Bertrand, „Virtual Screening Workflow Development Guided by the “Receiver Operating Characteristic” Curve Approach. Application to High-Throughput Docking on Metabotropic Glutamate Receptor Subtype 4“, *J. Med. Chem.*, Bd. 48, Nr. 7, S. 2534–2547, Apr. 2005.

[35] T. B. Knudsen, R. J. Kavlock, G. P. Daston, D. Stedman, M. Hixon, und J. H. Kim, „Developmental toxicity testing for safety assessment: new approaches and technologies“, *Birth Defects Res. B. Dev. Reprod. Toxicol.*, Bd. 92, Nr. 5, S. 413–420, Okt. 2011.

## *6.4* APPENDIX

List of protein targets or biological pathways that may participate in the molecular iniatiating event of a DART adverse outcome pathway.

| ***Name of Target Protein/pathway*** | ***Target/Receptor Code*** |
|---|---|
| Androgen Receptor | AR |
| Aryl hydrocarbon | Ah |
| Bone protein-matrix gla protein | MGP |
| Cyclooxygenase-1 | COX1 |
| Cytochrome P450 (CYP26) | CYP26 |
| Cytochrome P450 aromatase (CYP19) | CYP19 |
| Dihydrofolate reductase | DHFR |
| FGF signalling pathway | FGFR |
| Hedgehog signalling pathway | SHH |
| Hedgehog signalling pathway | PTCH |
| Hedgehog signalling pathway | SMO |
| Histone deacetylase | HDAC |
| N-methyl-D-aspartate-receptors | NMDA |
| Oestrogen Receptor: alpha | Era |
| Oestrogen Receptor: beta | Erb |
| Peroxisome proliferator activated receptor | PPARA |
| Retinoic acid receptor (alpha) | RARA |
| Retinoic acid receptor (beta) | RARB |
| Retinoic acid receptor (gamma) | RARC |
| Thymidylate synthase inhibition | TYMS |
| Thyroid hormone receptor (alpha) | TR (alpha) |
| Thyroid hormone receptor (beta) | TR (beta) |
| Microtubule depolymerisation | TUB |
| Microtubule stabilisation | TUB |
| VEGF signalling pathway | VEGFR2 |
| WNT signalling pathway | WNT |

| | |
|---|---|
| Cereblon | CRBN |
| Acetyl-CoA carboxylase | ACC1/2 |
| Copper chelation | |
| dihydroorotate dehydrogenase inhibition | dhod |
| HPPD inhibition | hpd |
| orthosteric nAChR agonists | nAChR (embryonic) |
| 5alpha Reductase | SRD5A2, also SRD5A1 & SRD5A3 |
| Acetylcholinesterase Inhibition | AChE |
| Angiotensin II receptor antagonist | AGTR1, AGTR2 |
| Angiotensin-converting enzyme (ACE) | ACE |
| Carbonic anhydrase | |
| DNA polymerases | |
| GABA-A receptor agonist | GABARA |
| Glucocorticoid receptor | GR |
| Lysyl oxidase | |
| Opiate agonist | ZOR, MOR, other subtypes? |
| Other enzymes involved in folate production & inhibition | |
| Phosphodiesterases | |
| Reductase involved in Vitamin K recycling | |
| Ribonucleotide diphosphate reductase | |
| Type III deiodinase | DIO3 |
| Vitamin D receptor | VDR |
| Farnesyl pyrophosphase synthetase | FPPS |
| GABA A receptor antagonists | GABARA |
| mevalonate / cholesterol pathway | CYP51 |
| mevalonate / cholesterol pathway | CYP17 |
| mevalonate / cholesterol pathway | INSIG1, INSIG2 |
| mevalonate / cholesterol pathway | Sc5d |
| mevalonate / cholesterol pathway | Dhcr24 |

| | |
|---|---|
| mevalonate / cholesterol pathway | DHCR7 |
| mevalonate / cholesterol pathway | NSDHL |