Correlating Cell Painting data with Bioassay Endpoints and Predicting Drug Structure Dependent Mechanisms of Action

A comparative study using PubChem assays and chemoinformatics tools

Master thesis by Luis Vollmers

Date of submission: March 5, 2021

1. Review: Prof. Dr. Katja Schmitz
2. Review: Dr. Andreas Bender

Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

University of Cambridge
Department of Chemistry
Bender Group

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Luis Vollmers, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 5. März 2021

L. Vollmers

Contents

1 Summary

This work concerns the exploration and generation of novel insights into the cell-painting (CP) data set published by Bray et al.[1] In a computational approach, the predictive power of CP descriptors is explored by predicting the endpoints of 52 PubChem assays. The 52 PubChem assays are selected for their relation to cytotoxicity[2] and their comparably large overlap with the CP data set. Their predictive power is tested against structural fingerprints, i.e. extended-connectivity fingerprint (ECFP), using a random forest classifier (RFC). Afterwards, feature engineering selects the most important features from both descriptor sets and another RFC is trained on the combined data set. The evaluation of the prediction metrics illuminates which strengths and shortcomings the morphological fingerprints have compared to the structural fingerprints. It turns out, that certain PubChem assays are generally better predicted with CP features whereas other bioassays have higher predictive potential when using ECFP. With this information as background, certain trends within the specificity and the sensitivity are uncovered. ECFP comprise higher specificity compared to CP data which shows higher sensitivity. It is argued, that CP is more suitable for toxicity prediction and drug safety based on these results.
Additionally, attempts are made to link the difference in general predictive capability between the PubChem assays to cellular morphology. The CP data's roots lie in cellular fluorescence microscopy. Five fluorescent channels have been used for imaging and these channels correspond to certain cell organelles.[3] Based on the fluorescent channels

an enrichment measure is introduced that is calculated for two groups of PubChem assays identified by their difference in predictive potential. As a final step phenotypic terms are manually generated for categorization of the different PubChem assays. These terms correspond to cellular mechanisms or morphological processes. The phenotypic terms are generated unbiasedly. A bioassay may or may not be associated with a phenotypic annotation and therefore an enrichment measure is defined for these terms as well. The phenotypic annotations that are found to be enriched for better performing PubChem assays might be able to guide the pre-selection of bioassays that are to be predicted with CP descriptors.

2 Introduction

Currently, pharmacological drug development focuses on well-established biochemistry based approaches to find and release new drugs. However, the challenges these methods face are manifold. High costs related to commercialization, drug failure rates and various clinical trials bottleneck the industry as a whole. Another important aspect is the occurrence of adverse drug reactions subjecting patients to hospitalization possibly ending up fatal. Therefore, the pharmaceutical industry is not only facing high financial risks but also humanitarian problems, that strains the trust-based relationship between the industry, physicians and patients.[4] Academia proved the usage of computational tools to be employable to many challenges of the health industry like costs of commercialization, drug safety issues and drug target validation.[5,6] However bioinformatics and computer-aided drug discovery are novel and complex disciplines only enabled by recent technological advancements in high-throughput methods. Hence, the health industry does not yet benefit from promises like personalized medicine or computer-aided identification of drug targets on a large scale.

Recent advancements in high-throughput methods and automated microscopy gave rise to the development of high-content imaging of small compound perturbations inflicted on biological systems. This method can be used to screen a wide range of different pharmacological compounds and produce cellular images, hence the method is sometimes

referred to as cell-painting.[7] Cell-painting assays capture compound perturbed biological systems and resolve morphological characteristics in an automated way that can be interpreted by computational models.[1]

The raw images from high-content imaging are processed, mostly by the software CellProfiler[6] which extracts up to 1800 morphological features per compound ranging from nucleus shape to mRNA expression. The features from the cell-painting assay can be interpreted as a morphological fingerprint, unique for each compound.[7] By now, many different cell-painting assays have been generated and specific strategies have been developed for specific purposes. A widely used cell-painting assay was developed by Bray et al.[8] The images were recorded with sixfold fluorescence staining for imaging five crucial cellular organelles.[1]

A recent publication[9] used the cell-painting assay of Bray et al.[1] to link morphological states to mechanisms of action via gene expression data. Hierarchical clustering was used to find clusters of compounds, addressing the same set of genes and therefore the same biochemical pathway. The results obtained from this cell-painting based approach mirrored the findings in the literature, which proved that cell-painting data is directly correlated to cellular pathways responding to compound perturbations. Nassiri and McCall[10] used a different cell-painting assay[11] and compound perturbed gene expression data in the context of machine learning methods. They used a LASSO model to predict cell morphological features against similar gene expression profiles. In-depth analysis of the results revealed, that model predictiveness becomes especially strong among compounds that steer gene expression in the same direction, suggesting common mechanisms of action. Hence, not only can cell-painting data be linked to mechanisms of action but also an in-depth analysis reveals linkage between compounds' mechanisms of action based on machine learning model performance. Furthermore, Rohban et al.[12] could validate and predict the connectivity of disease-associated genes using human cDNA constructs from various sources[13,14] and said cell-painting assay. The genes were manually annotated

to their known pathways, hence biased information was incorporated. Nevertheless, a single inexpensive morphological experiment could verify known connections between 110 of 220 tested genes. Moreover, a novel connection between cancer-related genetical factors could be uncovered with the aid of subpopulation visualization. Lapins and Spjuth[15] used cell-painting data to predict mechanisms of action and drug targets. In their work, they obtained mechanisms of action and drug targets from LINCS Canvas Browser[16] and the Drug Repurposing Hub.[17] Random forest classifiers were used to predict if a compound was addressing a certain mechanism of action or drug target and for some mechanisms of action predictive power could be shown. Simm et al.[7] studied a cell-painting assay specifically designed for glucocorticoid receptor nuclear translocation. However, they repurposed the data and used different machine learning models to predict certain compounds' protein targets that were completely unrelated to glucocorticoid receptor nuclear translocation whatsoever. Finally, they could prove high predictivity for 34 out of 600 assays (AUC $>$ 0.9) and enriched their protein target hit rate by 250 times in some cases proving informational abundance in cellular, morphological fingerprints.

3 Scientific Aim

The scientific aim of this project is to illuminate the connection between the clinical endpoints presented by the PubChem assays their predictive capability and the input arrays used, i.e. CP data, ECFP and a combination of both. Most problems are tackled using either Python or Bash or a combination of both. The Python library sklearn[18] is used for most machine learning applications. Further pitfalls, like data imbalance and feature selection problems, are also part of this work. synthetic minority over-sampling technique (SMOTE) is used in this work as a measure to cope with imbalanced data sets, i.e. the unequal distribution of active and inactive compounds within an assay. Many different approaches for selecting subsets of features and therefore reducing the dimensionality of the data are tested here. principal component analysis (PCA), hierarchical clustering in combination with random forest feature importance as well as minimal-redundancy-maximal-relevance criterion (MRMR) is used.

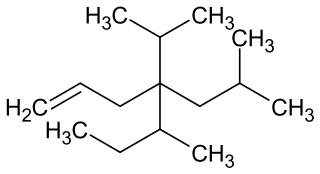
Eventually, the goal is to generate heuristics that simplify working with CP data. Generally, CP can be used to predict compound wise biochemical readouts. However, which types of readouts work well and why remains elusive. Therefore, this work aims at understanding the results obtained from RFC prediction and to link the results to cellular mechanisms and concepts.

4 Theoretical Background

4.1 Simplified Molecular Input Line Entry Specification - SMILES

simplified molecular input line entry specification (SMILES) refer to a certain formalism to generate identifiers for chemical compounds that are suited for chemists as well as for computational input. The identifier, in this case, is deduced from a two-dimensional graph of the chemical structure. The result ~~from the chemical structure~~ is a series of characters, that contain mostly alphanumeric symbols, brackets and some other symbols. The selection of those symbols follows a certain order and a certain set of rules. The set of rules ~~can be split six fold:~~ atoms, bonds, branches, cyclic structures, disconnected structures and aromaticity. SMILES is also able to process stereochemical information however, that is not mandatory since the initial approach to SMILES covers solely two-dimensional information.[19] Atoms are labelled by their letter within the periodic table of the elements. All elements of a SMILES string are written in brackets with the exceptions of the organic subset, i.e. B, C, N, O, P, S, F, Cl, Br, and I. Hydrogen atoms have further specifications. They can either appear implicitly with members of the organic subset. In that case remainder of the lowest normal valence is filled with hydrogen atoms. For example [C] refers to CH_4. Explicit notation of Hydrogen atoms occurs when they are attached to an element that is not part of the organic subset. Given a metal M, the nomenclature of four Hydrogens attached to that metal is [MH4]. Hydrogen can also be mentioned on its own

in brackets [H]. Charges are represented with a plus or minus with their respective count inside a bracket.[19] Bonds within the SMILES nomenclature are omitted if they are either aromatic or single covalent bonds. Double bonds are represented with '=' and triple bonds are represented by '#'. Ionic bonds are not specifically denoted by the SMILES algorithm. An ion pair is written as two disconnected structures with formal charges to them. Tautomeric bonds are not explicitly denoted either. One of the possible structures is translated into the SMILES string be it the enol or keto variation.[19]
Branches are depcited in parenthesis. 5-methyl-4-(2-methylpropyl)-4-(propan-2-yl)hept-1-ene is an example of nested branching using nested parenthesis. The result can be seen in figure 4.1.



5-methyl-4-(2-methylpropyl)-4-(propan-2-yl)hept-1-ene

CC(C)CC(CC=C)(C(C)C)C(C)CC

Figure 4.1: The chemical structure that examplifies branching is shown on the left side. To its right the molecular name and its SMILES string are shown. The SMILES string constitutes parenthesis that imply branching and nested branching.[19]

Cyclic structures are written linearly by breaking a single or aromatic bond within the cycles. Next, the broken bonds are arbitrarily labelled by writing the formerly connecting elements right in front of a number that is assigned to the broken bond. An illustrating example can be seen in figure 4.2.

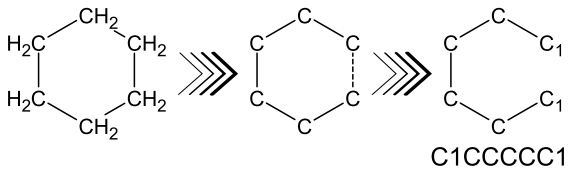


Figure 4.2: Cyclohexane as an example for a SMILES string generated from a cyclic structure. First, the explicit hydrogens are exchanged for implicit ones and the ring is linearized by breaking a bond conceptually, which is implied by the dashed bond. The Carbons where the bond was are being labelled and the resulting SMILES string is shown below the right-hand structure.[19]

A single atom can be part of multiple rings, which is then accounted for by using two or three single digits in sequence. For structures that have more than 10 rings, however, double digits are separated with a prefacing per cent sign. Also, a single digit can be reused for multiple broken bonds without creating ambiguity. A SMILES string is read from left to right and a ring closes on the first repetition of a respective digit. Cubane is an example that has multiple rings. In figure 4.3 the generation of a SMILES string is shown with the usage of the digit '1'.[19]

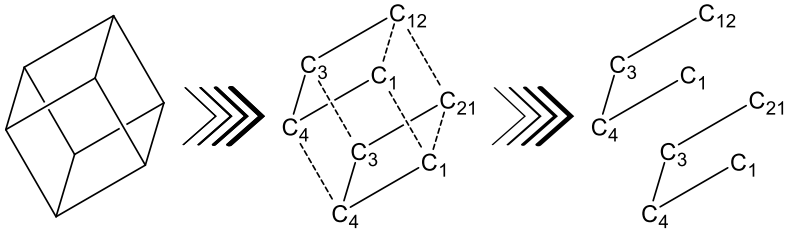


Figure 4.3: Cubane as an example of a structure that has multiple cycles. On the left, the structure is shown without explicit hydrogen atoms. In the middle picture, the bonds that are artificially broken to linearize the molecule for the SMILES string are shown in dashes. On the very right, the skeleton structure resembles the SMILES string, that is written below the molecular representations.[19]

One digit can be reused for multiple broken bonds without creating ambiguity. A SMILES

string is read from left to right and a ring closes on the first repetition of a respective digit. Disconnected structures are written as individual SMILES strings separated by a comma.[19]

Aromaticity is denoted by writing the atoms that are part of an aromatic cycle in lower case letters. Aromaticity is detected by applying an extended definition of H"uckel's rule. Another noteworthy convention is the treatment of aromatic nitrogen atoms. A nitrogen atom that is embraced by two aromatic bonds has no valency left per default. However, for aromatic nitrogen that is connected to a hydrogen atom, the Hydrogen atom is specified as shown in figure 4.4.[19]

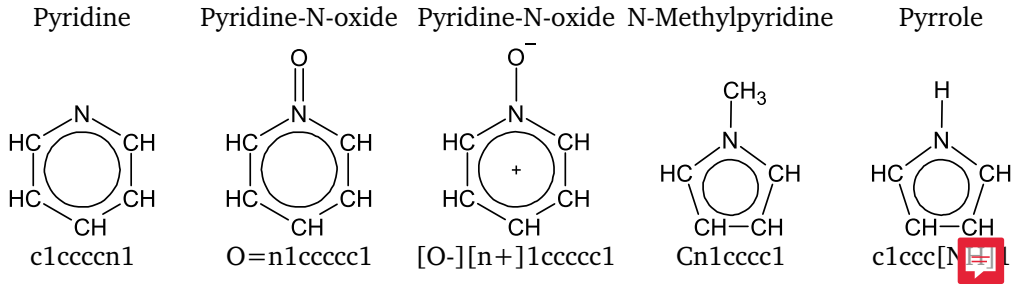
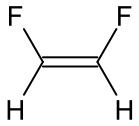


Figure 4.4: Different instances of aromatic nitrogen ~~are shown here~~. Notice that the SMILES string of pyrrole contains an additional Hydrogen that preceded the aromatic nitrogen. The aromaticity of an atom is denoted by writing it in lower case letters.[19]

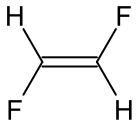
Furthermore, the SMILES algorithm introduces a convention for labelling double bond configurations and chirality. The double bond configuration is indicated by placing '/' or '\' between the atom constituting the double bond and their subsequent bonding partners. The indicators can be understood as a single bond type that gives information about their relative orientation. An example for (Z)-1,2-difluoroethene and (E)-1,2-difluoroethene is given in figure 4.5.[20]

(Z)-1,2-Difluoroethylen



F\C=C/F
F/C=C\F

(E)-1,2-Difluoroethylen

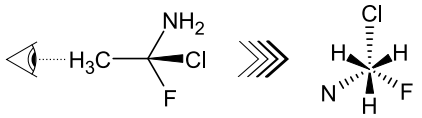


F\C=C\F
F/C=C/F

Figure 4.5: Example of double bond configuration in SMILES notation. On the left (Z)-1,2-difluoroethene can be seen and on the right is (E)-1,2-difluoroethene with their SMILES notation.[20]

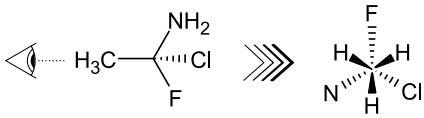
Chirality is assigned not only to chiral tetrahedral centres but also to any other chiral centre, e.g. allene-like or square planar centres. Herein, the SMILES notation is explained in the context of tetrahedral chirality centres, which is the simplest instance of chirality in organic chemistry. A chiral centre can not be the terminal node in a molecular representation, since a terminal node is either only connected to hydrogen atoms if any at all. With that in mind, the convention for tetrahedral chirality is most easily explained by investigating an example, which is (1S)-1-chloro-1-fluoroethan-1-amine which can be viewed in figure 4.6. The whole molecule is viewed along the CC-bond. Necessarily, the SMILES string contains the binding partners of the central C-atom in a certain order. This sequence can either correspond to the clockwise or anticlockwise order of binding partners when the molecule is viewed along the CC axis. Should the order be anticlockwise, an '@' is inserted after the central C-atom in brackets. The '@' is a visual mnemonic since it depicts an anticlockwise rotation around the central circle. For a clockwise order, '@@' is used instead of a single '@'.

(1S)-1-chloro-1-fluoroethan-1-amine



C[C@@](Cl)(F)N or C[C@](F)(Cl)N

(1R)-1-chloro-1-fluoroethan-1-amine



C[C@](Cl)(F)N or C[C@@](F)(Cl)N

Figure 4.6: Example of enantiomer SMILES strings.[20] Both molecules are pictures in the same way. Above each depiction is the name of the chemical formula followed by a schematic. The eye indicates the point of view along the CC axis. The resulting view of the structure is shown on the right of each subfigure. Written below are two equally adequate SMILES strings for each structure.

4.2 Canonical SMILES

SMILES strings, in general, do not claim to be unique identifiers. There are many equivalently feasible options to generate a SMILES string for a given structure. Since computational biochemistry in nowadays research accesses structures from many different sources and databases the need for a unique identifier is evident. The SMILES notation was developed with this objective in mind. So-called canonical SMILES strings fulfil this objective. It is based on the same set of rules that are described in section 4.1. The algorithm can be partitioned into two parts. The CANON part and the GENES part. The CANON part labels the atoms of the molecular structure canonically, i.e. a unique way that is based on the structural topology. The GENES part generates the unique SMILES from the aforementioned rule set (see section 4.1) and the canonical labelling.[21]
For finding a unique way of labelling molecular atoms, invariant structural properties are necessary. Thus, six properties are considered atomic invariants. Those would be the (1) number of connections, (2) the number of non-hydrogen bonds, (3) the atomic number, (4) the sign of the charge, (5) the absolute charge and (6) the number of attached hydrogens. The important attribute of those properties is that they do not change if the

order of the structural graph changes, i.e. if a different atom is picked as the starting node. The numbers in the parenthesis in front of each property correspond to a defined prioritization. From the so-called atomic properties, a so-called individual invariant can be generated for each atom of a compound by simply combining the properties into one number. Other atomic properties like isotopic mass and local chirality can be added if these six properties are not sufficient to discriminate all distinguishable nodes from each other. It is noteworthy, that some nodes are symmetrical and therefore indistinguishable.[21]

After assigning every atom an individual variant, those are compared among all constituting atoms and eventually ranked. The lowest combined variant gets rank 1 and the highest variant gets the highest rank. Furthermore, the final rank is not only depending on the individual atomic properties but also dependent on the topology. Therefore, the nodes (atoms) rank is replaced by its corresponding prime and then it is multiplied by the corresponding primes (starting from 2) of all its neighbours. Afterwards, ranks are assigned again starting from one and the procedure is repeated iteratively until the combined invariant is not changing anymore. Should there be constitutionally symmetric nodes present in the molecular graph, it becomes necessary to break ties since the symmetric groups make it impossible to find a ranking that offers a completely ordered set of nodes which is necessary for finding a canonical SMILES representation. For tie-breaking, all ranks are doubled and the first instance of the symmetric node is decremented by one. The resulting node ranking is considered a new invariant set that goes through the aforementioned iterative process of corresponding prime multiplication until it is no longer changing. After every rank is of the combined invariant is unique and not changing upon further iteration the uniquely ordered ranking has been accopmlished.[21]

THE GENES part of the algorithm can utilize the uniquely ordered ranking to decide which node it should start at and which nodes to prioritize at branching points. As an entry point for the generation of canonical SMILES, the node with the lowest ranking is chosen. Branching decisions are undertaken in the same fashion, i.e. the branching option with the lowest rank is chosen and followed through until a dead end is met. A

special rule applies when branching into a ring with a double or triple bond. To avoid the ring closure at any multi-bond the algorithm will always branch towards the multi-bond. Another solved problem occurs with cyclic and polycyclic molecules. Since the algorithm is working sequentially, it places parenthesis at branching points, which will not be closed for a cyclic system. Also, the digits that are assigned to nodes that have multiple ring closures need to be ordered in a specific way to yield a truly unique representation. And last but not least, the digits, in general, must be in the order of ring-opening nodes. Conclusively, a unique SMILES string can be assigned by first generating a unique invariant rank for every node that incorporates invariant atomic properties as well as topological information and then using these ranks as decision indicators for branching and cycles.[21]

4.3 Extended-Connectivity Fingerprints

The ECFP is a structural fingerprint that was developed to capture molecular features relevant to molecular activity.[22] They are also commonly referred to as Morgan-fingerprints since their development is partially based on the Morgan-algorithm.[23] The algorithm starts very similar to the CANON algorithm in section 4.2. However, the ECFP-algorithm introduced partial changes to the original Morgan-algorithm due to its different aim.[22]
The first change is that intermediate results, i.e. atom identifiers that are not associated with the final set, are not discarded. They are added to a set (or a list) that defines the ECFP and will be referred to as ECFP-set.[22]
As the second change, the algorithm is not stopped after the uniqueness of the identifier is achieved. Rather, the ECFP-algorithm terminates after a predetermined number of iterations specified by the user which is referred to as radius.[22]
The third change impacts recoding. Instead of rigorous canonicalization to maximize disambiguation the ECFP-algorithm uses a hashing algorithm that creates identifiers that are comparable across molecules, which would be an unfavourable trait for canonicaliza-

tion of molecules since it introduces redundancies.[22]

The Procedure of the algorithm can be distinguished into two parts: the initialization (or zeroeth iteration) and the iteration process. For initialization, all atoms in the molecule are given an identifier that is computed from the six atomic invariants, which were introduced by the canonical SMILES algorithm.[21] However a seventh invariant is taken into account as well: whether the atom is part of at least one ring structure. The atomic invariants are numerical and to obtain one integer from these properties, they are fed into a hashing algorithm that returns a 32-bit integer. Any functional hashing algorithm that maps the input onto a 32-bit integer is a sensible choice. The initial identifiers are also referred to as core identifiers since the ECFPs are circular fingerprints. Every step of the iteration procedure can be conceptually understood as a core centred substructure whose size around its core atom increases with every iteration.[22]

The iteration procedure is identical in every step. For every core, a temporary array (or list) of integers is computed. The entries in this temporary array comprise the iteration number and atomic identifier of the core atom as the first entry. Further entries comprise the bond order and atomic identifier of every neighbouring atom within one bond length in the first iteration, two bond lengths in the second iteration and so forth. This temporary array encodes for a substructure that is centred around the respective atom. Afterwards, the temporary array is inputted into a hashing algorithm that maps the atomic array onto a 32-bit integer. The identifier of the respective atom gets updated to said 32-bit integer, which is taken as input for the next substructure of increased size. Furthermore, the newly computed identifier gets appended to the ECFP-set, likewise all the new identifiers for all other atoms.[22]

The iteration process may continue as long as specified by the user, however, given a big enough structure and a sizeable radius duplicate identifiers can occur within the structure. Those duplicate identifiers are not appended to the ECFP-set. For example, a methyl group, which would be portrayed by its hashed 32-bit integer, represent a duplicate substructure, that gets appended to the ECFP-set only once.[22]

4.4 Cell-Painting Assay

Cell-Painting (CP) refers to a high-throughput screening method that generates cellular image data from fluorescence microscopy experiments. A CP assay consists of several consecutive steps which result in tabulated raw image data. ~~Aforementioned~~ steps, ~~which will be described in further detail below~~, consist of cell culture, treatment, staining and fixation, automated image acquisition and feature extraction.[1]

U-2 OS cells were used as the target organism in the cell painting assay. For every well within a 384-well clear bottom plate, 1500-2000 Cells were seeded and incubated at 37 °C for 24 hours.[24]

Thereafter, compounds were added to the cell in quadruplicates of varying concentrations. In total 30409 different compounds were ~~added individually to the different wells~~. ~~This step was followed by another~~ 48 hours of incubation time. The compounds used can be categorized as small molecules and are either taken from the Molecular Libraries Small Molecule Repository (MLSMR), the known bioactive compounds database of the Broad Institute, the Molecular Libraries Program (MLP) or compounds derived from diversity-oriented synthesis. Antibodies, enzymes and other biotherapeutics were not used in this bioassay.[1]

In total, six different fluorescent dyes were used to stain 5 different cell-organelles. Only two of the dyes were applied to the living cell culture, the remaining four were applied after fixation of the cells. Two of the dyes are used to stain the F-actin cytoskeleton, plasma membrane and Golgi apparatus. Another dye is used to stain the nucleoli and the cytoplasmatic RNA. Additionally, one stain ~~each~~ is used to stain the Endoplasmatic Reticulum (ER), the nucleus and the mitochondria. The six dyes are listed in 4.1 together with the cell organelles and the catalogue number they respond to.

Table 4.1: The fluorescent dyes used in the CP assay are listed here. The list contains the names of the fluorescent dyes, the cell organelle(s) that they are targeting and the catalogue number that refers to the Invitrogen catalogue.[24]

Fluorescent dye	Cell Organelle	Catalogue Number
Mitotracker Deep Red	Mitochondria	M22426
Wheat Germ Agglutinin	F-actin cytoskeleton, plasma membrane, Golgi	W11262
Concanavalin A	ER	C11252
Phalloidin	F-actin cytoskeleton, Plasma Membrane, Golgi	A12381
Hoechst 33342	Nucleus	H3570
SYTO 14 green fluorescent nucleic acid stain	Cytoplasmatic RNA Nucleoli	S7576

After the compound treatment ~~and the related incubation time~~, a staining solution of Mitotracker and wheat germ agglutinin (WGA) is added and incubated for 30 min at 37 °C. Afterwards, cells are fixed using paraformaldehyde. Afterwards, staining solutions from Phalloidin, Hoechst 33342, SYTO 14 and Concanavalin are ~~prepared and~~ applied to the cell containing wells and incubated for 30 min. Finally, the plates are thermally sealed and stored at 4 °C.[8 24]

In the next step, images are generated via automatic fluorescence microscopy. Five fluorescence channels are used that correspond to the different organelles in question. The channels are labelled DNA, RNA, AGP (F-actin cytoskeleton, Golgi and plasma membrane), Mito (mitochondrial) and ER (Endoplasmatic Reticulum).[1] After the automatic image acquisition is completed, the so-called CellProfiler[3 25] software generates numerical features from said images. CellProfiler has its standard pipeline that contains steps

to generate cellular features from fluorescence images. The concepts of this pipeline are visualized in figure 4.7. First, the images are being aligned, cropped and an illumination correction is applied. ~~Next up is the~~ cell identification step. CellProfiler first identifies nuclei by searching for bright, well-dispersed and non-confluent so-called primary objects. Another important step within this recognition is to identify clumped primary objects, then finding their dividing lines and removing these objects or merging them depending on their measurements.[3] Taking the nuclei as a starting point, the secondary objects, like cell edges, the cytoplasm and nuclear membrane are identified next. After the cells ~~are~~ identified, CellProfiler conducts different measurements to calculate a variety of features connected to cellular compartments and organelles. These features include ~~the~~ area, shape, texture and other more complex features.[3] The dataset that is used in this work constitutes 1768 features that have a variance greater than 0. This is important to consider since features that remain constant for every compound do not contain any information. After the feature extraction via CellProfiler, the ~~now~~ finalized data set is called raw image data.

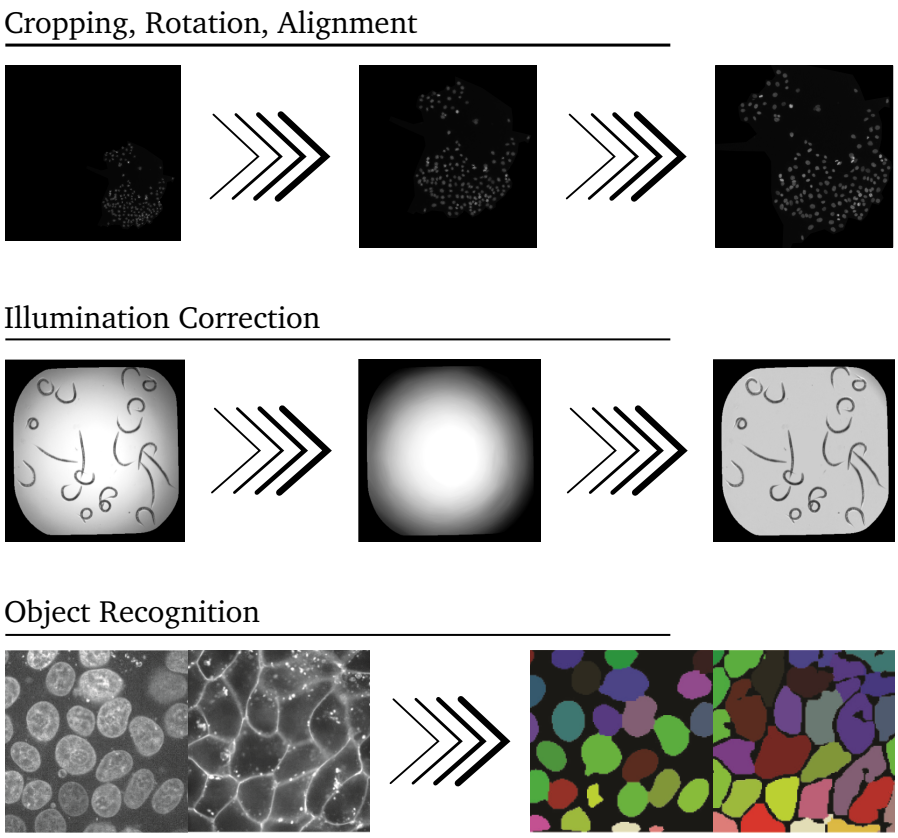


Figure 4.7: The workflow CellProfiler follows is visualized conceptually. The first ~~step~~ shows fluorescent microscopy images of human cells. These are cropped and rotated to yield a better view of the cells.[26][27] In the second step the illumination correction function is applied to a generic image.[27] In the last step (Object Recognition), the identification of stained nuclei and membranes of human pluripotent stem cells are shown.[28]

4.5 Raw Image Data

Before explaining the preprocessing steps it is cardinal to understand the content of the raw image data that was obtained in the aforementioned feature extraction by CellProfiler. The raw image data is a very large spreadsheet with features making up the columns and ~~th~~e rows that correspond to the well from which the original images were taken. Additionally, the rows can be identified by the compounds that have been used to treat the respective wells. ~~However, some wells have not been treated at all as control samples and~~ every compound is measured in quadruplicates as a minimum. Also, some are measured in octuplicates as well as in different concentrations. Compounds whose features have been extracted for only one concentration are referred to as single-concentration-compounds and the compounds that appear in multiple concentration are ~~henceforth~~ called matthews correlation coefficients.

The spread sheet does not only contain numerical features extracted from CellProfiler but so called meta data, too. Meta data refers to ~~data that explains~~ methodological information which was used during the experimental procedure. Hence, ~~the~~ compound concentration, ~~the~~ plate number, ~~the~~ plate map number and ~~many more~~ information are being categorized as meta data. Also, whether the row corresponds to a treated cell sample or to a mock, is stored within the first 17 columns before the actual CellProfiler features start to be listed from column 18 to 1801. ~~Hence, there are 17 columns containing meta data~~ from these 17 only five are important for the suceeding steps. These five columns contain information about the plate on which the sample was placed and about whether or not the sample was treated with a compound solvated by dimethyl sulfoxide (DMSO) or plain DMSO, referred to as mock. The remaining three columns contain ~~further~~ information about the molecular structure of the respective compound as a SMILES string (see section 4.1), its concentration, and an identifier assigned by the Broad Institute. In table 4.2 the column header names ~~can be found~~ together with a brief description. The names in table 4.2 correspond to the ones from the original raw image

data file.[1]

Table 4.2: Below, the names of the most important meta data column headers are listed verbatim from the source file. For every column header, a description is supplied.

Column Name	Description
Metadata_Plate	Contains the plate number of respective well
Metadata_ASSAY_WELL_ROLE	States if the well was treated with a compound or just with DMSO
CPD_SMILES	Contains the structural information as a SMILES string
Metadata_mmoles_per_liter	States if a compound was applied this column specifies the concentration
Metadata_broad_sample	Identifier assigned by Broad Institute that varies either with compound, concentration or plate number

4.6 PubChem-Assay

The database that supplies the targets for this project is the PubChem database.[29] The PubChem database contains ~~a variety of~~ information about chemical compounds and their bioactivities. The bioassays in PubChem are assigned a unique assay identifier (AID) and they consist of descriptive information and the ~~testing results of the~~ corresponding read-out. The descriptive part contains, among others, information like the name and theoretical background and relevance to the experiment. ~~Also,~~ the data source and descriptions

of the readout.[30]

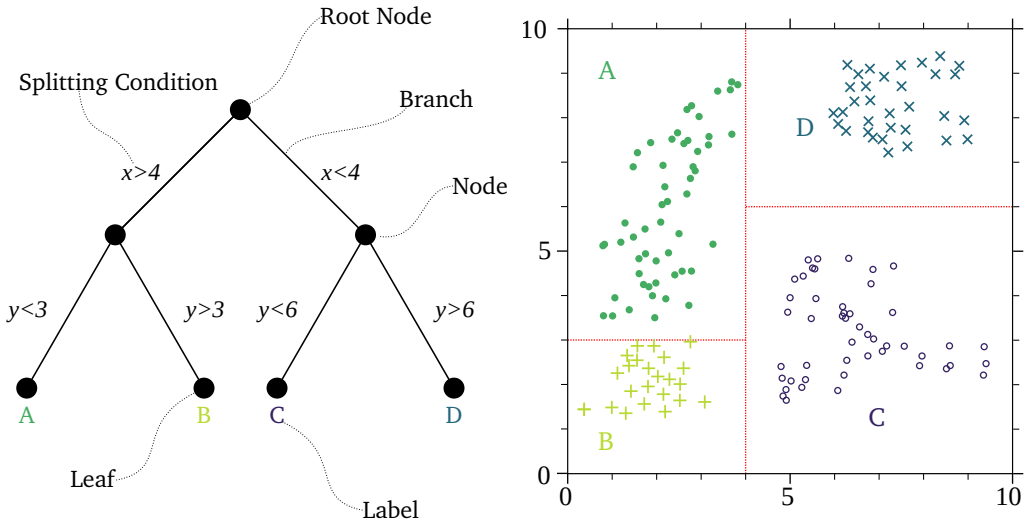
The results of a bioassay can be very diverse since the experimental procedure depends on the specific assays which are numerous. In general, the depositor of a bioassay can provide as many detailed results as necessary.[31] However, PubChem requires the depositor to ~~record~~ a summary result for each chemical sample. This summary result constitutes the 'bioactivity score' and the 'bioactivity outcome'. The bioactivity outcome can ~~take~~ five mutually exclusive values: 'chemical probe', 'active', 'inactive', 'unspecified' and 'inconclusive'. The rationale behind the bioactivity outcome is usually provided in the assay comment section to enable a detailed interpretation of the results by the users.[30]

4.7 SMOTE

SMOTE can be used to overcome problems associated with imbalanced data sets. The method uses the given data as input and creates synthetic samples from that data. The process is most easily described for a two-dimensional data set. For every point in that data set, the k nearest neighbours are found. New data points are then generated on the connecting lines in between the central point and its neighbours. The total number of new data points generated between the central point and its surrounding neighbours depends on the sampling strategy, i.e. how many data points the total semi-synthetic data set is supposed to have. The distance at which the synthetic points are inserted is chosen at random.[32]

4.8 Random Forests

In machine learning (ML) classification denotes a (mathematical) rule that produces a label y from a set of input features X. One way of classification utilizes a decision tree. A decision tree is a common mathematical in stochastics. In ML the nodes are used as entry points for data to be classified and the paths after each node are called branches. The final nodes at which the classification process ends are called leaves. ~~From~~ starting from the root node, the decision tree splits the complete data set, D, into subsets, D_l and D_r, funnelling them to the left or right branch from the root node. The information given by the feature x_i guides the splitting of the dataset at each node. Moving further down the branches the subsets are split further into smaller and smaller subsets until they reach the leaves. The leaves correspond to labels present in the data and the complete subset that reached the leaves down from the root node is assigned the label at the final leaf. A certain numerical rule, the splitting condition, defines how to split the data set and each node splits the entering data set on one certain feature. However, since the decision tree is sequential, the information from earlier nodes are always part of the decision process. Therefore, the final decision for the label at the leaves incorporates multidimensional data in a quite simple fashion. A decision tree can be visualized in the context of its data since it applies geometrical decision boundaries (see figure 4.8).[33]



(a) Example of a decision tree with description of the individual elements.

(b) Data that is classified based on the examplary decision tree.

Figure 4.8: A decision tree and the geometrical representation of the classification process is shown on a simple 2-D data set that shows 4 different classes with the labels A, B, C and D. The data virtually enters the decision tree at the root node. The splitting condition defines if the subsets get funnelled into the right or left branch. From the Second nodes the other splitting criteria are applied and eventually the data reaches the leaves where assigning the labels happens.[33]

From the described procedure three cardinal algorithmic questions arise: How to choose the feature on which to split the data? How to choose the splitting condition? And how to assign class labels to the leaves?[33]

The simplest of these questions is the first one. The features of a decision tree choose at random. This is not the only way of generating a decision tree but the most common. Interestingly, carefully choosing the features that are taken into account by different nodes does not add great value to the classification process.[33] However, if X contains many features that do not contribute to the classification problem (e.g. by having a variance

close or equal to zero) that approach can backfire.[33]
The second question about the splitting condition can be solved by applying information theory ~~to it~~. There are several methods ~~on how~~ to determine a splitting condition, ~~nevertheless~~, the information gain method is described here as an example. First ~~of all~~, the splitting conditions are determined during the training phase or fitting of a decision tree. During training, all true labels z of the data set are known to the algorithm. Therefore a good heuristic is to gain as much information as possible for a given split. Information gain refers to the enrichment of datapoints with common labels within each subset. The entropy of a subset (e.g. left subset) D_l after splitting $H(D_l)$ is given by the sum over all different class labels c of the sum of the product of the relative frequency of the given class label within the subset D_l and its binary logarithm. This relative frequency is obtained by dividing the number of data points of class c, $n(c; D_l)$, by the total number of data points in the subset $N(D_l)$.[33]

$$H(D_l) = \sum_c \frac{n(c; D_l)}{N(D_l)} \log_2 \left(\frac{n(c; D_l)}{N(D_l)} \right) \tag{4.1}$$

(4.2)

$H(D)$ corresponds to the bits that are necessary to classify a data point in the parent data set. Thus, $H(D_l)$ and $H(D_\mathrm{r})$ are the bits required to encode the labels within the left and right branch subsets. The information gain is defined as weighted entropies of the split subsets subtracted from the entropy of the parent data set. Herein, the entropy of the split subsets $H(D_l)$ is weighted by the probability to find items in the corresponding pool (w_l or w_r).[33]

$$w_\mathrm{r} = \frac{N(D_\mathrm{r})}{N(D)} \qquad w_l = \frac{N(D_l)}{N(D)} \tag{4.3}$$

Finally, the information gain I of a certain node is given by ~~the following~~ equation:[33]

$$I = H(D) - w_{\mathrm{r}} \cdot H(D_{\mathrm{r}}) - w_l \cdot H(D_l) \tag{4.4}$$

In general, the better the information gain, the better the split. From here, it is straightforward to obtain the optimal splitting condition. The inputs X of the data set D have a certain number of features f (usually corresponding to columns) and datapoints d (usually corresponding to rows). Like aforementioned, the node is assigned a certain feature f at random. Within f, there are d data points which means there are $d-1$ possible splits that would change the composition of D_l and D_{r}. Hence, the information gain is computed for every $d-1$ possible splits and the threshold resulting in the best information gain is kept as a parameter for that node.[33]

Finally, the last question can receive its answer. The leaves that are reached after splitting the data set numerous times on different features with different threshold need to assign the data points in the final subset ~~a label~~. Since there are lots of random choices involved within the decision tree classifier, a random forest is simply the generation of a bunch o trees and letting them vote for labels of a data set.[33]

Aside from the splitting conditions, the feature selection other parameters dictate the behaviour of a decision tree and therefore of the RFC. Those parameters are referred to as hyperparameters. These hyperparameters control how many trees are included in a RFC, how deep the branches go, how many features they are allowed to use and many more.[18] Opposing to the splitting condition that is chosen by mathematical optimization, the hyperparameters have to be inputted by the operator.[33]

4.9 Cross Validation and Splitting

Usually, an advanced ML model has enough parameters to fit a given data set perfectly, therefore 'memorizing' the inputs and their corresponding labels. If a dataset that was

used for training in its entirety was also used to test the resulting model, the performance of the model would be near perfect. However, the performance on samples outside the training set would be very poorly represented, since the model would not be able to abstract from the dataset. To solve the overestimation in performance ~~one would encounter this way~~, cross-validation (CV) is introduced.[34]

One solution to this problem is to split the given data set up into training and test set, which is also referred to as ~~a~~ validation set. However, given random splitting of the data, the classifier will not be best possible because too ~~less~~ data ~~was~~ used during ~~the~~ training. The issue of overfitting (or selection bias) and insufficient use of the data set can be circumvented by CV.[33]

During CV the data set is split in k subsets of equal ~~lengths~~. These subsets are referred to as folds. Next, the model iterates through k cycles of training and evaluation. Every cycle, another fold is used as ~~a~~ validation set, whilst the remaining $k-1$ folds are used for training. The evaluations of each cycle are then averaged to yield a better estimate of the true model performance. This method of splitting the data set into k folds and then cycling through each combination is called k-fold CV[34]

Nevertheless, there is yet another pitfall, that results in an overestimation of model performance. In section 4.8 hyperparameters were mentioned, which are user-inputted parameters that dictate the architecture of the model (i.e. the number of decision trees, branching depth, etc.). Those parameters are usually optimized by applying an automatic sampling of different values for each parameter. Thus, the choice of hyperparameters itself contains information about the samples that the model is trained on, otherwise, their choice would make no difference in prediction performance. Optimizing the hyperparameters and using K-fold cross validation (KFCV) only on the RFC parameter optimization therefore still reports exaggerated evaluation metrics. The application of KFCV to the hyperparameter optimization as well as to the training of the RFC solves this issue and is called nested KFCV.[34] The ~~splitting~~ strategy that is chosen to split the data set should be as unbiased as possible to diminish performance overestimation. However, the distribution of labels within each fold should be comparable to avoid underestimation

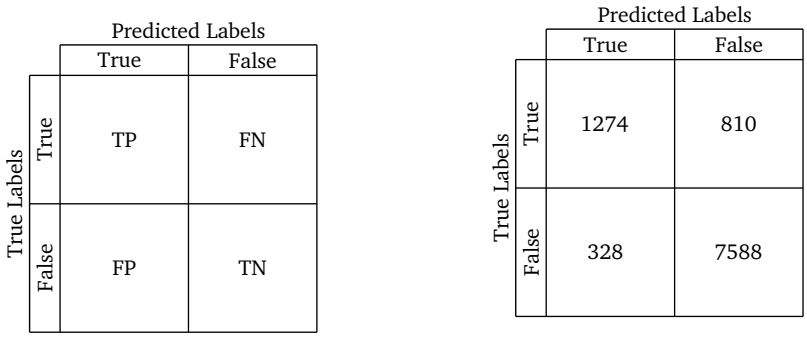
of the performance. In the worst case, one fold could contain all labels, which would lead to very poor prediction performance. The random stratified split strategy splits the data set into subsets that are randomly chosen ~~however~~ each of them exhibits an equal label distribution, which counteracts overpessimistic prediction performance.[35]

4.10 Performance Evaluation

To evaluate the performance of a RFC there are several different metrics available. Since this work is concerned with a binary classification problem (i.e. only two labels are possible per sample) the evaluation metrics will concern binary classification as well. The most fundamental performance assessment of a classifier is given by the confusion matrix which is simply comparing the predicted labels with the true labels. From the confusion matrix more applied metrics can be calculated, namely the true positive rate (TPR), the true negative rate (TNR), the balanced accuracy, the matthews correlation coefficient. Furthermore, the receiver operating characteristic curve (ROC-curve) and area under the ROC curve (AUC-ROC) supply further information about the goodness of the fitted model.[36]

4.10.1 Confusion Matrix

A confusion matrix compares the predicted labels with the true labels of the classification problem. The confusion matrix is a quadratic matrix of the order of the number of different classes. For a binary classification problem, the confusion matrix is therefore a two by two matrix. On the diagonal of the matrix the correctly identified instances are shown, either true positive (TP) or true negative (TN) instances. Off diagonal~~,~~ erroneously identified instances are presented, either false positive (FP) or false negative (FN) values. The general structure of a confusion matrix is shown in figure 4.9.



(a) Structure of a confusion matrix

(b) Confusion matrix with exemplary values.

Figure 4.9: The general structure of a binary confusion matrix is shown on the left and on the right exemplary values are inserted in the respective fields.

4.10.2 TPR, TNR, Balanced Accuracy and Matthews Correlation Coefficient

From the confusion matrix several other metrics can be computet. For example the TPR and TNR, more widely known as the sensitivity and the specfificity, aswell as the balanced accuracy (BA) and the matthews correlation coefficient (MCC). The TPR describes the frequency of correctly positive labeled samples within the predictions whereas the false positive rate (FPR) depicts the frequency of incorrectly positive labeled predictions.[36]

$$TPR = \frac{TP}{TP + FP} \tag{4.5}$$

$$FPR = \frac{FP}{TP + FP} \tag{4.6}$$

Analogously, the TNR describes the frequency of the correctly negative labeled samples within the predictions.[36]

$$TNR = \frac{TN}{TN + FN} \tag{4.7}$$

The BA is simply the average ~~between~~ the TPR and the TNR.[37]

$$BA = \frac{TPR + TNR}{2} \tag{4.8}$$

The MCC is a metric that scores high only if all entries within the confusion matrix perform well and it is sensitive to class imbalances.[38]

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.9}$$

The MCC can score between -1 and 1 where 1 depicts a very good performance whilst -1 describes a poor fit.

4.10.3 ROC and AUC-ROC

RFCs assign discrete prediction labels z to their input samples. However, as stated in section 4.8 a majority voting of the total number of trees is conducted. The majority voting can be expressend ~~in~~ as probability p for a certain class label to which a threshold t can be applied.

$$z = \begin{cases} 0 & \text{if } p < t \\ 1 & \text{if } p \geq t \end{cases} \tag{4.10}$$

Per default, the threshold is 0.5. However, the threshold can vary from 0 to 1. For a threshold of zero, all labels will be positive, since no sample will have a total vote smaller than zero. For a threshold of one, however, all predictions will be negative. Hence, for each threshold, a different confusion matrix is obtained and therefore a different TPR and FPR. For the ROC-curve the TPR is plotted on the y-axis and the FPR is plotted on the x-axis. The aforementioned threshold varies from 0 to 1 to obtain a set of different confusion matrices from which the TPR and FPR is calculated and plotted. A perfect ROC-curve has a TPR of 1 and a TNR of 0. The AUC-ROC corresponds to the goodness

of the model depicted by the corresponding ROC-curve an AUC-ROC of one is a perfect score since it corresponds to the aforementioned perfect ROC-curve. The benefit of the ROC-curve and AUC-ROC is that it contains more information than the other metrics and enable quick visual confirmation.[36]

4.11 Feature Importance

There are three different methods of choice for feature importance measuring that are used in this project: PCA, random forest feature importance and the MRMR. PCA can be understood as a method of dimensionality reduction. The original data is mapped to a new coordinate system that corresponds to the maximum variation exhibited by the data.[39] Therefore, the number of new axes which are referred to as principal components have the same number of dimension as the original data. However, the first few principal components usually account for a high part of the variance within the given data set, whereas the lowest principal components account for no variance at all, which is referred to as noise in data science.[33] Furthermore, the contribution of each original feature to the major principal components can be deduced and used to infer feature importance for the original data set.[39]
Random forest feature importance can be either measured by gini impurity or by entropy (or information gain, see 4.8). The gini impurity is defined as the product of probabilities to encounter positive samples, p_1 and negative samples p_0 respectively at a given node k.[40]

$$G_k = 2p_1p_0 \tag{4.11}$$

From equation (4.11) follows, that a small gini impurity refers to a very succesful split from the parent node with the index $k-1$. The best split of a node is achieved when the descendant nodes contain a subset that comprises one label only. The feature importance

for a single decision tree $I_{\mathrm{d}}^{(f)}$ is therefore defined as the sum of reductions in gini impurity from every parent node k that uses feature f in its splitting condition to its descendants $k+1$.[40]

$$I_{\mathrm{d}}^{(f)} = \sum_{k} G_{k}^{(f)} - G_{k+1} \tag{4.12}$$

Notice, that G_{k+1} comprises the contribution of the left and right descendant nodes. The overall feature importance of feature f is calculated as the the average of tree wise feature importances.[40]

$$I^{(f)} = \frac{1}{N_d} \sum_{d} I_{\mathrm{d}}^{(f)} \tag{4.13}$$

N_d depicts the number of trees in the RFC. Since the splitting at node k always incorporates the information from prior nodes this method of feature importance accounts for non-linear feature interaction. One shortcoming of this method is, that features that are strongly correlated tend to be split their importance between each other, which results in two difficulties. The first one is, that very important features get assigned lower values since the decision trees of the RFC uses other instances from a cluster of highly correlated features instead of just one. The second is, that the final selection of features will contain very many features that belong to the same highly correlated cluster. Adding many features that contain similar information has no beneficial effect on the model and fosters overfitting based on the information of a few clusters. To overcome these two pitfalls, hierarchical clustering can be performed on the features beforehand and from every cluster, only one feature is picked for further feature engineering.

MRMR is a feature selection algorithm that was proposed by Peng et al.[41] It utilizes mutual information between features to calculate their redundancy and mutual information between features and each class label to calculate maximum relevance to the categorization problem. The algorithm then optimizes the subtraction of the features redundancy and the feature relevance to obtain a scoring for each feature. MRMR was found to be very suitable for data sets with more than 1000 numerical features.[41]

5 Methods

In the following sections, the computational process is described. The implementation including the data sets will be available at https://github.com/Foly93/masterthesis. For programming either Python or Bash was used. So called jupyter notebooks are used as user interface for python programming. Furthermore python scripts and bash scripts were used as well.

5.1 Preprocessing

As mentioned in section 4.5 the raw image data contains meta and data columns. The metadata columns dictate the decision-making process during the preprocessing and the data columns themselves are the subject of preprocessing. The data columns or features vectors are the inputs for machine learning applications described in the following chapters.
In brief, the preprocessing combines the bioassay data set and the raw image data into 52 fully annotated and machine learning ready data sets. Eventually, they contain information about the features, about the endpoint and some metadata information, e.g. for compound identification.
During the preprocessing the Metadata_broad_sample turned out to be a suboptimal

identifier because it is not unique for every compound. Different Metadata_broad_sample values can be used for the same compound if it corresponds to a different compound concentration or plate. Therefore, the Metadata_broad_sample was exchanged for the SMILES string.

First, the individual 384-well plates were centred on the mock samples. For that purpose, the plate-wise average of the untreated samples was calculated for every morphological feature and then subtracted from the treated samples. The next step is to calculate the feature-wise medians of concentrations of the same compounds. However, some compounds were measured in different concentrations. Therefore, a new metadata column was introduced that labelled each row either as a single-concentration-compound or a matthews correlation coefficient. The single-concentration-compounds medians could be computed in a straight forward fashion, which was done for the whole raw image data set. The resulting preprocessed raw image data frame has 31692 rows and 1768 feature vectors. The rows contain median features for each single-concentration-compound and the unprocessed features of the matthews correlation coefficients.

For combining the individual assay data frames (see section 4.5) with the preprocessed raw image data the two data tables are first merged on the Metadata_broad_sample identifier. Hence, only compounds that exist in both data frames are present. The preprocessed raw image data's metadata features SMILES and canonical SMILES which is unique for every compound (see 4.2), thus compounds are properly identified, by now. From here, the matthews correlation coefficients that are present in the combined data frame is inspected since they require further consideration. Since the machine learning approach requires one morphological profile per compound, as well as one label per compound, the decision was to calculate the median of the concentration that was measured most often.

This procedure is repeated for all 52 bioassay data frames. The preprocessed raw image data only needs to be done once, however, every bioassay data frame is then merged with the preprocessed raw image data and consecutively the matthews correlation coefficients' medians are computed. This computational process results in 52 combined

ML-ready data set with 1768 relevant features and varying row number since the compound wise overlap varies from assay to assay. A table of the resulting 52 assays with the number of active, inactive and total compounds can be found in table 5.1.

Table 5.1: This table gives an overview over the combined ML-ready data sets obtained from the preprocessing procedure. The AID from the original PubChem data set is given and the number of active, inactive and total compounds. All 52 assays combined have 371978 inactive labels and 12140 active lables.

AID	Inactives	Actives	Total	AID	Inactives	Actives	Total
1030	4804	832	5636	588334	6978	133	7111
1458	6547	487	7034	588458	8850	117	8967
1529	7794	150	7944	588852	8840	128	8968
1531	7818	122	7940	588855	7536	151	7687
1578	7816	146	7962	602340	21297	102	21399
1688	6814	158	6972	624202	8342	237	8579
1822	7822	141	7963	624256	8574	139	8713
2098	7719	132	7851	624296	6475	439	6914
2156	7868	149	8017	624297	7440	252	7692
2216	7328	154	7482	624466	8844	173	9017
2330	1752	131	1883	651610	18234	218	18452
2540	8015	127	8142	651635	8036	125	8161
2553	7908	109	8017	651658	18839	163	19002
2599	7913	229	8142	651744	297	207	504
2642	7821	196	8017	720504	8197	341	8538
2796	7837	345	8182	720532	1164	185	1349
485270	7992	190	8182	720582	8933	121	9054
485313	7497	491	7988	720635	248	126	374
485314	7589	172	7761	720648	8928	126	9054
504333	6598	526	7124	743012	315	195	510
504444	5296	275	5571	743014	315	188	503
504466	6909	260	7169	743015	320	211	531
504582	8022	110	8132	777	2831	911	3742
504652	7829	312	8141	894	4769	324	5093
504660	8094	131	8225	932	6399	420	6819
504847	9047	175	9222	938	2528	158	2686

5.2 Inputs

As a baseline model for the prediction of the bioassay endpoints present in the PubChem ECFPs are chosen (see 4.3). Therefore ECFPs are calculated from the canonical SMILES utilizing the Chem package from the RDkit python library for all of the 52 PubChem bioassays.[42] The obtained compound identifiers are 2048-bit vectors with radius 2. Every bit in this vector can be considered a feature and the whole of it is used as input for the machine learning algorithm.[42]
Furthermore, the preprocessed CP descriptors are used as input. In the first, modelling cycle, predictions for structural and morphological fingerprints are generated individually. Afterwards, a feature importance analysis is performed to find the most important features in each of the two descriptors. The feature engineered descriptors are then combined and predictions are generated once more.

5.3 Targets

For creating annotations for the input vectors the PubChem bioassay database is queried. PubChem comprises more than 1 200 000 bioassays.[43]The amount of information stored at the PubChem database is so vast that the process of finding data sets fitting for this project becomes a problem in itself. The filtering process conducted to find bioassays that measure endpoints relevant to this work is depicted below.
First, the 11 biggest folders are downloaded from the PubChem database, each containing up to 1000 bioassays.[44] Then the 100 assays with the most compounds are kept from each of the eleven folders resulting in 1100 bioassay data sets with noticeable size. The next step is to find assays with an endpoint that might be related to toxicity or cell morphology. For that purpose, two auxiliary files are generated. The first file contains detailed information about each of the 1100 assays. That includes the AID, the assay

name and a description of the assay and the endpoint tested. The second file is a list of protein targets, which are enriched for cytotoxic and cytostatic phenotypes generated by Mervin et al.[2] A program searches the assay information file for instances from the protein targets list and saves the AIDs that are related to said targets to another list. The resulting list of supposedly cytotoxic compounds consists of 671 assays. For the next step, the compound overlap with the raw image data needs to be found (see section 4.5). However, the compounds in the data set are annotated with their PubChem assigned compound identifier (CID) which is not a widely used identifier. Therefore a more general identifier needs to be generated for each compound that can be used to screen against the CP data set. The PubChem website offers functionality that generates a description for a given CID or a list of them. Part of that description is the international chemical identifier key (InChI-key), which is a much more general, unique identifier that can be translated into other identifiers like SMILES strings. Therefore the next step is to concatenate all compounds into a list that is then uploaded onto the PubChem website. The description of the CIDs is then downloaded. The CIDs in the 671 bioassays are then exchanged for the InChI-keys. The compound overlap with the CP is conducted concerning the $\mathrm{Metadata\_broad\_sample}$ (which turned out to be suboptimal in section 5.1 and needed to be corrected for). Therefore the compounds of each assay are merged with the InChI-key annotations of the CP data set and only entries present in both data sets are kept. Next the InChI-keys are exchanged for their $\mathrm{Metadata\_broad\_sample}$ identifiers. Not all 671 assays are used for further investigation. Like already mentioned, PubChem labels their compounds 'active', 'inactive' 'unspecified' or 'inconclusive'. If a dataset contains no actives or too less, machine learning applications will have trouble to categorize the data correctly since the two classes (active and inactive class) are too unbalanced. Thus, in the next step, the threshold of at least 100 active compounds is applied as a filter, resulting in 52 bioassays. From these 52 bioassays, 'inconclusive' and 'unspecified' rated compounds are deleted. Notice that 100 active compounds can be a comparably small amount of actives since some of the 52 final bioassays have more than 20 000 compounds. Conclusively, 52 spreadsheets are obtained, containing the metadata broad sample, as a

molecular identifier and the PubChem activity outcome as a label for the later prediction.

5.4 Prediction

For each endpoint represented by a PubChem assay, an RFC was developed and trained. Three different modelling cycles can be distinguished. The first cycle was solely concerned with the CP combined PubChem assays, the second cycle was concerned with the ECFPs, whilst the last cycle used the feature engineered combined set of descriptors for prediction.
Nested 5-Fold CV was used to train the model and tune the hyperparameters with a stratified split strategy. For the inner loop that fit the parameter of the RFC, a random split strategy was used also with 5-fold CV. Before splitting the data in the inner loop, SMOTE is applied to increase the minority class label by 100% effectively doubling its size and random undersampling is applied right after. Thereby, the minority class amounts to 75 % compared to the majority class label.
The hyperparameters are optimized using a halving random search method from sklearn. [45] [18]
For that purpose a random grid was used. The parameters which were covered can be seen in table 5.2.

Table 5.2: Hyperparameters covered by the RFC

Hyperparameter	Values Covered
max_depth	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
max_features	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
min_samples_leaf	5, 6, 7, 8, 9, 10, 11, 12, 13
min_samples_split	4, 5, 6, 7, 8, 9, 10, 11, 12, 13
n_estimators	100, 200, 300, 400, 500
bootstrap	False, True
oob_score	False
criterion	gini, entropy
class_weight	None, balanced

From this grid 500 parameter sets were sampled for each outer CV loop and only one best estimator is returned and evaluated. For all five best estimators from every outer fold the BA, MCC, TPR, TNR, ROC-curve and AUC-ROC are calculated.
This procedure is conducted for every PubChem assay mentioned before (see table 5.1) in combination with the CP descriptors, the ECFPs and finally with the feature engineered combination of the two.

5.5 Feature Engineering

The feature engineering is performed using three distinct methods further described in section 4.11. First of all the PCA is performed using the PCA method available from scikit learn.[18] The method is applied to the CP-PubChem data sets to find the features that comprise most of the variance. The 100 features that explain the most of the variance in the first principal components are added to the list of most important features.

The next step is to pick important features by using a RFC algorithm with gini impurity. However, before the gini impurity feature importance can be applied, redundancy of similar features need to be reduced. For that reason, features are clustered based on their Spearman correlation with all other features. The resulting clusters are cut-off in a way that at most 400 clusters remain. From each cluster, one feature is picked at random. The resulting 400 features are used to filter the original data set which is then funnelled into the random forest-based feature selection algorithm. This RFC uses 250 estimators from which the features are scored using the gini impurity (see equation (4.13)).[46]
The last method is MRMR which was used to extract the thirty most important features based on a maximum-relevance-minimum-redundancy criterion. The computational python implementation from Peng et al.[41] was used (https://pypi.org/project/pymrmr/).
Since the ECFPs are binary features (either 0 or 1) spearman-clustering, and MRMR will not work. Therefore, only the random forest feature importance of sci-kit learn was used to score the structural fingerprint features. Nonetheless, instead of only using the gini impurity, the entropy-based feature selection was utilized as well. This results in 200 most important features from the 2048 features present in the original fingerprint data set for each of the 52 PubChem bioassays.[46]
In the next step, the most important CP and ECFP features are combined into one set of features for each of the 52 assays. First, the CP features are combined into a list and duplicate features are removed. The same is done for the ECFP features. Finally the complete set of features and labels enter the RFC described in section 5.4.

6 Results and Discussion

For every PubChem assay, predictions were performed by using the CP descriptors first, then the ECFP descriptors and eventually by using the combined feature engineered descriptors. In this chapter, first, the absolute performances of the ECFPs and the CP predictions are given. Afterwards, the performance metrics are subject to a detailed comparison. Furthermore, from the information obtained from the features engineering, conclusion are drawn about why some PubChem assays perform better and why some are performing worse and if there can be found rules on which data sets CP might be most useful to be applied.

6.1 Comparative Analysis of ECFP and CP Predictions

For better visualisation, the predictions for the bioassays (identified by their AID) are sorted by the AUC-ROC of the CP-only prediction run. Hence the order of the following figures (figure 6.1, figure 6.2, etc) is the same.
The CP prediction run yields AUC-ROCs from around 0.4 to 0.8. They outperform the ECFP in 12 bioassays. The BA and the MCC show the same trend. The PubChem assays with the AIDs 720532, 651635, 624297, 2330, 743014, 588855, 743012, 624296, 743015, 1578, 504444, 651744 outperform the ECFP based prediction in all evaluatin

metrics (see figure 6.1. These PubChem assays are referred to as high performing assays whilst the remaining 40 bioassays are referred to as low performing assays. Notice that high and low performing refers to the prediction that only uses CP descriptors.

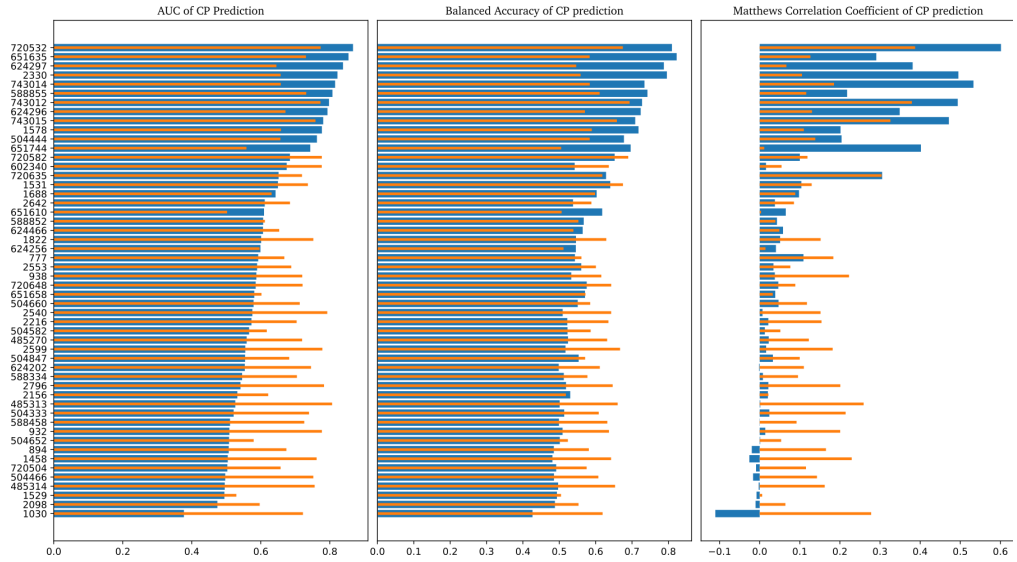


Figure 6.1: Comparison of the absolute AUC-ROCs of the CP and ECFP Predictions.

In figure 6.1 no evaluation metric from the combined features prediction is shown. When comparing the CP and ECFP with the combined features' evaluation, information about the shortcomings and strengths of each identifier can be gained. Looking at the AUC-ROC comparison in figure 6.2 it is evident and was expected, that the feature engineered predictions perform generally better than the CP features only. However, it was not expected that the fingerprints still outperform the combined feature space by a significant margin which can be seen in the middle panel in figure 6.2. Also, the combined feature space does not necessarily improve the AUC-ROC of the high performing assays significantly. Every improvement in the high performing assays is less than 5 % and some prediction performances for AIDs 651744, 504444, 1578 and 624297 get worse (see figure 6.2 right

panel).

Even though the ECFPs outperform the combined features, the combined features outperform the CP-only prediction in almost every low performing assays. In some cases up to 20 %. This shows, that the engineered ECFP features do contain information that helps to mitigate the shortcoming of the CP descriptors to some extend. The right panel in figure 6.2 shows that the engineered features add new information that helps to classify the low performing assays notably, but the high performing assays too to some extend. On the other hand, the middle panel in figure 6.2 shows that the information is not sufficient to outperform the ECFPs in the low performing assays even though structural as well as morphological information are present. The fact that the ECFP features outperform the combined features concerning the AUC-ROC leads to the assumption that the feature engineering of the ECFPs needs to be optimized and more features need to be included in the final model.

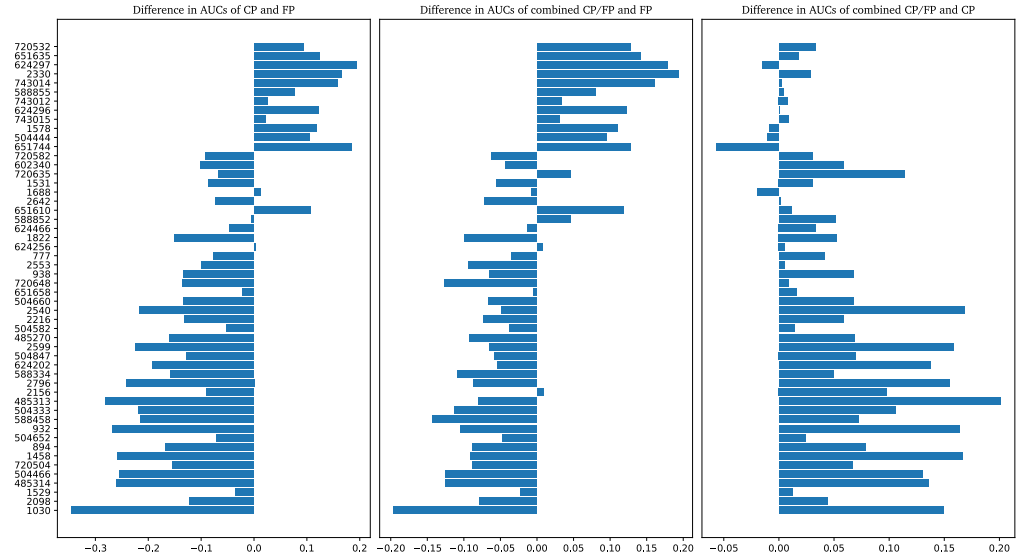


Figure 6.2: Difference in AUC between the CP, ECFP and the combined prediction run

The results shown by the comparison of BA in figure 6.3 support the conclusion presented so far, albeit there are some new information that need to be discussed in detail. In the left panel of figure 6.3 it can be seen that the difference in BA of the low performing assays is not as significant as the AUC-ROC in figure 6.2. This trend is continued in the both the other panels, showing a smaller increas of BA when combined fingerprints are in use. Therefore, the middle panel the ECFPs only perform around 5 % to 10 % better on low performing assays compared to the combined features, compared to approximately 15 % concerning the AUC-ROC.

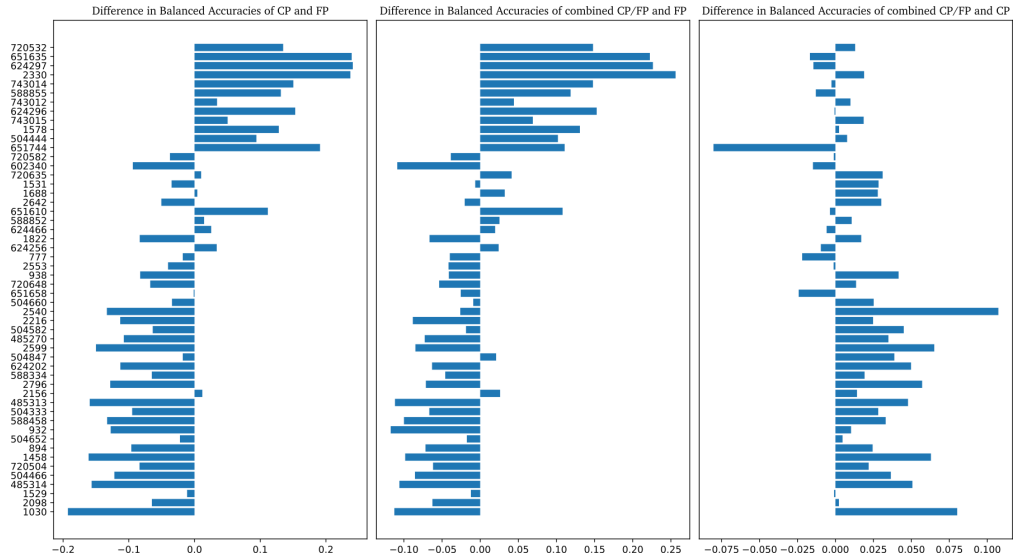


Figure 6.3: Difference in balanced accuracy between the CP, ECFP and the combined prediction run

The MCC measure the prediction performance of imbalanced data sets more reliably compared to the AUC-ROC and BA. In figure 6.4 the same trend can be observed that was already described above which is indicating that the advanced sampling by using SMOTE mitigates the class imbalance succesfully.

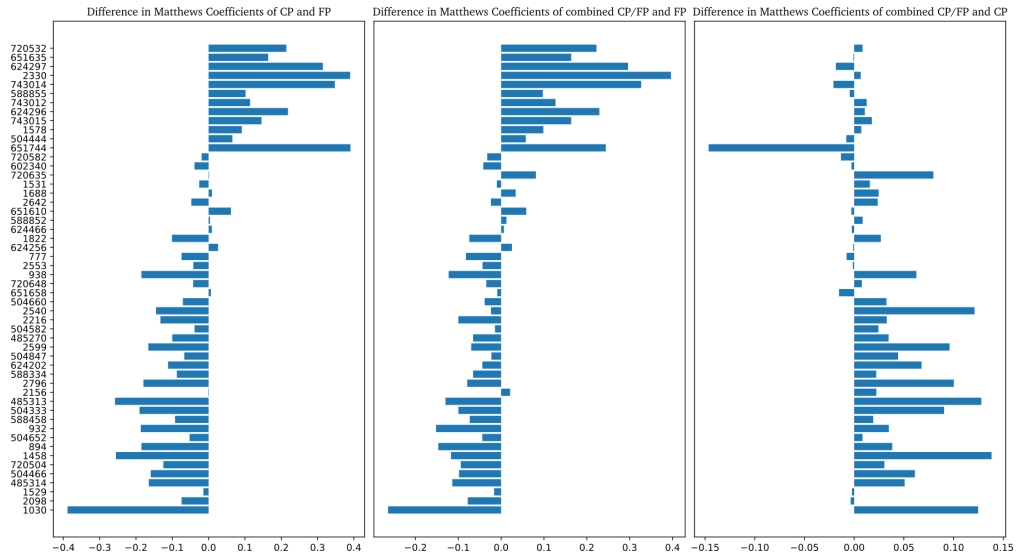


Figure 6.4: Difference in matthews correlation coefficient between the CP, ECFP and the combined prediction run

When it comes to TPR and TNR of the different models the behaviour looks differently. As can be seen in the left panel of figure 6.5 many low performing assays show a greater TPR when CP features are used as descriptors compared to ECFP features. In the middle pannel the difference in TPR is almost non-existant compared to the left panel and the right panel clearly shows, that the inclusion of the feature engineering has an alternating effect on the performance of the different PubChem assays. In some cases the TPR increases and in some cases the TPR decreases. The prior two panels lead to the assumption, that the inclusion or exclusion of ECFPs does not matter a lot for the TPR, which would mean that the CP descriptors in general and their feature engineering is mostly responsible for the differences in TPR in the right hand panel. On average, the high performing assays have 27 % higher TPR when CP is compared to ECFPs. Comparing combined descriptors to ECFPs they score 27 % higher on average as well. However when comparing the scoring of CP and combined descriptors within high performing assays the

combined features score 0.4 % lower on average. Within the low performing assays CP and combined features compare equally bad against the ECFP features (on average 7 % and 8 % respectively). However the combined features features score on average 0.5 % worse compared to CP dscriptors. The meaning behind this could be that ECFP dscriptors do not contribute significantly to the TPR.

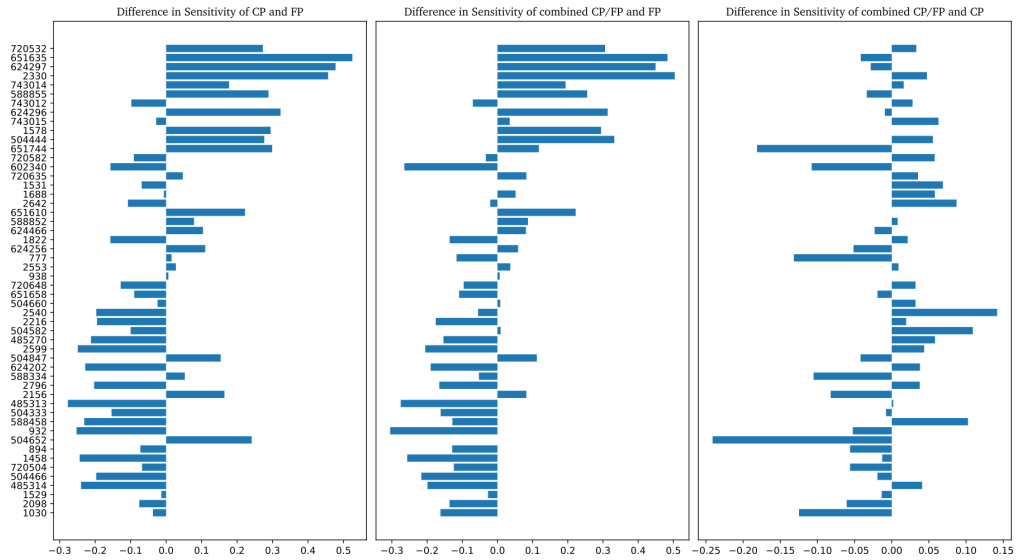


Figure 6.5: Difference in TPR between the CP, ECFP and the combined prediction run

When it comes to TNR or specificity, the ECFP seem to make a more significant contribution to the overall performance. As can be seen in the left panel of figure 6.6 the high performing assays are not as clearly on top of the ECFP. Comparing CP and ECFP descriptors on average the high performing assays exhibits 2.5 % higher TNR which is vanishingly small compared to the difference in TPR or other metrics discussed before. The combined features perform 2 % compared to the ECFP and 0.5 % worse compared to the CP descriptors when focussing on the high performing assays.
Looking at the low performing assays the CP perform 6 % worse than the ECFP on av-

erage and the combined features perform 0.6 % worse than the ECFP and 6 % better than the CP descriptors. Notably, the difference in TNR performance between CP and ECFP compares very well to the difference between CP and combined fingerprints, which means that the fingerprints should contain information that enriches the TNR but only for the low performing assays. The high performing assays however are not significantly influenced by the addition of ECFP which can be seen in the TNR difference between the CP and combined fingerprints (0.5 %). Furthermore, one could conclude that the high performing assays must contain readouts that are especially accessible by the information contained by the CP data.

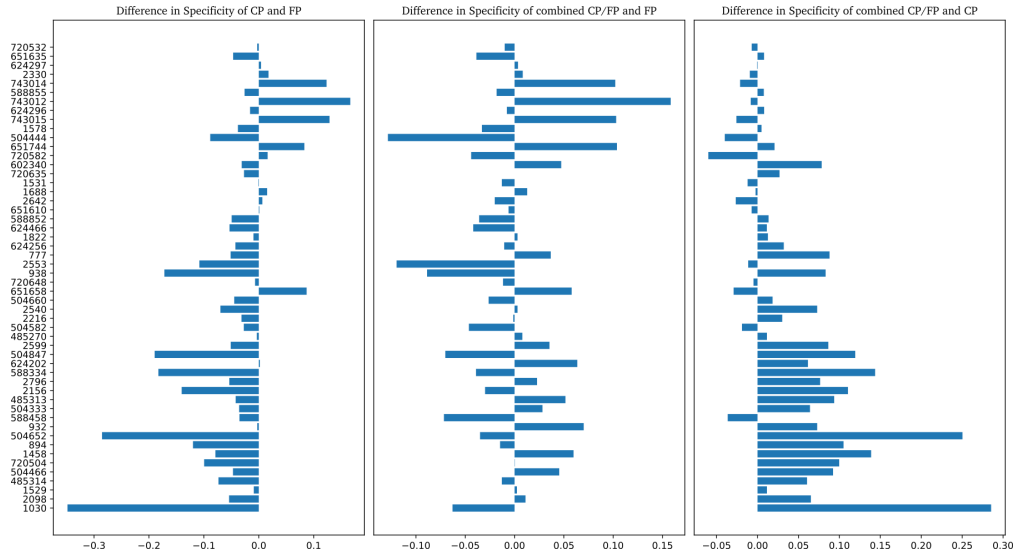


Figure 6.6: Difference in TNR between the CP, ECFP and the combined prediction run

In table 6.1 the average difference between the possible descriptor combinations can be seen for the high performing assays and low performing assays. The obvious trends are better performance of CP in high performing assays and ECFP performing better in low performing assays (hence the names). The same can be said for the comparison between

the combined feature space and the ECFP with the difference that the combined features perform significantly better against ECFP within the low performing assays. In the high performing assays CP only performs generally better than the combined features and in the low performing assays the combined perform better than CP. Exceptions from these general trends can be especially seen for TPR and TNR like described above.

Table 6.1: Average evaluation metrics sorted by high performing assays and low performing assays. CP denote the prediction run that used the cell-painting descriptors, FP denotes the run with the structural fingerprints and SF the combined selected features. The listed values are the differences in the respective evaluation metric.

	high performing assays			low performing assays		
Metric	CP vs FP	SF vs FP	SF vs CP	CP vs FP	SF vs FP	SF vs CP
AUC	0.1255	0.1167	0.0012	-0.1337	-0.0611	0.0726
BA	0.1488	0.1440	-0.0048	0.0662	-0.0412	0.0250
MCC	0.2132	0.2019	-0.0113	0.0926	-0.0546	0.0380
TPR	0.2721	0.2678	-0.004	0.0713	-0.0763	-0.0051
TNR	0.0255	0.0202	-0.0053	0.0613	-0.0061	0.0552

6.2 Evaluations from Feature Engineering for Low and High Performing PubChem Assays

The feature engineering of the CP data might be able to further illuminate why some assays are better predictable with CP descriptores and others are not. For clarification reasons, in this section the notation of high performing assays and low performing assays introduced in section 6.1 is still in use.
Experimentally the features in the CP data set are obtained by fluorescence microscopy.

The staining and image generation process described in section 4.4 utilizes six fluorescent dyes and thereby measures five difference fluorescence channels corresponding to five different cellular organelles or compartments, namely DNA, RNA, Mito, ER and AGP (see table 4.1).
If a given assay exhibits good prediction performance unusual activity within those channels should be noticeable and therefore features belonging to certain channels should have higher importance to them compared. Depending on the cellular process that corresponds to the assay a certain channel could be of utmost importance. For example, a genotoxic assay should have higher contributions from the RNA and DNA channels and in turn lower contributions from the remaining channels.
On the other hand, if an assay is not well predictable by CP descriptors the importance of distinct features and channels is supposed to be less enriched and in the worst case features are picked at random and are therefore uniformly represented.
Conclusively, the normalized standard deviation of each channel should be high within the high performing assays and low within the low performing assays. For that purpose the features related to each channel are counted, their frequencies are calculated by dividing the total number of channel counts present in the most important feature of the corresponding assay. Afterwards, the frequencies are normalized per channel for easier inter-comparability. The normalization was performed using the standard scaler from scikit-learn's preprocessing library.[18] This scaler transforms each channel to an average of zero and unit standard deviation. Afterwards, every value is multiplied by 100 for easier visual interpretation. The resulting average standard deviation for high performing assays and low performing assays is given for each channel in table 6.2. There it can be seen, that the DNA, RNA, Mito and ER channels exhibit a ratio bigger than one, which corresponds to more channel-wise variance within that assay group.

Table 6.2: Assay-normalized standard deviation per channel for the low performing assays and high performing assays. In the last row the ratio of the two is shown as well. A ratio greater 1 means, that the channel is enriched in the high performing assays compared to the low performing assays which is the case for DNA, RNA, Mito and ER.

Assay Group	DNA-std	RNA-std	AGP-std	Mito-std	ER-std
low	72.826	94.684	103.124	82.02	99.702
high	164.78	115.08	95.022	139.055	109.517
ratio	2.263	1.215	0.921	1.695	1.098

According to abovementioned rationale, a higher ratio between low performing assays and high performing assays indicates that the reason for the different predictive capabilities of the high performing assays and low performing assays is rooted in the CP information. Furthermore, the different averaged standard deviations lead to a direct connection between cell morphology and the predictive capabilities of the respective assay.

6.3 In Depth Analysis of High Performing PubChem Assays

In an attempt to categorize further annotate the PubChem assays, the descriptions of the PubChem assays were manually screened for further information. The goal of said screening was to find terms or keywords that could relate the bioassay endpoint to cellular morphology. The first screening served the purpose to find terms that are present at all in the 52 PubChem assays of choice. The terms that could be filtered out are shown in table 6.3. The presented term list does not claim to be exhaustive, but the terms refer to phenomena that are most likely visible on a cellular scale and might be modifiable by the small compounds used in the CP assay.

Table 6.3: Phenotypic terms that can be associated with individual PubChem assays. These terms were manually filtered from the descriptions of the PubChem assays available at https://pubchem.ncbi.nlm.nih.gov/.

Acronym	Associated Phenotypic Terms
Adipgen	Adipogenesis, Obesity
Ang-gen	Angiogenesis
C-Grwth	Cell Growth, Cell Viability
C-Death	Apoptosis, Cell Death
Im-Resp	Immune Response
Inflamm	Inflammation
Snscnce	Senescence
Mei/Mit	Meiosis, Mitosis
C-Stres	Xenobiotics, Toxins, Cell Stress
Signall	Signalling, Secretion, Hormones
CNS/NDD	CNS, Epilepsy, Depression, NDD
C-Entry	Invasion, Cell Entry
Mitotox	Mitotoxicity
A-Cancr	Anti-Cancer
Genome	Genome Integrity, DNA-Repair, genotoxicity
Prteom	Ubiquitinylization, Protein Regulation, Proteome influencing

In table 6.4 some PubChem with their annotations are shown. 1 is inserted if the phenotypic term is related with the corresponding PubChem assay and a 0 assigned if the term can not safely be associated. The decision-making process is intuitive to a certain account. For example, genotoxicity and cell death are very related to each other. If a large portion of the DNA is damaged the cell initiates apoptosis and therefore dies. However, the AID 2540 probes inhibitors for a protein called SENP8 that moderates the maturation

of Nedd8 which plays a crucial role in DNA-repair. Herein SENP8 is considered a modulator of DNA-Repair and not directly connected to apoptosis. Therefore AID 2540 has a 1 at 'Genome' and a 0 at 'C-Death' even though the two are hard to separate.

Most importantly the complete annotation matrix was created before the prediction performances were recorded and can therefore be considered unbiased. The idea behind this matrix is to test if some annotations are enriched for the high performing assays and low performing assays respectively. For that purpose, the complete phenotypic terms table is split into high performing assays and low performing assays. Next, the entries for each term within each sub-table are summed up to yield the abundances of each term for the assay groups high performing assays and low performing assays respectively. The partial abundances are divided by the total abundance of a phenotypic term to yield the frequencies of each assay group. Next, the frequencies are divided by the number of bioassays present in each assay group. The resulting measure is the relative term frequency per assay and described the enrichment of a phenotypic term within the corresponding assay group and is therefore dubbed phenotypic enrichment.
In table 6.5 the enrichment for the high performing assays (high) and low performing assays (low) as well as the difference of the two is shown. A negative difference means that this phenotypic term is enriched in the low performing assays and vice versa. It can be seen that phenotypic enrichment is positive for 'C-Grwth', 'C-Death', 'Im-Resp', 'Genome', 'Inflamm', 'Mei/Mit', 'C-Stres', 'C-Entry', and 'A-Cancr'. 'Im-Resp', 'Genome' and 'C-Death' are the three terms showing the highest enrichment.

Table 6.4: Phenotypic terms that can be associated with individual PubChem assays. These terms were manually filtered from the descriptions of the PubChem assays available at https://pubchem.ncbi.nlm.nih.gov/.

Phenotypic Term	1030	1458	1529	1531	1578	1688	1822	2098	2156	2216	2330
Adipgen	1	0	0	0	0	0	0	0	0	0	0
Ang-gen	0	0	1	1	0	0	1	0	0	0	0
C-Grwth	0	0	1	1	0	0	1	0	0	1	0
C-Death	0	0	1	1	1	1	1	0	0	0	1
Im-Resp	0	0	0	0	1	0	0	0	1	0	0
Prteom	0	0	0	0	0	0	0	0	0	0	0
Genome	0	0	0	0	0	0	0	0	0	0	0
Inflamm	0	0	0	0	1	0	1	0	0	0	0
Snscnce	0	0	0	0	0	0	1	0	0	0	0
Mei/Mit	0	0	0	0	0	0	1	0	0	1	0
C-Stres	1	0	0	0	0	0	0	1	0	0	0
Signall	0	0	0	0	0	0	0	0	1	0	0
CNS/NDD	0	0	0	0	0	1	0	0	1	0	0
C-Entry	0	0	0	0	0	0	0	0	0	0	0
Mitotox	0	0	0	0	0	0	0	0	0	0	0
A-Cancr	0	0	0	0	0	0	0	0	0	0	1

Table 6.5: Enrichment of phenotypic terms in high performing assays and low performing assays and the difference between the two. A high number corresponds to a higher frequency of the corresponding term within the group of assays. The difference clarifies if the relative frequency is higher or lower in the high performing assays and quantifies that enrichment in a comparable manner.

Group	Adipgen	Ang-gen	C-Grwth	C-Death	Im-Resp	Prteom	Genome	Inflamm
low	0.025	0.025	0.019	0.016	0.008	0.025	0.013	0.019
high	0.0	0.0	0.021	0.029	0.056	0.0	0.038	0.021
diff	-0.025	-0.025	0.002	0.013	0.047	-0.025	0.025	0.002
Group	Snscnce	Mei/Mit	C-Stres	Signall	CNS/NDD	C-Entry	Mitotox	A-Cancr
low	0.025	0.018	0.015	0.023	0.022	0.017	0.025	0.017
high	0.0	0.024	0.035	0.006	0.01	0.028	0.0	0.028
diff	-0.025	0.006	0.02	-0.017	-0.011	0.011	-0.025	0.011

From this analysis, it can be concluded that assays that probe endpoints related to these specified phenotypes exhibit high predictive capability with CP descriptors. The fact that genome integrity and DNA-repair scores a high value is also in agreement with the channel enrichment analysis. As can be seen in table 6.2 the DNA channel is the one that had the highest ratio among all five channels.

7 Conclusion and Outlook

This work aimed to explore the capabilities of the CP assay by Bray et. al[1] and to evaluate the prediction of said descriptors on various biochemical endpoints related to toxicity. Furthermore, new insights on how and when to apply the CP are desirable. For this purpose 52 bioassays from the PubChem database were selected and used as targets for CP, ECFP and combined descriptors.

The evaluation of an RFC showed diverse performance overall assays, 12 however were able to outperform ECFP on nearly every metric. However, the specificity and sensitivity showed different behaviour in comparison to the other reported metrics. The TPR seemed to be less influenced by the information present in ECFPs. That leads to the conclusion, that CP descriptors have a higher chance that a positively labelled compound is indeed positive. This characteristic is especially useful for toxicity prediction since the ability to correctly predict toxic (positive) compounds can prevent unnecessary testing and harm. On the other hand, the TNR seems to be very much dependant on the information stored in the ECFP. A comparison between the TNR in the high performing assays and low performing assays showed that combined features greatly increase the specificity, especially in the low performing assays. The general trend of the performances showed, that the high performing assays did not improve by a lot when combined features were in use. That leads to the conclusion that CP is in general not applicable to any bioassay.

The first approach to find heuristics by which to pre-select bioassays that might be suitable for CP descriptors was to check if any fluorescence channels from the microscopy experiment were enriched for the high performing assays and low performing assays. It was found that four out of five channels showed a higher variance within the high performing assays compared to the low performing assays. It was argued, that this indicates that the high performing assays are indeed better predictable, because their endpoints relate to cellular morphological processes.
The relative enrichment has the drawback that it can only be applied comparatively to better or worse performing data sets within a modelling problem. To mitigate this shortcoming and to further illuminate rules by which bioassays could be preselected as suitable for CP descriptors, phenotypic annotations have been manually generated. These annotations were generated unbiased and connect the PubChem bioassays to morphological changes induced by various cellular processes (e.g. signalling, proteome regulation, genotoxicity etc.). By calculating the phenotypic enrichment for high performing assays and low performing assays it was found that endpoints that relate to genome integrity, DNA-repair, genotoxicity, cell-death, apoptosis, cell stress, toxins and immune response.
Future work should consider two main problems. The first problem is that of feature engineering with CP and ECFP. The evaluation metrics showed that the combined features were regularly outperformed by ECFP-only prediction, at least within the low performing assays. This concludes that too many important features were removed during the feature engineering process, thus invaluable information was lost. The same goes for the feature engineering of CP referring to section 6.1 in some instances CP-only predictions outperformed the combined feature predictions especially in the high performing assays. It should be possible to obtain better performances for combined features for each of the 52 bioassays.
Secondly, the manual annotation of phenotypic terms showed potential when it comes to understanding why the RFC performs well and when it comes to pre-selecting assays. The list that was generated herein, is not at all comprehensive and a more comprehensive list of keywords could further the understanding of CP descriptors. Another useful approach

is to combine the channel enrichments with the annotation enrichment. For example, it would make sense if assays that are concerned with genotoxicity are enriched for the RNA and DNA channels. Assays that test ion channel inhibitors on the other hand should, in theory, be enriched for the AGP channel. It is an interesting approach, however, the annotation of phenotypic terms is not empirically possible. First of all, because of the lack of information. Secondly, phenotypic terms are ambiguous. It is left for scientific intuition if an assay is concerned with cell stress, genotoxicity, cell death or all three of them. A method that circumvents this problem is computational pathway analysis. Future work should try to either define very easily applicable phenotypic terms that are as mutually exclusive as possible or should conduct pathway analysis as a less intuitional phenotypic annotation method.

Bibliography

[1] Bray, M.-A. et al. *GigaScience* **2017**, *6*.

[2] Mervin, L. H.; Cao, Q.; Barrett, I. P.; Firth, M. A.; Murray, D.; McWilliams, L.; Haddrick, M.; Wigglesworth, M.; Engkvist, O.; Bender, A. *ACS Chemical Biology* **2016**, *11*, 3007–3023.

[3] Carpenter, A. E.; Jones, T. R.; Lamprecht, M. R.; Clarke, C.; Kang, I.; Friman, O.; Guertin, D. A.; Chang, J.; Lindquist, R. A.; Moffat, J.; Golland, P.; Sabatini, D. M. *Genome Biology* **2006**, *7*, R100.

[4] Katara, P. *Network Modeling Analysis in Health Informatics and Bioinformatics* **2013**, *2*, 225–230.

[5] Myers, S.; Baker, A. *Nature Biotechnology* **2001**, *19*, 727–730.

[6] Nelson, M. R.; Bacanu, S.-A.; Mosteller, M.; Li, L.; Bowman, C. E.; Roses, A. D.; Lai, E. H.; Ehm, M. G. *The Pharmacogenomics Journal* **2008**, *9*, 23–33.

[7] Simm, J. et al. **2017**,

[8] Bray, M.-A.; Singh, S.; Han, H.; Davis, C. T.; Borgeson, B.; Hartland, C.; Kost-Alimova, M.; Gustafsdottir, S. M.; Gibson, C. C.; Carpenter, A. E. *Nature Protocols* **2016**, *11*, 1757–1774.

[9] Gustafsdottir, S. M.; Ljosa, V.; Sokolnicki, K. L.; Wilson, J. A.; Walpita, D.; Kemp, M. M.; Seiler, K. P.; Carrel, H. A.; Golub, T. R.; Schreiber, S. L.; Clemons, P. A.; Carpenter, A. E.; Shamji, A. F. *PLoS ONE* **2013**, *8*, e80999.

[10] Nassiri, I.; McCall, M. N. *Nucleic Acids Research* **2018**, *46*, e116–e116.

[11] Wawer, M. J.; Jaramillo, D. E.; Dančík, V.; Fass, D. M.; Haggarty, S. J.; Shamji, A. F.; Wagner, B. K.; Schreiber, S. L.; Clemons, P. A. *Journal of Biomolecular Screening* **2014**, *19*, 738–748.

[12] Rohban, M. H.; Singh, S.; Wu, X.; Berthet, J. B.; Bray, M.-A.; Shrestha, Y.; Varelas, X.; Boehm, J. S.; Carpenter, A. E. *eLife* **2017**, *6*.

[13] Wiemann, P. C. e. a., S. *Nature Methods* **2016**, *13*, 191–192.

[14] Yang, X. et al. *Nature Methods* **2011**, *8*, 659–661.

[15] Lapins, M.; Spjuth, O. **2019**,

[16] Duan, Q.; Flynn, C.; Niepel, M.; Hafner, M.; Muhlich, J. L.; Fernandez, N. F.; Rouillard, A. D.; Tan, C. M.; Chen, E. Y.; Golub, T. R.; Sorger, P. K.; Subramanian, A.; Ma'ayan, A. *Nucleic Acids Research* **2014**, *42*, W449–W460.

[17] Corsello, S. M.; Bittker, J. A.; Liu, Z.; Gould, J.; McCarren, P.; Hirschman, J. E.; Johnston, S. E.; Vrcic, A.; Wong, B.; Khan, M.; Asiedu, J.; Narayan, R.; Mader, C. C.; Subramanian, A.; Golub, T. R. *Nature Medicine* **2017**, *23*, 405–408.

[18] Pedregosa, F. et al. *CoRR* **2012**, *abs/1201.0490*.

[19] Weininger, D. *Journal of Chemical Information and Modeling* **1988**, *28*, 31–36.

[20] Inc., D. C. I. S. Daylight Theory Manual. https://www.daylight.com/dayhtml/doc/theory/index.pdf, 2011; last time opened: 24.02.2021.

[21] Weininger, D.; Weininger, A.; Weininger, J. L. *Journal of Chemical Information and Modeling* **1989**, *29*, 97–101.

[22] Rogers, D.; Hahn, M. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.

[23] Morgan, H. L. *Journal of Chemical Documentation* **1965**, *5*, 107–113.

[24] Wawer, M. J. et al. *Proceedings of the National Academy of Sciences* **2014**, *111*, 10911–10916.

[25] Kamentsky, L.; Jones, T. R.; Fraser, A.; Bray, M.-A.; Logan, D. J.; Madden, K. L.; Ljosa, V.; Rueden, C.; Eliceiri, K. W.; Carpenter, A. E. *Bioinformatics* **2011**, *27*, 1179–1180.

[26] Moffat, J. et al. *Cell* **2006**, *124*, 1283–1298.

[27] Institute, B. CellProfiler example images and pipelines. https://cellprofiler.org/examples, 2020; https://cellprofiler.org/examples, Last accessed: 26.02.2021.

[28] McQuin, C.; Goodman, A.; Chernyshev, V.; Kamentsky, L.; Cimini, B. A.; Karhohs, K. W.; Doan, M.; Ding, L.; Rafelski, S. M.; Thirstrup, D.; Wiegraebe, W.; Singh, S.; Becker, T.; Caicedo, J. C.; Carpenter, A. E. *PLOS Biology* **2018**, *16*, e2005970.

[29] Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. *Nucleic Acids Research* **2015**, *44*, D1202–D1213.

[30] Wang, Y.; Bolton, E.; Dracheva, S.; Karapetyan, K.; Shoemaker, B. A.; Suzek, T. O.; Wang, J.; Xiao, J.; Zhang, J.; Bryant, S. H. *Nucleic Acids Research* **2009**, *38*, D255–D266.

[31] Wang, Y.; Xiao, J.; Suzek, T. O.; Zhang, J.; Wang, J.; Zhou, Z.; Han, L.; Karapetyan, K.; Dracheva, S.; Shoemaker, B. A.; Bolton, E.; Gindulyte, A.; Bryant, S. H. *Nucleic Acids Research* **2011**, *40*, D400–D412.

[32] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357.

[33] Forsyth, D. *Applied Machine Learning*; Springer-Verlag GmbH, 2019.

[34] Raschka, S. *CoRR* **2018**, *abs/1811.12808*.

[35] Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. San Francisco, CA, USA, 1995; p 1137–1143.

[36] Fawcett, T. *Pattern Recognition Letters* **2006**, *27*, 861–874.

[37] Kelleher, J. *Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies*; The MIT Press: Cambridge, Massachusetts, 2015.

[38] Boughorbel, S.; Jarray, F.; El-Anbari, M. *PLOS ONE* **2017**, *12*, e0177678.

[39] Jolliffe, I. T. *Principal Component Analysis*; Springer-Verlag GmbH, 2002.

[40] Cha Zhang, Y. M., Ed. *Ensemble Machine Learning*; Springer-Verlag GmbH, 2012.

[41] Peng, H.; Long, F.; Ding, C. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2005**, *27*, 1226–1238.

[42] Landrum, G. *RDKit Documentation Release 2019.09.1*; Creative Commons: o Creative Commons, 543 Howard Street, 5thFloor, San Francisco, California, 94105, USA, 2019.

[43] of Medicine, N. L. PubChem. https://pubchem.ncbi.nlm.nih.gov/, Last Access: 04.03.2021.

[44] PubChem Database. ftp.ncbi.nlm.nih.gov/pubchem/, Last Access: 05.03.2021.

[45] Bergstra, J.; Bengio, Y. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.

[46] Scikit-Learn, Feature importances with forests of trees. https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html, Last Access: 04.03.2021.