Prediction of Cytotoxicity Related PubChem Assays Using High-Content-Imaging Descriptors derived from Cell-Painting

A comparative study investigating the applicability of cell-painting data using machine learning methods and chemoinformatics tools

Master thesis by Luis Vollmers

Date of submission: March 14, 2021

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

Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

UNIVERSITY OF CAMBRIDGE

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, 14. März 2021

L. Vollmers

Contents

Prediction of Cytotoxicity Related PubChem Assays Using High-Content-Imaging Descriptors derived from Cell-Painting

Das Vorhersagepotenzial neuartiger High-Content-Imaging Datensätze, aus Cell-Painting-Assays, soll in dieser Masterthesis anhand von statistischen und praktischen Methoden überprüft werden. Dabei werden die prozessierten Rohdaten mit Datenbanken verglichen, die Informationen über toxikologische Endpunkte enthalten. Weiterhin werden verschiedene Machine Learning Algorithmen anhand der Datensätze trainiert und die Ergebnisse extensiv analysiert. Dabei wird besonderes Augenmerk auf die Unterschiede im Vorhersagepotential zwischen den einzelnen Endpunkten gelegt, um daraus Informationen über die Anwendbarkeit der Cell-Painting Datensätze zu gewinnen.

•**Applied computing**~**Physical sciences and engineering**~**Chemistry**
•Applied computing~Life and medical sciences~Computational biology ~Recognition of genes and regulatory elements
•Applied computing~Life and medical sciences~Systems biology
•Computing methodologies~Machine learning~Machine learning algorithms~Feature selection
•Computing methodologies~Machine learning~Machine learning approaches~Classification and regression trees
•Computing methodologies~Machine learning~Cross-validation

Luis Vollmers Heidelberger Str. 103 · 64285 Darmstadt
015736748013 luis.vollmers@web.de

1 Summary

The pharmaceutical industry is centered around small molecules and their effects. Apart from the curative effect, the absence of adverse or toxicological effects is cardinal. However, toxicity is at least as illusive as it is important. A simple definition is: 'toxicology is the science of adverse effects of chemicals on living organisms'.[1] However, this definition comprises several caveats. What is the organism? Where does therapeutic and adverse effects start and end? Even for the toxicity in simplest organisms, cytotoxicity, the mechanisms are mannifold and difficult to unravel. Hence, it remains obscure which characteristics a compound has to combine to be labelled as toxic.

One attempt to illuminate these characteristics are the novel cell-painting (CP) assays. 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.[2] Before the images are taken and further processed, the cells are perturbed by small compounds that might affect the cellular morphology. Therefore CP contains information about each compound formatted as a unique cell-structure anomaly. Which sub-information are actually valuable within these morphological fingerprints remains elusive and therefore a significant part of this project is dedicated to exploring the CP data and their predictive capabilities comparatively. They will be compared among bioassays and to different descriptors, too. The CP data used by this project contains roughly 30 000 compounds and 1800 features.[3]

In chemistry, the structure determines the function of a compound or substance. Therefore, apart from CP, structural fingerprints are used as a benchmark to compare the information content of CP against. In this project extended-connectivity fingerprints (ECFPs) were used to encode the compounds' structures into numerical features.

This work is concerned with morphological changes that correspond to toxicity. Thus, the CP data were combined with toxicological endpoints from a selection of PubChem assays. The selection process implemented a minimum number of active compounds, a size criterion and the occurence of toxicologically relevant targets.[4]

After the selected assays were combined with each of their descriptors, machine learning models were trained and their predicitive power was evaluated against certain metrics. The pre-

dictions can be separated into 3 cycles. In the first cycle, the CP data are used as descriptors, the second cycle used the structural fingerprints and the last cycle used a subset of both. The subsets were selected by a rigorous feature engineering process.
The evaluation of the prediction metrics illuminates which strengths and shortcomings the morphological fingerprints have compared to the structural fingerprints. It turned out that there are two groups of assays: those PubChem assays that are generally better predicted with CP features and those that have higher predictive potential when using ECFP. Additionally, it was uncovered that ECFP comprise higher specificity compared to CP data which shows higher sensitivity on the other hand. A high sensitivity means the prediction rarely mislabels a sample as negative (e.g. non-toxic) compared to the number of correctly labelled positive samples (e.g. toxic compounds.). Based on these results, CP is better suited for toxicity prediction and drug safety since the mislabelled, positive compound results in expenses or even damage to health. Furthermore, based on the fluorescent channels an enrichment measure was introduced that is calculated for the aforementioned two groups of PubChem assays. This enrichment metric indicates unusual cell organell activity. The hypothesis was that PubChem assays well predictable from CP data should have increased enrichment, which was the case for four out of five fluorescence microscopy channels.
As a final step phenotypic terms were manually generated for categorization of the different PubChem assays. These terms correspond to cellular mechanisms or morphological processes. The phenotypic terms were generated unbiasedly but are subject to imperfect knowledge and human error. A bioassay may or may not be associated with a phenotypic annotation. The phenotypic annotations that are found to be enriched for better performing PubChem assays might be able to guide the pre-selection of bioassays in future projects with CP descriptors. The enrichment analysis of phenotypic annotations detected that PubChem assays related to immune response, genotoxicity and genome regulation and cell death are best characterized by CP data.

2 Introduction

Currently, pharmacological drug development focuses on well-established biochemistry based approaches to find and optimize new drugs. However, the challenges these methods face are manifold. High costs related drug failure rates during various clinical trials and commercialization 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.[5] Academia demonstrated computational tools to be employable to many challenges of the health industry like costs of drug target validation, drug safety and commercialization.[6][7] Albeit, chemo- and bioinformatics are novel and complex disciplines recently fostered by technological advancements in high-throughput methods. Thus, the health industry does not yet benefit from promises like computer-aided identification of drugs and drug targets on a large scale.

New high-throughput methods and automated microscopy gave rise to the development of high-content imaging which is frequently combined with small compound perturbations inflicted on biological systems. High-content-imaging applies up to six fluorescent dyes allowing to portray up to five different compartments per cell. This method, also referred to as CP can be used to screen a wide range of compounds and capture the induced morphological changes.[8] CP assays capture compound perturbed biological systems and automatically resolve cellular characteristics that can be interpreted by computational models in context of a morphological fingerprint (or CP feature vectors).[3] The raw images from high-content imaging are processed, mostly by the software CellProfiler[2] which extracts up to 1800 numerical features per image (e.g.nucleus shape, endoplasmatic reticulum (ER) texture). The features from CP assays can be interpreted as a morphological fingerprint, unique for each compound.[8] By now, many different CP assays have been conducted and specific strategies have been developed for specific purposes. A widely used CP data set was generated by Bray *et al.*[9] The images were recorded with sixfold fluorescence staining for imaging five crucial cellular organelles (further information see section 4.4).[3] The concept of CP is visualized in figure 2.1.

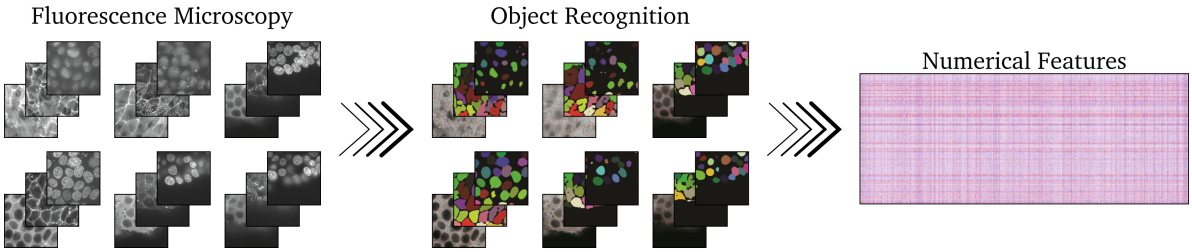


Figure 2.1: Visualization of a CP Assay. First cellular images are generated by fluorescence microscopy, then cellular objects and compartments are recognized by CellProfiler from which a large data table is generated containing the morphological fingerprint for each compound. Figure from Rohban *et al.* and Carpenter *et al.*, modified.[2,10]

Recently, Gustafsdottir *et al.*[11] used CP data 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 CP based approach mirrored the findings in literature, which showed that CP data is directly correlated to cellular pathways responding to compound perturbations. Nassiri and McCall[12] used different CP assay data[13] 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 strong model predictiveness among compounds that steer gene expression in the same direction, suggesting common mechanisms of action. Hence, not only can CP data be linked to mechanisms of action but also an in-depth analysis reveals a relation between compounds' mechanisms of action based on machine learning model performance. Furthermore, Rohban *et al.*[10] transduced U-2 OS cells with lentiviral particles carrying cDNA constructs for gene overexpression.[14,15] In their approach, they conducted a CP assay to annotate the overexpressed genes with morphological fingerprints (numerical CP features). After calculating the Pearson correlation between each morphological fingerprint they used hierarchical clustering resulting in 25 clusters for 110 overexpressed genes. The clusters generated from the CP data clustered genes that correspond to similar or identical pathways showing that genes can be connected using relatively inexpensive CP assays. Furthermore they predicted an unknown relationship between the Hippo- and NF-κB-pathway. Lapins and Spjuth[16] annotated compounds of the CP data from Bray *et al.*[3] and from the CMap[17] (gene expression profiles) with their mechanism of action (MoA) or target protein. The information about the compound-wise MoA and targets were obtained from the Drug Repurposing Hub or the Touchstone data base. In total they annotated 1484 compounds present in CP and CMap data with 234 MoAs or targets. As a third set of descriptors structural fingerprints were generated. For several targets and MoAs a trained

random forest classifier (RFC) could present significant discriminatory power (AUC> 0.7). Furthermore, it was found that the 3 different descriptors were complementary to a certain degree, each excelling at different MoAs or targets. Lapins and Spjuth not only showed that CP data could be used to predict compounds MoA but also, that a combination with other identifiers is likely to enhance their applicability domain.

Simm *et al.*[8] studied a CP assay specifically designed for glucocorticoid receptor (GCR) nuclear translocation. After treating H4 brain neuroglioma cells with 524 371 compounds, hydrocortisone is added to stimulate GCR translocation. Next, the treated cells were stained, imaged and processed analogous to the work of Gustafsdottir *et al.*[11] The 524 371 compounds were not only annotated with morphological information, but also with target activity information from 600 biochemical assays. Noticeably, most compounds were covered in few assays only, amounting in a fill rate of 1.6 %. From this sparse activity matrix they built a machine learning (ML) model with CP data as side information to predict all labels within the activity matrix. They evaluated the discriminatory power of their model and 34 bioassays (out of 600) showed high predictivity (AUC>0.9). One of the assays was part of an ongoing discovery project. Within this assay the highest ranking 342 compounds, by matrix factorization, were experimentally tested. 141 (41.2%) of these resulted in submicromolar hits which means a 60-fold enrichment over the initial high-throughput-screening (HTS). Another assay with AUC greater 0.9 was part of an ongoing drug discovery project and could achieve a 250-fold hit enrichment over the initial HTS in an analogous way. Their work presented CP data as highly informative descriptors, that might be repurposed for prediction of sparse activity matrices. Additionally, they demonstrated their potential in ongoing drug discovery projects.[18]

Data science in computational biochemistry has great potential, however, there are caveats that need to be aware of when working with CP and drug safety. The first one is imbalanced data. An assay testing compounds on a potential target will always feature less actives than inactives. Chawla *et al.*[19] propsed a technique that mitigates this effect called synthetic minority oversampling technique (SMOTE). This technique allows to generate synthetic samples that fit into the distribution of the real data points. Another problem is the high dimensionality of CP data which is an intrinsic problem connected to its richness in information. Only a comparably small number of features contains most of the information necessary to predict a given target, which is why feature engineering is usually conducted in CP studies. A CP data set that contains too many features is bound to overfit the data and give overoptimistic predictions on the test set without generalizing particularly well. The aforementioned project of Rohban *et al.*[10] tackled this problem by conducting a principal component analysis (PCA). They reduced the number of features from 2769 to 158 that comprise most of the variance.

In this explorative project, the CP data set of Bray *et al.*[3] is used for the prediction of PubChem assays. The PubChem assays are selected based on their relation to targets presumably

contributing to cytotoxicity.[4] The results are compared against structural fingerprints of the small molecules and the performance metrics are analyzed extensively. From this analysis, conclusions can be drawn, whether and when to use CP data. Whether structural fingerprints add complementary information and which cellular processes are corresponding to excelling predicitve performance of CP descriptors. Insights gained from this project might be able to guide future decision making when it comes to prediction of biological endpoints.

3 Scientific Aim

The goal of this project 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 the concept of cytotoxicity.

Conceptually, this means finding bioassays whose endpoints are related to toxicity for annotation of the CP descriptors. Since this is a comparative approach, structural fingerprints (ECFPs) are used as another set of descriptors. Both annotated data sets are inputted into a ML model and the discriminatory power of this model is evaluated using generic statistical metrics. Another model is trained using the combined set of descriptors which is evaluated analogously.

The detailed procedure starts by preprocessing of the CP raw image data into a machine learning ready data frame. Next, the bioassays are selected from PubChem,[20] based on size and their relation to cytotoxicity. Every bioassay data frame is combined with the CP data frame to obtain annotated sets containing inputs as well as targets for prediction. As a comparison, structural descriptors are generated for all data sets using sklearn funtionalities.[21] To this end, the annotations in all data sets are highly imbalanced, comprising mostly samples labelled as inactive. Herein, this problem is tackled by applying SMOTE to the data sets in combination with undersampling of the majority label (inactive). Two RFC models are trained on each data set and their predictive power is evaluated. Furthermore, to examine if shortcomings of one descriptor can be mitigated by the other, both descriptors are combined and another model is trained and evaluated. Since there is no apparent way to select features manually, statistical methods can be used to meaningfully conduct features selection. PCA, minimal-redundancy-maximal-relevance criterion (MRMR) and random forest feature importance are applied to score and select features from each set of descriptors for merging. Eventually, another RFC model is trained and evaluated using the joint descriptors. Based on the prediction evaluation and feature selection process a rigorous analysis is conducted aiming to explain the results with respect to cellular morphology and cytotoxicity. The focus lies in detecting and analyzing prediction similarities and dissimilaries between either descriptor sets, bioassays or

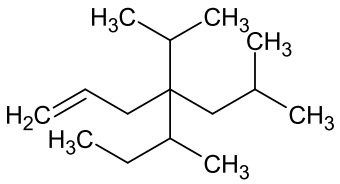
groups characterized by their performance. Thereby, patterns can be detected that transform into heuristics which facilitate the application of CP data to ML problems.

4 Theoretical Background

4.1 Simplified Molecular Input Line Entry Specification - SMILES

The 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 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 addresses six categories: atoms, bonds, branches, cyclic structures, disconnected structures and aromaticity. Also, SMILES considers stereochemical information, however, that is not mandatory since the initial approach to SMILES covers solely two-dimensional information.[22] Atoms are labelled by their element symbol. All elements of a SMILES string are written in square 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, the 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 hydrogen atoms attached to that metal is [MH4]. Hydrogen can also be mentioned on its own in brackets [H], e.g. in its molecular form H_2. Charges are represented with a plus or minus with their respective count inside a bracket.[22] 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.[22]
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 name of the structure is according to the convention of International Union of Pure and Applied Chemistry (IUPAC).[23] The structure and SMILES is shown 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: On the left, side a model compound is shown as an example of nested branching in SMILES. On the right, the IUPAC name and its SMILES string are shown. The SMILES string features parenthesis that imply branching and nested branching.[22]

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 is shown in figure 4.2.

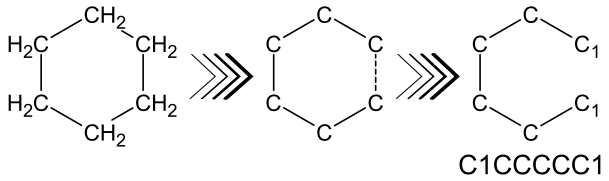
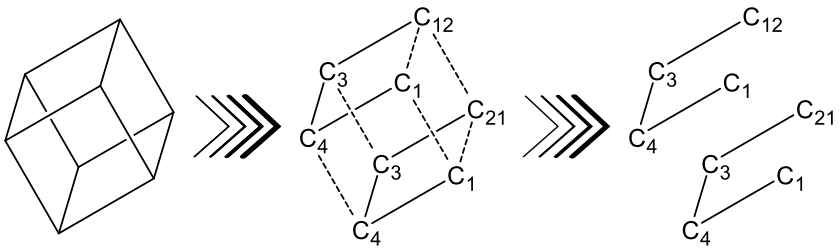


Figure 4.2: Cyclohexane as an example for a cyclic structure. First, the explicit hydrogens are exchanged for implicit ones and the ring is linearized by conceptually breaking a bond which is implied by the dashed line. The carbons connected by the dashed line are being labelled and the resulting SMILES string is shown below the right-hand structure.[22]

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'.[22]



C12C3C4C1C1C4C3C21

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.[22]

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.[22]

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 Hückel'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.[22]

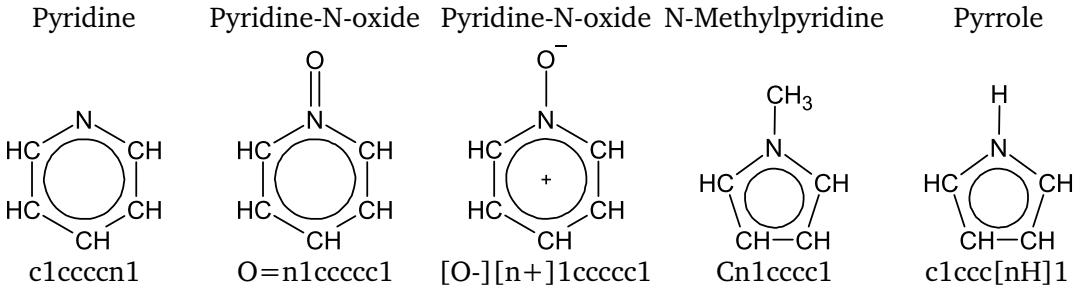


Figure 4.4: Different instances of aromatic nitrogen. Notice that the SMILES string of pyrrole contains an additional hydrogen that preceded the aromatic nitrogen. The aromaticity of an atom is denoted by writing it in lower case letters.[22]

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.[24]

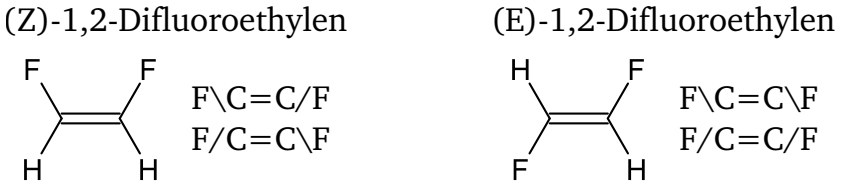


Figure 4.5: Example of double bond configuration in SMILES notation. On the left (Z)-1,2-difluoroethene is shown and on the right is (E)-1,2-difluoroethene with their SMILES notation. Both notations shown for each structure are valid.[24]

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 seen 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 '@'.

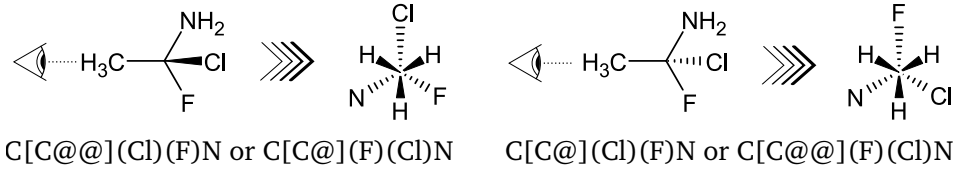


Figure 4.6: Example of enantiomer SMILES strings.[24] Both molecules are pictures in the same way. Above each depiction is the name of the chemical formula followed by a schematic. The eye indicates the point of view along the CC axis. The resulting view of the structure is shown on the right of each subfigure. Written below are two equally adequate SMILES strings for each structure.

4.2 Canonical SMILES

In general, SMILES strings do not claim to be unique identifiers. There are many equivalent options to generate a SMILES string for a given structure. Nowadays, computational biochemical research accesses structures from many different sources and databases making the requirement of a unique identifier evident. The SMILES notation was developed with this objective in mind. So-called canonical SMILES strings fulfil this objective. They are 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.[25]

For finding a unique way of labelling atoms in a molecule, invariant structural properties are necessary. Thus, five properties are considered atomic invariants. Those would be (1) the number of connections, (2) the number of non-hydrogen bonds, (3) the atomic number, (4) the sign of the charge and (5) the number of attached hydrogens. They are called 'invariant' because they are invariant to atomic order changes within structural notation and they are exchangeable as long as this principle is not violated. The numbers in the parenthesis in front of each property correspond to a pre-defined prioritization. In summary, the atomic invariants assign five integers to every atom in a molecule. The so-called individual invariant can be obtined by simply combining these integers in order of their prioritization. Given the mehtyl carbon of 2-(acetyloxy)benzoic acid in figure 4.7 with the individual invariant 110603. From this individual invariant it can be concluded that this atom has 1 connection, 1 bond to a non-hydrogen neighbour, an atomic number of 06, 0 charge and 3 attached hydrogen atoms. 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 symmetric and require a tie-breaking function for absolute uniqueness (see next paragraph).[25]

After assigning every atom an individual variant, those are compared among all constituting atoms and are ranked by magnitude. The final atom labels depend on the topology, as well. Therefore, the nodes (atoms) rank is extended by its corresponding prime (for 1, 2, 3, the corresponding primes are 2, 3, 5). The so called new invariant is obtained by multiplying the corresponding primes of all neighbours for every atom. Afterwards, ranks are assigned again, based on their current rank and their new invariant. 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 a 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.[25] The canonical labelling process for 2-(acetyloxy)benzoic acid is shown in figure 4.7.

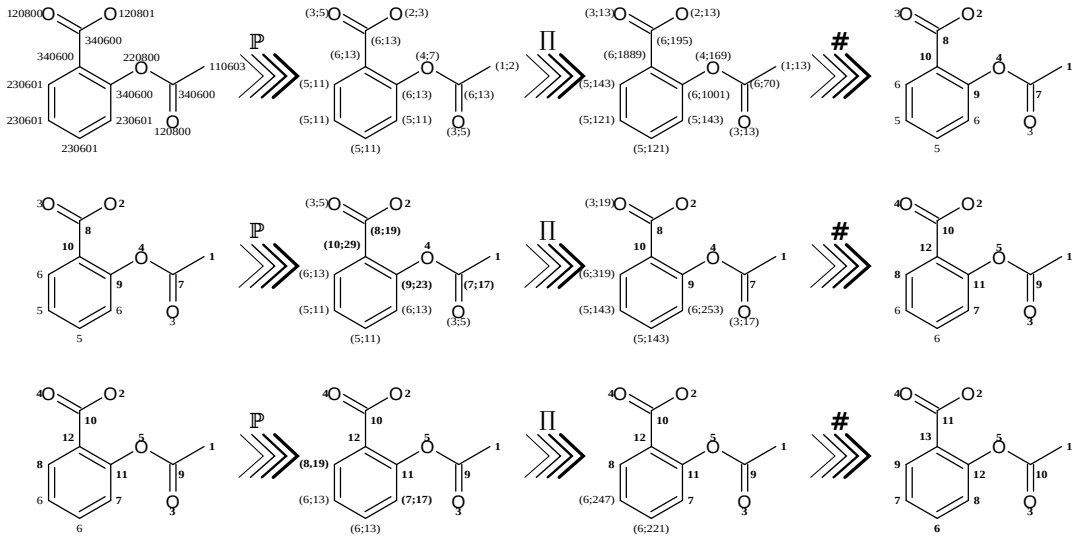


Figure 4.7: Canonical labelling with 2-(acetyloxy)benzoic acid. Every row corresponds to consecutive iterations of the CANON algorithm. The blackboard bold $\mathbb{P}$ denotes finding corresponding primes. The greek letter Π denotes taking the prime products of all atoms and the hashtag denotes the ranking of atoms. Bold numbers denote ranks that reached invariance.

THE GENES part of the algorithm can utilize the uniquely ordered ranking to chose the start node and which nodes to prioritize at branching points, etc. As an entry point for the generation of canonical SMILES, the node with the lowest ranking is chosen. Branching decisions are made in the same fashion, i.e. the branching option with the lowest rank is chosen and followed until a dead end has been reached. A special rule applies when branching into a ring with a double or triple bond. To avoid opening the ring at any multi-bond the algorithm will always branch towards the multi-bond. Also, the ring-opening digits 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.[25]

4.3 Extended-Connectivity Fingerprints

The ECFP is a structural fingerprint that was developed to capture molecular features relevant for molecular activity.[26] ECFPs are also commonly referred to as Morgan-fingerprints since their development is partially based on the Morgan-algorithm which pursues rigorous canonicalisation similar to the CANON algorithm in section 4.2.[27] 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 six atomic invariants, similar to the ones in section 4.2 canonical SMILES algorithm (number of connections, number of bonds, atomic number, atomic mass, atomic charge, number of attached hydrogens).[24] However a seventh invariant is taken into account as well: whether the atom is part of at least one ring structure. A hashing algorithm maps the atomic invariants to a 32-bit integer. Any functional hashing algorithm that reproducibly maps the input onto a 32-bit integer is a sensible choice, since the only requirement is that a variant is uniquely mapped to the same 32-bit integer every time it is hashed. The 32-bit integer is also referred to as core identifier. Within the initialization step, it corresponds to a substructure that contains information about one atom and its bonds and is appended to a list called ECFP-set. After completion of the algorithm, the ECFP-set is equal to the fingerprint.[26] Therefore, the result of the initialization are a set of core identifiers $[c_0^{(0)}, c_1^{(0)}, c_2^{(0)}, ..., c_n^{(0)}]$. The core identifier for atom 1 of the initialization (zeroeth iteration) would be $c_1^{(0)}$.

After the initialization the first iteration starts by randomly picking an atom, let that be atom 4. Next, the iteration number (1) and the core identifier, $c_4^{(0)}$, are appended to a temporary list. Afterwards, the neighbouring core identifiers are appended to the temporary list together with their respective bond order. b_{4j} denotes the bond order between atom 4 and its neighbour, j. The bond order can either be 1, 2, 3 or 4 for aromatic bonds. Let the resulting temporary list have the following format $[1, c_4^{(0)}, b_{34}, c_3^{(0)}, b_{45}, c_5^{(0)}, b_{46}, c_6^{(0)}]$. In this example the temporary list comprises eight entries which are inputted into a hashing function that returns a 32-bit integer. This integer is the core identifier of atom 4 for, $c_4^{(1)}$. Once this process is completed for every atom, all core identifiers are updated to the new core identifier simultaneously and are appended to the ECFP-set. The temporary lists are being discarded. After iteration 1, the ECFP-set comprises of the elements $[c_0^{(0)}, ..., c_n^{(0)}, c_0^{(1)}, ..., c_n^{(1)}]$.[26]

The second iteration is conducted in exactly the same way as the first iteration. However, the identifier of core j of the first iteration contains information about its surrounding cores and their bonds. Therefore, information from up to two bonds are incorporated into the core identifiers of the second iteration, $c_j^{(2)}$. Thus, from iteration to iteration the core identifiers can be understood as the core atom within a larger and larger structural neighbourhood. $c_j^{(0)}$ is

a 32-bit integer that encodes the atom j and its adjacent bonds. $c_j^{(1)}$ is a 32-bit integer that encodes atom j, its neighbouring atoms within one bond length, their bond orders and their adjacent bonds and so on.[26]

After the specified number of iterations is completed, duplicate 32-bit integers are removed from the ECFP-set since they encode for the same subtructure. Butyramide is shown in figure 4.8 as a conceptual example of fingerprint generation.[26]

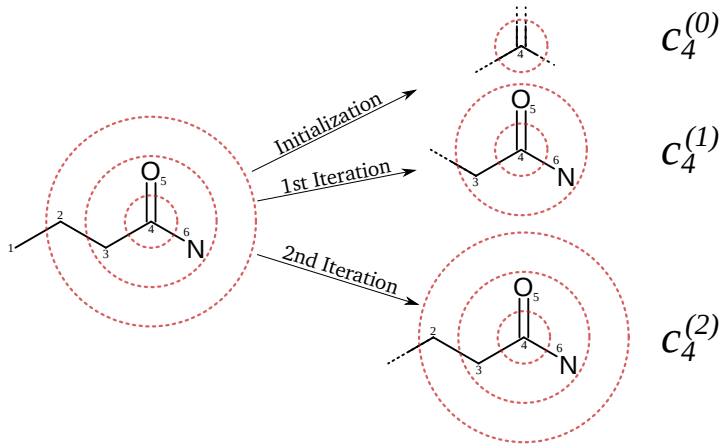


Figure 4.8: Fingerprint iterations with substructures for one atom. Atom 4 of butyramide iterated, denoted by the red circles. The smallest circle denotes initialization of atom 4, resulting in $c_4^{(0)}$ only containing information about the core atom and adjacent bonds. The first iteration is denoted by the intermediate circle and results in $c_4^{(1)}$. The second iteration is denoted by the outer circle and results in $c_4^{(2)}$.

So far ECFP-set contains 32-bit identifier. A substructure could be encoded by the integer, e.g. 1559650422 which would correspond to an ”on” bit within a bit set of 2^{32} bits. Since a hash space of 2^{32} is quite vast, usually the 32-bit integers are mapped onto a vector of 1024 (or 2048) bits by yet another hashing algorithm. Even though the bits in the new 1024-bit-vector cannot be directly decoded into molecular substructures, the identifiers and substructure pairs can be saved and subsequently accessed. In summary, ECFPs are generated by hashing atomic invariants into identifiers, which are then updated a specified number of times with information of their immediate surroundings. Eventually, the core identifiers are hashed into a bit-vector (usually 1024 or 2048 bits) that indicates present substructures by ”on” bits (1) and missing ones by ”off” bits (0). The procedure is visualized in figure 4.9.

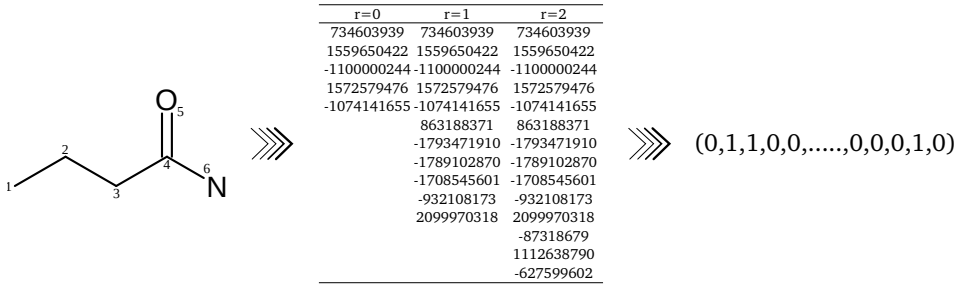


Figure 4.9: Butyramide as an example of ECFP generation. Starting from the structure, the identifiers are generated for radius 0, 1, and 2 and appended to the ECFP-set. From the identifiers with radius 2, a bit vector of a certain length is obtained.

4.4 Cell-Painting Assay

Cell-Painting (CP) refers to a high-content-screening method that generates cellular image data from high-throughput fluorescence microscopy experiments. A CP assay consists of several consecutive steps which result in tabulated raw image data. These steps consist of cell culture, treatment, staining and fixation, automated image acquisition and feature extraction.[3]

This project uses CP data generated by the Broad Institute.[3] U-2 OS cells were used as the target organism in the cell painting assay reported by Bray *et al.*[3] 1500-2000 cells were seeded into every well of multiple 384-well clear bottom plates and incubated at 37 °C for 24 hours.[28] Then, compounds were added to the well in quadruplicates of varying concentrations. In total 30409 different compounds were added and incubated for 48 hours. The compounds used can be categorized as small molecules and were 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.[3]

In total, six different fluorescent reagents were used to stain 5 different cell-organelles. Only two of the reagents were applied to the living cell culture, the remaining four were applied after fixation of the cells. A combination of two reagents was used to stain the F-actin cytoskeleton, plasma membrane and Golgi apparatus. Another reagent is used to stain the nucleoli and the cytoplasmatic RNA. Additionally, individual reagents are used to stain the ER, the nucleus and the mitochondria, respectively. The six reagents are listed in 4.1 together with the cell organelles they respond to and the catalogue number.

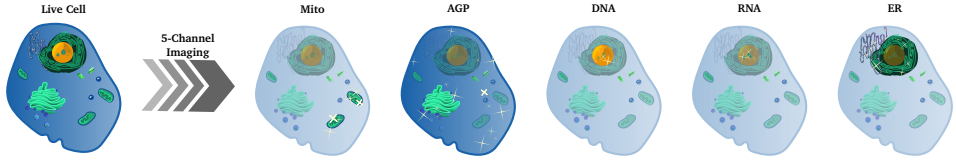


Figure 4.10: Concept of 5-channel imaging. The live cell is stained with fluorophors and the imaged in 5 different channels. Each highlighting different compartments in the cell. The highlighted compartments of each channel are opaque and light emission is implied by sparkles.

Table 4.1: The fluorescent dyes used in the CP assay are listed here. The list contains the names of the fluorophours, the cell organelle(s) that they are targeting and the catalogue number that refers to the Invitrogen catalogue.[28] The maxima of the excitation and emission wavelengths of the fluorophors are shown in nanometer.

Fluorescent reagents	Cell Organelle	$\hat{\lambda}_{ex}\text{nm}^{-1}$	$\hat{\lambda}_{em}\text{nm}^{-1}$	Invitrogen
Mitotracker Deep Red	Mitochondria	644	665	M22426
Wheat Germ Agglutinin, Alexa Fluor 594	F-actin cytoskeleton, plasma membrane, Golgi	589	615	W11262
Concanavalin A. Alexa Fluor 488	ER	495	519	C11252
Phalloidin, Alexa Fluor 594	F-actin cytoskeleton, plasma membrane, Golgi	581	609	A12381
Hoechst 33342	Nucleus	350	461	H3570
SYTO 14 green fluorescent nucleic acid stain	Cytoplasmatic RNA, Nucleoli	521	547	S7576

After the compound treatment, a staining solution of Mitotracker and wheat germ agglutinin (WGA) was added and incubated for 30 min at 37 °C. Afterwards, cells were fixed using paraformaldehyde. Afterwards, staining solutions containing Phalloidin, Hoechst 33342, SYTO 14 and Concanavalin were applied to the cell containing wells and incubated for 30 min. Finally, the plates were thermally sealed and stored at 4 °C.[9 28]

In the next step, images were generated via automatic fluorescence microscopy. Five fluorescence channels were used which scan the plates for different wavelengths that were emitted by the fluorophors tagging specific cell organelles. The channels are labelled DNA, RNA, AGP (F-actin cytoskeleton, Golgi and plasma membrane), Mito (mitochondria) and ER (Endoplasmatic Reticulum).[3]

After the automatic image acquisition was completed, the so-called CellProfiler[2 29] software generated numerical features from these images. CellProfiler has its standard pipeline to generate cellular features from fluorescence images. The concepts of this pipeline are visualized

in figure 4.11. First, the images were aligned, cropped and an illumination correction is applied followed by 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.[2] Taking the nuclei as a starting point, the secondary objects, like cell edges, the cytoplasm and nuclear membrane are identified next. After the cells have been identified, CellProfiler conducts different measurements to calculate a variety of features related to cellular compartments and organelles. These features include the area, shape, texture and other more complex features.[2] The dataset that is used in this work comprises 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 relevant to this work. After the feature extraction via CellProfiler, the finalized data set is called raw image data.

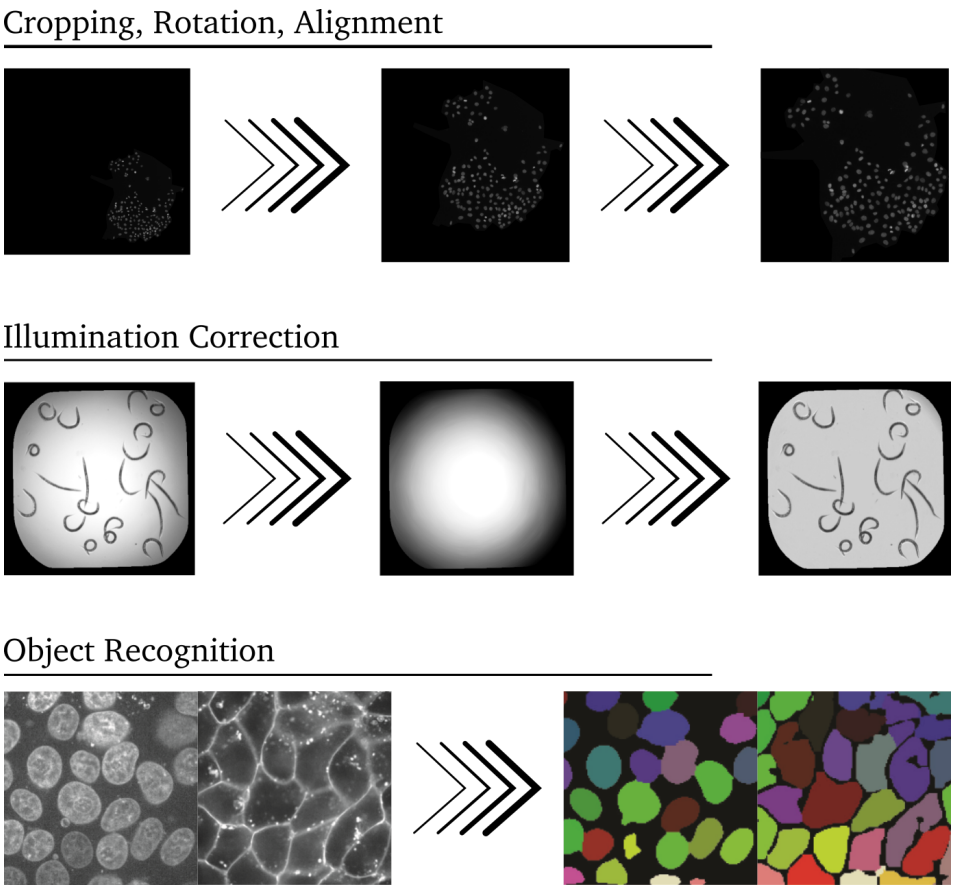


Figure 4.11: Conceptual visualization of the CellProfiler workflow. The first image shows fluorescent microscopy images of human cells. These are cropped and rotated to yield a better view of the cells.[30][31] In the second step the illumination correction function is applied to a generic image.[31] In the last step (Object Recognition), the identification of stained nuclei and membranes of human pluripotent stem cells are shown as an example.[32] The obtained images were slightly modified.

4.5 Raw Image Data

CellProfiler extracts numerical features from cellular images. The resulting raw image data is a very large spreadsheet whose columns contain the features and the rows correspond to wells from which the original images were taken. Additionally, the rows are identified by the compounds that have been used to treat the respective wells, exempt control wells which are treated with dimethyl sulfoxide (DMSO) only. Every compound is measured in quadruplicates as a minimum. Also, some are measured in octuplicates as well as in different concentrations. Relating to the raw image data spreadsheet for every compound there are at least four rows

corresponding to four wells. Compounds whose features have been extracted for only one concentration are referred to as single-concentration-compounds and the compounds that appear in multiple concentrations are called multi-concentration-compounds.
The spread sheet does not only contain numerical features extracted from CellProfiler but so-called metadata, too. Metadata refers to methodological information relevant for the experimental procedure. Hence, compound concentration, plate number, plate map number and other information are being categorized as metadata. Also, the information whether the row corresponds to a treated or control well, is stored within the first 17 columns before the listing of CellProfiler features from column 18 to 1801. From these 17 only five are important for the suceeding steps. Among plate, location, role and concentration these columns contain further information about the molecular structure of the respective compound as a SMILES string (see section 4.1). In table 4.2 the column header names are listed together with a brief description. The names in table 4.2 correspond to the ones from the original raw image data file.[3]

Table 4.2: Below, the names of the most important metadata 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 compound as a SMILES string
Metadata_mmoles_per_liter	States the compound concentration for treated samples
Metadata_broad_sample	Identifier assigned by Broad Institute that varies inconsistently either with compound, concentration or plate number

4.6 PubChem-Assay

Within the subject of ML targets are features of interest. A ML algorithm attempts to predict targets from a given input analogously to a function that calculates y from a given x. The label of a photograph is a classical example of a target. A worked example classifies dogs and cats from images, where the targets would be a either 'cat' or 'dog'. The same principle can be applied for bio- and chemoinformatics. Typical targets in this scientific area are 'active' and

'inactive' for a certain bioassay. Nevertheless, data that is annotated correspondingly is not abundant.

The database that supplies the targets for this project is the PubChem database.[33] The PubChem database contains information about chemical compounds and their bioactivities found in various assays. The bioassays in PubChem are assigned a unique assay identifier (AID) and data page features descriptive information and the corresponding readout. The descriptive part contains, among others, information like the name and theoretical background, experiment procedure, data source and a description of the readout.[34]

In general, the depositor of a bioassay can provide as many detailed results as necessary.[35] However, PubChem requires the depositor to submit a summary result for each chemical sample. This summary result constitutes the numerical 'bioactivity score' and the categorical 'bioactivity outcome'. The bioactivity outcome can assume 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.[34]

AID 720532

Kolokoltsov et al.[36] could prove that virus cell-entry strongly depends on the host-cell mediators. Mediator for cell-entry can be Signalling factors, membrane attachment factors and endosomal and lysomal factors.[37] By targeting the host-cell mediators the emergence of drug-resistant virus mutants is less likely, since the cellular mutation rates are up to 6 orders of magnitude smaller compared to viruses.[38] The bioassay AID 720532 is an assay that targets cell-entry of the Marburg virus. Non-pathogenic vesicular stomatitis virus (VSV) with the envelope glycoproteins of the Marburg virus were used as a pseudovirus referred to as VSV-MARV. The pseudovirus contains a Photinus luciferase reporter gene within its genome. Therefore, cell-entry is detected by changes in luminescence. The assay uses HEK293 cells in 1536-well plates. After applying compounds to the respective wells the plate is incubated. Next VSV-MARV is applied inf sub-saturating amounts. Luciferase signals reflect the virus titer able to infect the cells for the different compounds.[39][20] During a similar screening against Ebola virus entry (same family as Marburg virus) many signalling pathways relevant to cancer, gene regulation and cell cycle control were found relevant in preventing cell-entry.[38]

AID 651635

The expansion of a polyglutamine domain of Ataxin-2 due to a mutation in the Ataxin-2 gene (ATXN2) causes a neurodegenerative disease called spinocerebellar ataxia type 2 (SCA2).[40] The objective of bioassay AID 651635 is to find small molecules that inhibit ATXN2 expression. For this purpose, SH-SY5Y cells with a ataxin-2-firefly-luciferase transgene are transferred into a 1536-well plate, then treated with compounds and Gly-Phe-7-amino-4-trifluoromethyl coumarin to access cell-viability. Eventually the luminescence is measured to infer expression of the ataxin-2-firefly-luciferase transgene.[41] The cell line SH-SY5Y is neuroblastoma cell line obtained from a human bone marrow biopsy.[42]

4.7 SMOTE - Synthetic Minority Oversampling Technique

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.[19] For a strong label imbalance the ML algorithm performs well, even if it only classifies one label sufficiently. By generating data points that are presumably representative of the minority class, the ML algorithm has to shift its focus towards the minority class to achieve a better performance.

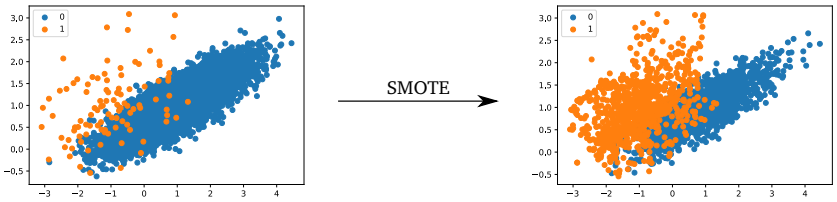
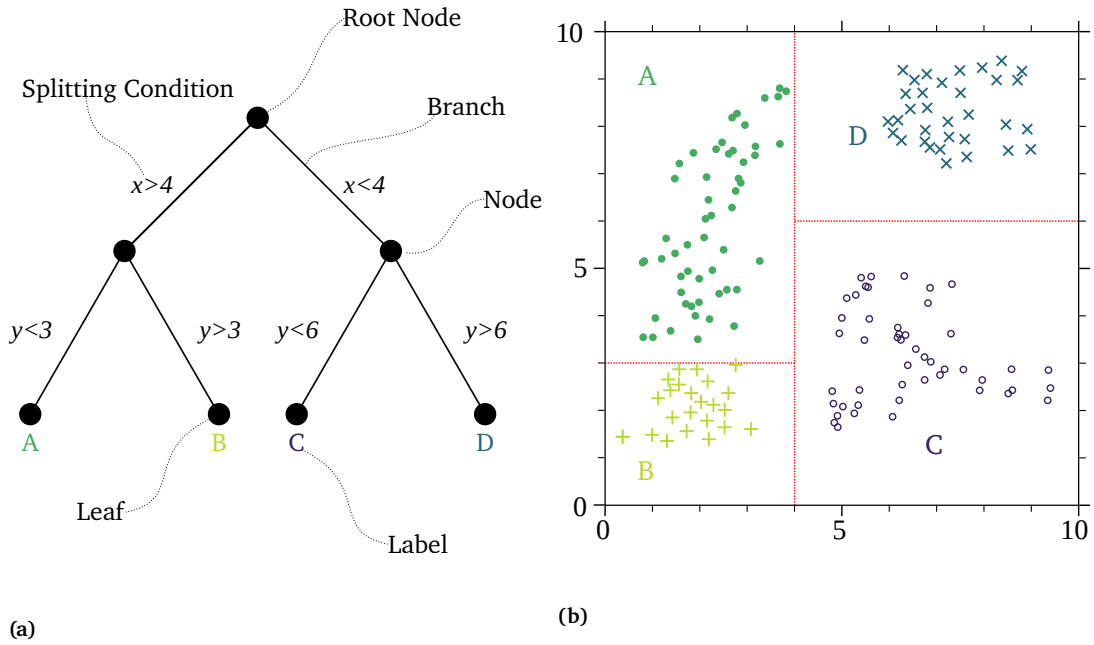


Figure 4.12: SMOTE applied to a 2D data set. The minority class (orange) is oversampled and the majority class (blue) is undersampled.

4.8 Random Forests

In 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 tool in stochastics, closely related to the propability tree. 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. 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, f, guides the splitting of the dataset at each node. Moving further down the branches the subsets are split into smaller and smaller subsets until they reach the leaves. The leaves correspond to labels present in the data. A leaf assigns its label to every sample of a subset that arrives at said leaf. A certain numerical rule, the splitting condition, defines how to split the data set and each node splits the entering data set based on one certain feature. However, since the decision tree is sequential, the information from earlier nodes is 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 4.13).[43]



(a) (b)

Figure 4.13: (a) Example of a decision tree with the description of the individual elements; (b) Data that is classified based on the examplary decision tree; A decision tree and the geometrical representation of the classification process is shown for a simple 2-D data set comprising 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. At the second nodes the other splitting criteria are applied and eventually the data reaches the leaves where the labels are assigned.[43]

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?[43]

The simplest of these questions is the first one. The features of a decision tree are chosen 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.[43] 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. The propability that too many redundant features are chose by the decision tree hinders its predictive capabilities.[43]

The second question about the splitting condition can be solved by applying information theory. There are several methods to determine a splitting condition. The information gain method is described here as an example. First, 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. To obtain the entropy $H(D_l)$ of the left subset D_l the frequency of a class label c within this subset is multiplied with its binary logarithm and then summed up over all different class labels. The frequency of c within D_l is calculated by dividing the number of corresponding labels $n(c; D_l)$ by the total number of data points in D_l, $N(D_l)$. The entropy of the right subset after splitting is obtained accordingly.[43]

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

$H(D)$ corresponds to the number of 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).[43]

$$w_\mathrm{r} = \frac{N(D_\mathrm{r})}{N(D)} \qquad\qquad w_l = \frac{N(D_l)}{N(D)} \tag{4.2}$$

Finally, the information gain I of a certain node is given by equation (4.3).[43]

$$I = H(D) - w_\mathrm{r} \cdot H(D_\mathrm{r}) - w_l \cdot H(D_l) \tag{4.3}$$

In general, the greater 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). Within the assigned feature of the node 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.[43] The concept of splitting is visualized in figure 4.14.

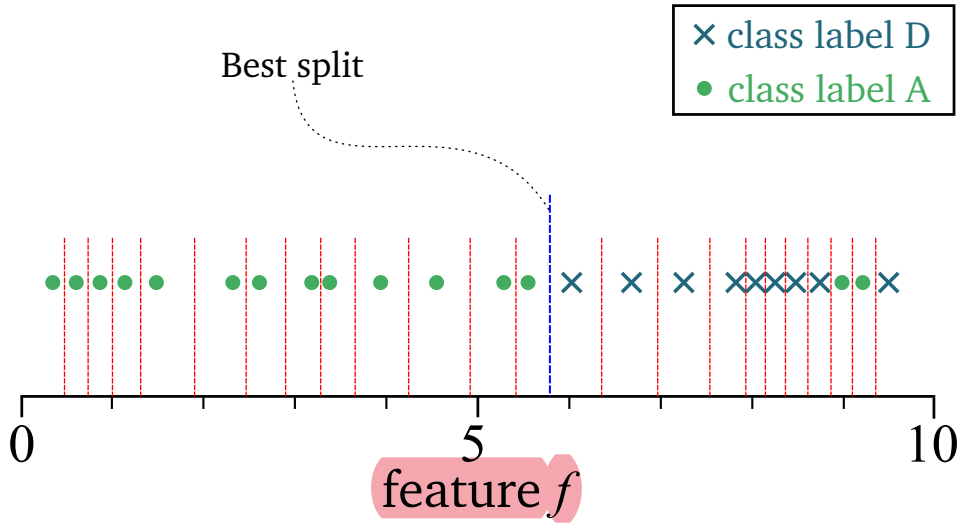


Figure 4.14: Visualization of the splitting condition. One feature of the data points is tested for the optimal information gain. The two classes A and D are plotted aling feature f and the optimal split is marked in blue, whereas all other splits are denoted as red dotted lines.

Eventually, the leaves are reached and have to assign labels to the data points in the final subsets. One leave will assign the majority label, present in the final subset during training. One decision tree is considered a very poor classifier. Many decision trees on the other hand can become very powerful. A RFC comprises a large set of decision trees. A sample enters every pre-trained decision tree within the RFC and every decision tree computes a label. The simplest RFC uses a majority vote to determine each label, i.e. the label most trees computed for that sample is assigned.[43]
Apart from the splitting conditions, the feature selection and 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 to name a few.[21] As opposed to the splitting condition which is chosen by mathematical optimization, the hyperparameters have to be entered by the operator.[43]

4.9 Cross Validation and Splitting

Usually, an advanced ML model has enough parameters to fit a given data set optimally, 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 unseen data would be very poor, since

the model would not be able to abstract from the training dataset.[44]
Splitting the data and using one part for training and one part for performance estimation mitigates the selection bias. However, given random splitting of the data, the model will not explore the complete data set and the individual splits can have a biased label distribution, as well. This would result in an underestimation of the performance. Both, the issue of overfitting (or selection bias) and insufficient use of the data set can be circumvented by cross-validation (CV).[43] During CV the data set is split in k subsets with equal sample count. These subsets are referred to as folds. Next, the model iterates through k cycles of training and evaluation. In every cycle, another fold is used as validation set, whilst the remaining $k-1$ folds are used for training. All evaluations 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[44] A visual representation is shown in figure 4.15.

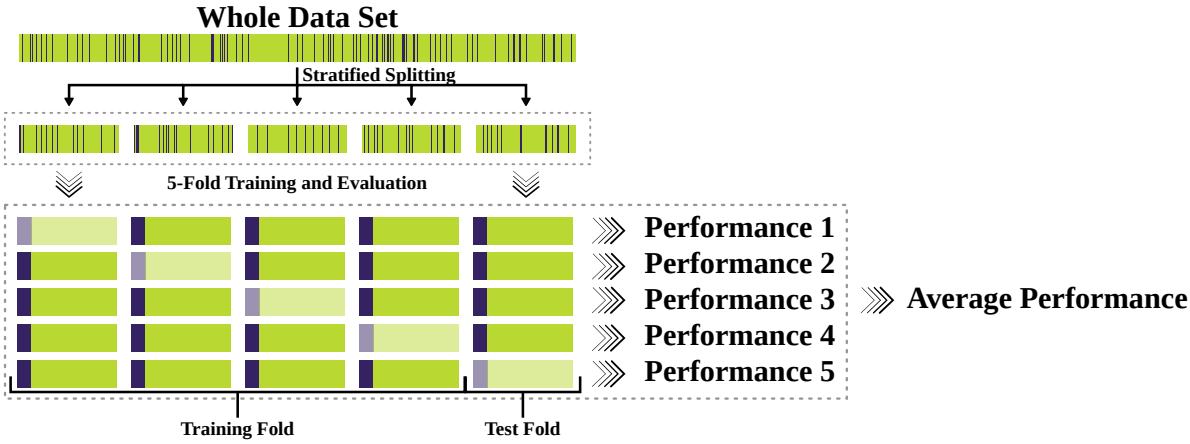


Figure 4.15: Visualization of CV. The whole data set contains 83% negative (green) data points and 17% positive (purple) data points. It is split up into five equally large subsets of equal distributions of positive and negative samples. For 5-fold CV the ML model is trained five times. Every time another subset is used for validation (indicated by the lighter shade) whilst the others are used for training. The individual performances are averaged to yield a good estimate of the predictive power of the model.

Selecting the hyperparameters is another caveat, that results in performance overestimation. Hyperparameters were mentioned in section 4.8 which are parameters specified by the user 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. One possibility is to supply a value list for each hyperparameter which is called a parameter grid. The algorithm uses every combination of values consecutively to find the hyperparameters that perform best. Thus, the hyperparameter selection itself exploits information in the data. Otherwise, the hyperparameter variation would not impact

the prediction performance. Therefore, optimizing the hyperparameters and using k-fold cross validation (KFCV) on the RFC parameter optimization only, still results in exaggerated model performance. 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. The concept is shown in figure 4.16[44]

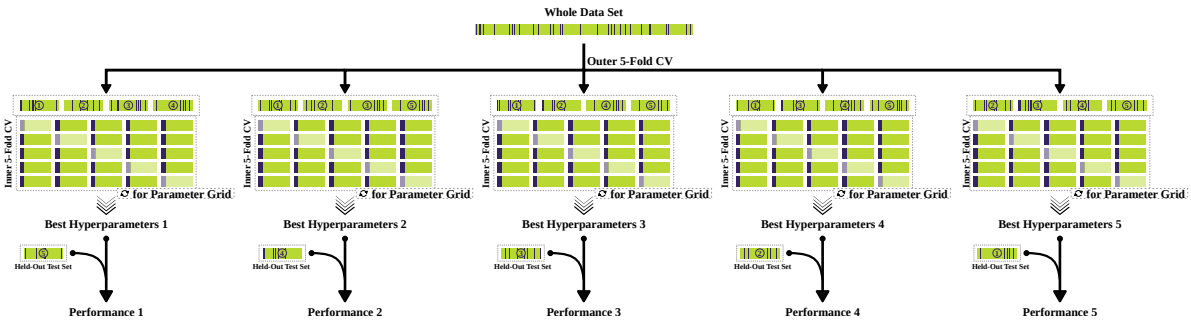


Figure 4.16: Visualization of nested CV. The whole data set contains 83% negative (green) and 17% positve (purples) samples. The outer CV splits the data into five sub sets. Four of these sub sets are used in ther inner CV. The other one will be used as a held out test set for performance evaluation. The outer CV results in five instances, each considering a different subset for evaluation. The inner CV starts by splitting the outer training set into five subsets. Next, those subsets are used for training and evaluating all hyperparameter combinations supplied in the parameter grid. The best parameters are then tested to estimate the model performance.

The splitting strategy that is chosen 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 sub sets that are randomly chosen, each exhibiting an equal label distribution which counteracts pessimistic prediction performance.[45]

4.10 Performance Evaluation

To evaluate the performance of an RFC there are several different metrics available. Since the work presented here is concerned with a binary classification problem (i.e. only two labels are possible per sample) the descriptions below are only valid for binary classification. 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 (MCC) and many more.

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.[46]

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 (also referred to as contingency table). 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 4.17.

		Predicted Labels: True	Predicted Labels: False
True Labels	True	TP	FN
	False	FP	TN

(a) Structure of a confusion matrix

		Predicted Labels: True	Predicted Labels: False
True Labels	True	1274	810
	False	328	7588

(b) Confusion matrix with exemplary values.

Figure 4.17: 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 computed. For example the TPR and TNR, more widely known as the sensitivity and the specficity, aswell as the balanced accuracy (BA) and the MCC. The TPR describes the frequency of correctly positive-labelled samples whereas the false positive rate (FPR) depicts the frequency of incorrectly positive-labelled predictions.[46]

$$TPR = \frac{TP}{TP + FN} \tag{4.4}$$

$$FPR = \frac{FP}{TP + FN} \tag{4.5}$$

Analogously, the TNR describes the frequency of the correctly negative-labeled samples within the predictions.[46]

$$TNR = \frac{TN}{TN + FP} \tag{4.6}$$

The balanced accuracy, BA, is simply the average of the TPR and the TNR.[47]

$$BA = \frac{TPR + TNR}{2} \tag{4.7}$$

TPR, TNR and BA can all adapt values between zero and one. One correponds to a perfect model. The Matthews correlation coefficient, MCC, is a metric that scores high if the number of correctly predicted samples is significantly higher than the number of incorrectly assigned ones.[48]

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \tag{4.8}$$

The MCC can score between -1 and 1 where 1 represents very good performance, 0 refers to a random model and -1 describes consistently false predictions.[48]

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 interpreted as a probability p for a certain class label by dividing the votes for a certain class label by the total number of votes. The resulting propability is inbetween 0 and 1. A threshold t can be applied that determines the label depending on the propability.

$$z = \begin{cases} 0 & \text{if } p < t \\ 1 & \text{if } p \geq t \end{cases} \tag{4.9}$$

A majority voting refers to a threshold of 0.5. However, the threshold can be varied from 0 to 1. For a threshold of zero, all labels will be predicted as 'positive' (i.e. '1'), since no label will have p smaller than zero. For a threshold of one 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 is iterated from 0 to 1 to obtain all different confusion matrices from which the TPR and FPR are calculated. 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.[46] In 4.18 different roc curves are shown that correspond to well or not so well performing models.

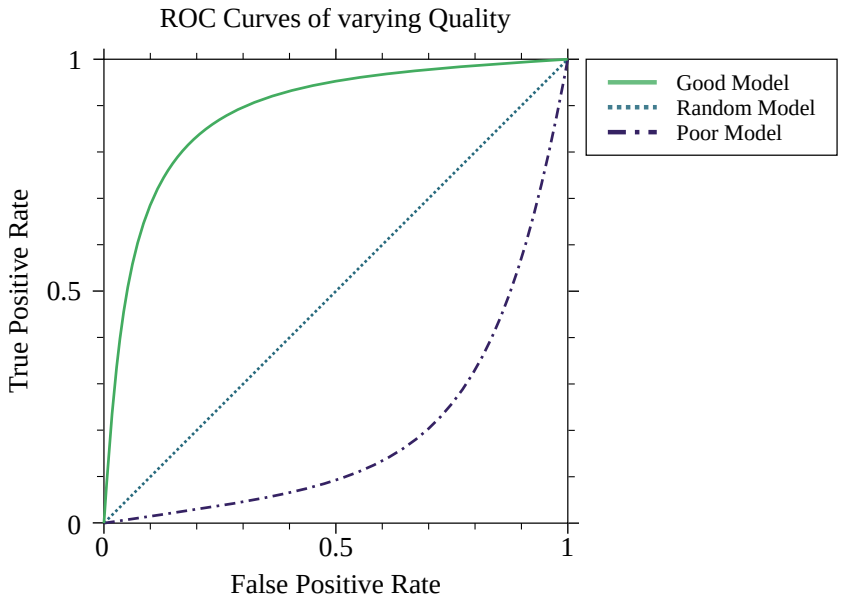


Figure 4.18: Examples of ROC-curves. The diagonal depicts a model that chooses randomly. If a curve is above the diagonal it predicts better than random, below means worse.

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 is chosen by maximizing the variation in each dimension.[49] Therefore, the number of new axes, referred to as principal components, is the same as the original dimensionality. However, the first few principal components usually account for a large 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.[43] Transforming the coordinate system also gives information about the contribution of each feature to the principal components, which can be used to infer feature importance for the original data set.[49]
Random forest feature importance can be either measured by gini impurity or by entropy (or information gain, see section 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.[50]

$$G_k = 2p_1p_0 \tag{4.10}$$

From equation (4.10) follows, that a small gini impurity refers to a very succesful split from the parent node with the index $k-1$. If both descendant nodes contains one label only, the parent node has achieved the best split possible. 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 $k+1$.[50]

$$I_{\mathrm{d}}^{(f)} = \sum_{k} G_k^{(f)} - G_{k+1} \tag{4.11}$$

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.[50]

$$I^{(f)} = \frac{1}{N_d} \sum_{d} I_{\mathrm{d}}^{(f)} \tag{4.12}$$

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~~ share their importance between each other which results in two difficulties. Firstly, very important features are scored lower since the importance measure is split within highly correlated feature clusters. The second is, that the final selection 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 biased predictions based on the information of a few clusters. To overcome these two problems, 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.[51] 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.[51]

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. Jupyter notebooks are used as user interface for python programming. Furthermore python scripts and bash scripts were used as well.

5.1 Descriptors- CP and ECFPs

As inputs for the RFC two different descriptors are chose. The first descriptor are the morphological information that are extracted from the **rid!** of the CP assay of Bray et al.[3] Before the **rid!** can be inputted they need to be preprocessed, which is described in detail in section 5.3. The baseline descriptors that CP data has to compete with are ECFPs. For all compounds present in the final data sets the structural identifier is calculated from the canonical SMILES utilizing the Chem package from the RDkit python library.[52] The obtained compound identifiers are 2048-bit vectors with radius 2. Every bit in this vector is considered a feature for the machine learning algorithm.[52]
First, the CP featurs and ECFP features are used separately as inputs. Afterwards, selected features from both features spaces are used as input together.

5.2 Targets

For creating annotations for the input vectors the PubChem bioassay database is queried. PubChem comprises more than 1 200 000 bioassays.[53]The amount of information stored at the PubChem database is so vast that the process of finding data sets fitting for this project is a problem on its own. A step wise filtering process is created to find relevant bioassays.
First, the 11 biggest folders are downloaded from the PubChem database, each containing up to 1000 bioassays.[54] 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 is downloaded directly from https://pubchem.ncbi.nlm.nih.gov/ and 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.[4] 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 list of CIDs. 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 compounds is conducted by means of the Metadata_broad_sample (which turned out to be suboptimal in section 5.3 and had to be corrected). 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 Metadata_broad_sample identifiers.

Not all 671 assays are used for further investigation. As mentioned in section 4.6, 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 correctly categorize the data since the two classes (active and inactive class) are too imbalanced. 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 classification.

5.3 Preprocessing

As mentioned in section 4.5 the raw image data contains meta and data columns. The metadata columns of the CP data dictate the decision-making process during the preprocessing and the data columns themselves are the subject of preprocessing. The data columns or CP 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 ML-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 $\mathrm{Metadata\_broad\_sample}$ turned out to be a suboptimal identifier because it is not unique for every compound (see section 4.5). Different $\mathrm{Metadata\_broad\_sample}$ values are used for the same compound if it corresponds to a different compound concentration or plate. Therefore, the $\mathrm{Metadata\_broad\_sample}$ was exchanged for the canonical 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 compound-concentration-wise medians of each feature. However, some compounds were measured in multiple concentrations. Therefore, a new metadata column was introduced that labelled each row either as a single-concentration-compound or a multi-concentration-compound. 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 multi-concentration-compounds.
Next the preprocessed raw image data is merged with each of the 52 bioassays on the $\mathrm{Metadata\_broad\_sample}$ identifier. Hence, only identifier that exist in both data frames are accepted. The preprocessed raw image data's metadata features SMILES and canonical SMILES for proper compound identification. From here, the multi-concentration-compounds that are present in the combined data frames are inspected since they require further consideration. The ML algorithm requires one morphological profile per compound, as well as one label per compound. Thus, the most frequent concentration was accepted for each multi-concentration-compound. For this concentration the median of all CP features was calculated.
The preprocessed raw image data only needs to be generated once, however, every bioassay data frame is then merged with the preprocessed raw image data followed by computation of the multi-concentration-compounds' medians. This computational process results in 52 combined ML-ready data set with 1768 relevant features and varying row number since the com-

pound 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

5.4 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 classification.
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 (see section 4.9). 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 as well (see section 4.7). The minority class amounts to 75 % compared to the majority class label after application of this sampling strategy. SMOTE was not applied to the held out test set of the outer loop. In turn, the model is validated with real data points only.
The hyperparameters are optimized using a halving random search method from sklearn.[55 21] For that purpose a parameter grid was used for each inner CV iteration. 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 the parameter grid 500 combinations were randomly sampled for each inner CV iteration and only one best estimator is returned and evaluated in the outer CV iteration. Since a 5-fold CV is in use, five best estimators are evaluated by calculating the BA, MCC, TPR, TNR, ROC-curve and AUC-ROC.

This procedure is conducted for every PubChem assay mentioned in table 5.1 merged with the CP descriptors, the ECFPs and eventually with the feature engineered combination of both.

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 sci-kit learn.[21] The method is applied to the CP-PubChem data sets to find the features that comprise most of the variance. The 100 features that account for most of the variance in the first principal component 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 (further details on gini impurity are given in section 4.11). 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 (4.12)).[56]
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.[51] was used (https://pypi.org/project/pymrmr/).
The description above only applies to the numerical features of the CP data set. Since the ECFPs are boolean 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.[56]
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. Additionally, to validate the feature selection another prediction run was conducted using the complete feature space, i.e. all 2048 ECFP features and 1768 CP features. In this chapter, the individual modelling of ECFPs and the CP data set is presented and discussed. Furthermore, from the information obtained from the feature engineering, it is concluded why some PubChem assays perform better and why some are performing worse and if there are rules on which data sets CP might be most plausible to use with CP data. The analysis also involves annotation information obtained from biochemical knowledge, aswell as from gene ontology (GO) terms.

6.1 Comparative Analysis of ECFP and CP Predictions

The predictions were performed individually for each assay resulting in individual metrics that are compared among different descriptor sets. The first prediction run and the second prediction run are ~~being~~ compared first. They use substantially different inputs, CP and ECFP descriptors, to predict the same bioassay targets. First~~ly~~, three performance metrics are discussed. The AUC-ROC, BA and MCC. The results of both runs are presented in figure 6.1. figure 6.1a contains the AUC-ROC and b and c contain the balanced accuracy and Matthews correlation coefficient. The results from ECFP and CP prediction are plotted in the same panel for the same metric. The PubChem assays are ~~decreasingly~~ ordered by their AUC-ROC. Thus, assays which exert predictive potential with CP data are listed first. This order is kept through out this chapter.

The CP prediction run yields AUC-ROCs from around 0.4 to 0.8 as shown in figure 6.1a. They outperform the ECFP for 14 bioassays namely 720532, 651635, 624297, 2330, 743014, 588855, 743012, 624296, 743015, 1578, 504444, 651744, 1688 and 651610. However, 1688 and 651610 score an AUC-ROC below 0.7. The remaining 12 PubChem assay are henceforth

referred to as high performing assays. All other PubChem assays are referred to as low performing assays. In general, the ECFPs perform well with 27 AUC-ROC scoring higher than 0.7. In figure 6.1b the balanced accuracy is shown for all assays. The balanced accuracy is calculated by means of the specificity and the sensitivity (see equation (4.7)). The monotonously descending trend of the CP predictions is broken up in this graph. Several assays perform better or worse compared to their AUC-ROC. However, the high performing assays are still better scoring compared to the low performing assays. The ECFPs exhibit no apparent trends. Although, they do not score a balanced accuracy higher than 0.7 which is achieved by most (83%) of the high performing assays (hpa) when they are predicted using CP data.
For the CP predictions the Matthews correlation coefficient is not as consistent throughout the high performing assays (see figure 6.1c). The performances fluctuates from 0.2 to 0.6 depending on the assay. Five out of twelve achieve a Matthews correlation coefficient higher than 0.4. Within the low performing assays the CP prediction score very low. The trend is almost monotonously descreasing, mimicing the scoring of the AUC-ROC. A familiar trend is presented by the ECFPs. They score worse within the high performing assays but better within the low performing assays (lpa) given a few exemptions. None of the ECFP prediction scores a Matthews correlation coefficient higher than 0.4.

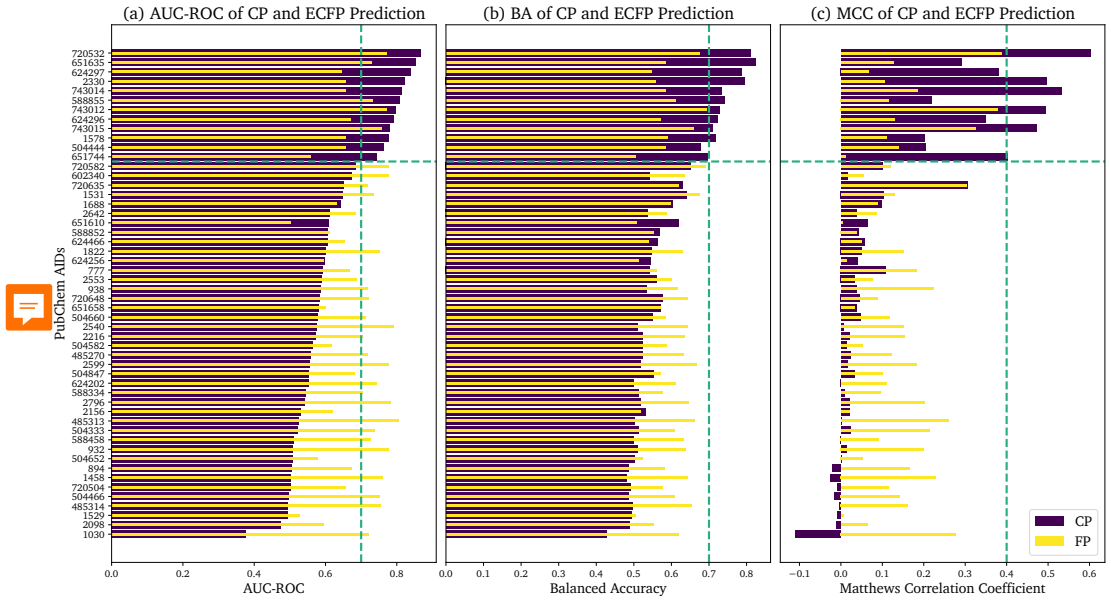


Figure 6.1: Performance comparison of CP and ECFP predictions. The AUC-ROC, balanced accuracy (BA) and Matthews correlation coefficient (MCC) ~~are being compared in three bar plots.~~ The CP prediction are shown in purple and ECFP in yellow with narrower bars. The PubChem AIDs are listed on the y-axis. Two supporting lines are drawn in each subplot. One to segregate high performing assays and low performing assays and one marks high performance for each metric.

The first two prediction runs, allowed to categorize the PubChem assays. A few performed consistently better with CP and the others exhibited better predicitve potential using the ECFPs. Especially for balanced accuracy and Matthews correlation coefficient, two metrics that are more sensitive to label imbalance, the predictivity of CP decriptors was distinguished among the hpa. Apart from the various differences these feature spaces hold, one explanation be rooted in SMOTE. This oversampling strategy is reported to perform better on numerical data, like the CP features. The ECFP features on the other hand are boolean and therefore more unsuitable.

6.2 Comparative Analysis of Modelling with Selected Features

The third prediction round concerned a mixed feature set containing CP as well as ECFP descriptors selected by methods descibed in section 5.5. This set of features will be referred to as selected features (SF). When comparing the CP and ECFP with the SF evaluation, the SF are expected to overcome the shortcomings of each identifier. By closely inspecting the differences within each metric and unveil unexpected trends further information about each identifier's shortcomings and strengths can be gained. The first panel in figure 6.2 shows the assay-wise difference between the CP and ECFP descriptors that was shown in absolute values in figure 6.1a. Again, the hpa are clearly distinguished by their scoring, compared to the lpa. This figure quantifies the difference in AUC-ROC scoring between CP and ECFP with up to 20% better results for AID 624297. On the other hand AID 1030 performes 30% worse when predicted with CP.

The comparison between SF and ECFP qualitatively exhibits a very similar trend. The hpa score very similar, on the other hand the lpa score not as low. They perform up to 20% worse, in case of AID 1030. Also some assays that have negative difference in the left panel score positive, i.e. better than ECFPs when predicted with SF (AIDs 588852, 720635 and 2156).

The range on the x-axis in figure 6.2c is the narrowest indicating more subtle changes when switching from CP to SF. The hpa present mixed differences. Some assays are better predicted with CP descriptors only (e.g. 651744) and others are better predicted with the SF (e.g. 720532). For that reason, the specific effect that the combination exerts on hpa remains inconclusive. Almost all lpa perform better when SF are used for modelling. The only exemption is AID 1688. The improvements within the other PubChem assays are as high as 20%.

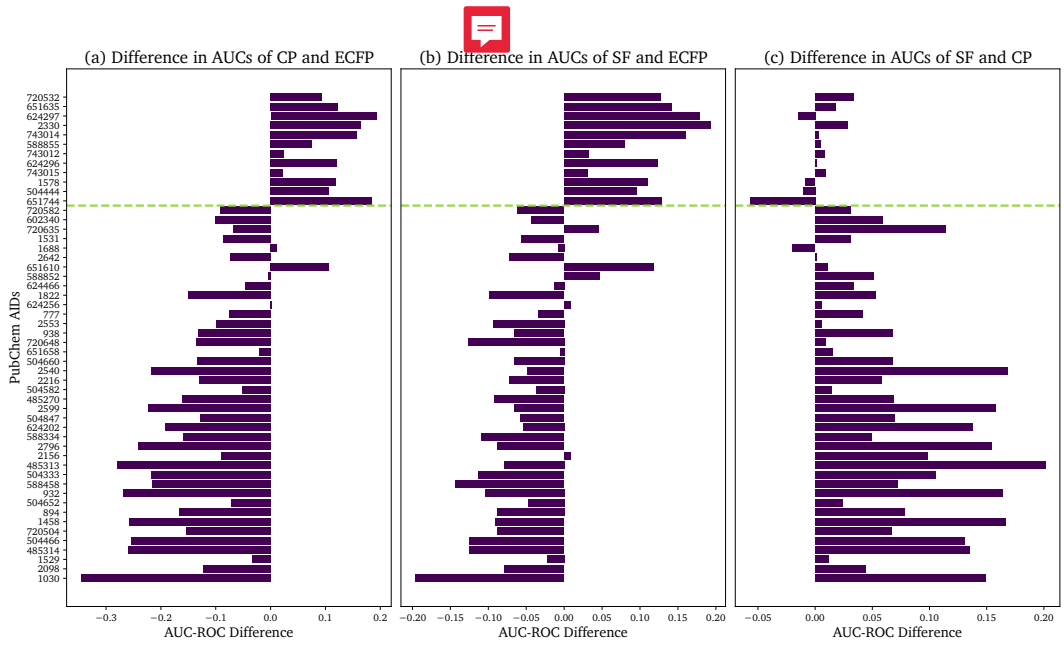


Figure 6.2: Difference in AUC between the CP, ECFP and SF for all 52 PubChem assays. The dashed green line separates hpa and lpa.

The process of features engineering fosters expectations on improvements in predictive capability and complementation of CP and ECFP descriptors. Inspecting the AUC-ROC comparison to CP data improvements are achieved within the lpa in particular, which can be attributed to the complementary information supplied by the selected ECFP features. The feature selection process described in section 5.5 does not only combine the two feature spaces, but also removes features that are expected to hinder the predictive capabilities of the ML model. However, the assumed improvement within the hpa is missing when comparing SF to CP. Indeed, the AUC-ROC of the hpa in figure 6.2c varies to a small extent that could be explained by the amount of randomness ~~that is~~ associated with RFCs. Strikingly, the improvements over the ECFP in the lpa are absent as well. This infers, that the feature selection process~~,~~ might have removed descriptors that were vital for the ECFP performance on lpa and CP performance on hpa.

To solidify the findings drawn from 6.2 further metrics are compared and analyzed. The balanced accuracy takes TNR and TPR into account and the descriptor comparison is shown in figure 6.3.

Qualitatively the balanced accuracy between CP and ECFP in figure 6.3 shows the same trend as seen before. The ECFP perform slightly worse which is also registered in figure 6.1b. The worst drop in balanced accuracy is presented by 1030 with 20%. Furthermore, seven lpa score higher balanced accuracys using CP instead of ECFP. The difference for hpa is almost identical to the AUC-ROC concerning CP and ECFP comparison.

The SF compared to the ECFP achieves consistently positive results within the hpa and mostly negative results for the lpa. Even though the scoring within lpa is slightly higher compared to the AUC-ROC with eight assays that perform better using the engineered features.
In figure 6.3c the difference for balanced accuracy between CP data and SF is shown. This panel show the highest variation compared to the AUC-ROC scoring. The superiority of **fs!** (**fs!**) within the lpa is less definitive and the hpa lean more towards the CP as descriptors of choice, too. Within the lpa there are nine assays that decrease their scoring by using SF and the increase reaches up to 10%. Then again, AID 651744 within high performing assays diminishes by 8% after feature engineering.

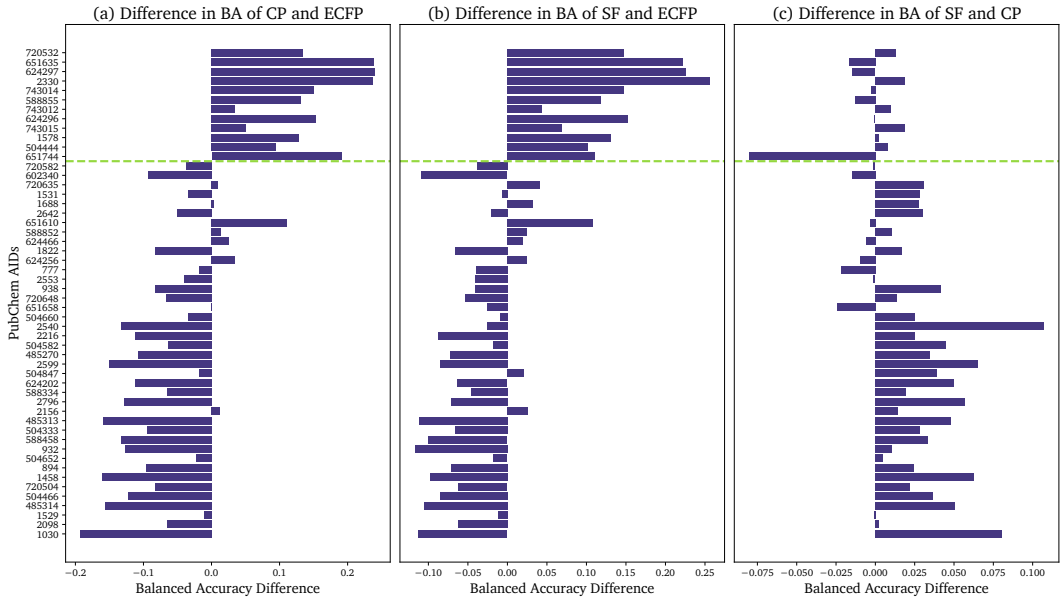


Figure 6.3: Difference in balanced accuracy between the CP, ECFP and SF for all 52 PubChem assays. The dashed green line separates hpa and lpa.

Even though very similar trends can be observed for the balanced accuracy as ~~can be seen~~ for the AUC-ROC, the findings from the balanced accuracy incentivize a stronger objection against the SF' capabilities since CP data obtains better overall scores within hpa and lpa. Also, the margin between SF and ECFP for balanced accuracy is still significantly large even if it is slightly smaller compared to the AUC-ROC comparison in figure 6.2b.
The Matthews correlation coefficient is a performance metric that is sensitive to label imbalance and that has a larger range compared to the balanced accuracy. Instead of ranging between 0 and 1 the Matthews correlation coefficient ranges between -1 and 1, therefore conveying more information about possible confusion matrix configurations. This metric is consulted to confirm the findings in the balanced accuracy comparison.
The differences between the ECFP and CP descriptors mirror figure 6.3a. The hpa in figure 6.4a

score higher with CP data opposed to lpa which score generally better with ECFPs, apart from a few exemptions that are in agreement with the exemptions found for the respective balanced accuracy comparison.

In figure 6.4 the same trends that are apparent for the balanced accuracy recur for the Matthews correlation coefficient. The ECFPs perform consistently worse for hpa and behave conversely for the low performing assays with exemption of eight assays that show a positive difference indicating better performance with SF.

The assay-wise performances in right panel of figure 6.4 behave almost identical to the balanced accuracy comparison between CP and SF. The results show alternating performances within hpa with a slight lean towards the CP descriptors. Furthermore, the SF outperform the CP on low performing assays which is in good agreement with the results from the AUC-ROC comparison and even more so with the results from the balanced accuracy.

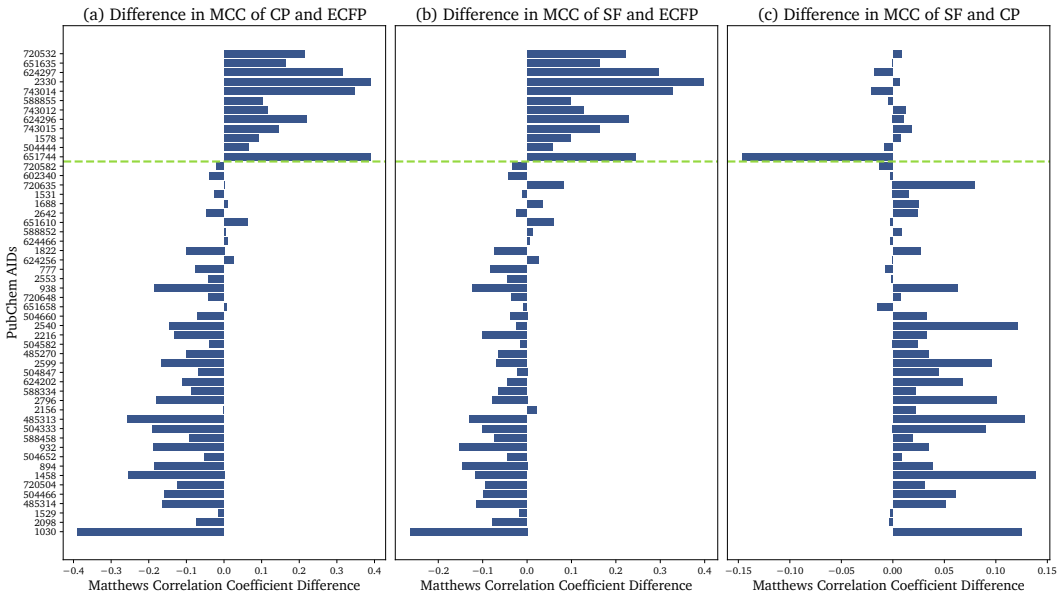


Figure 6.4: Difference in Matthews correlation coefficient between the CP, ECFP and SF for all 52 PubChem assays. The dashed green line separates hpa and lpa.

Finally, the implications from the Matthews correlation coefficient solidify the results from prior analysis. The SF are not able to distinctly outperform neither CP features nor ECFPs within their adapted niche. The reduction of noise by finding and eliminating redundant features via feature engineering is expected to elevate performance on the validation training set, even if initial descriptors perform well. Presumably, feature engineering elimiated informative features, since a performance increase within the specialized niches of CP and ECFP is not observed. Nonetheless, they achieve higher performance when all assays are taken into account

compared to CP and ECFP. Hence, each descriptor compensates for the other's weaknesses by combining the two feature spaces.

Compared to the prior metrics, the Sensitivity or true positive rate (TPR) is less complex and cannot accomplish in-depth model validation. On the other hand, TPR is more accessible when it comes to interpreting results. The TPR indicates how many samples were correctly predicted as positives. It is also a measure for the count of samples falsely predicted as negatives. ML models that have been validated as functional, can be specialized in either detecting negative samples exceptionally well or positives. Analysing the TPR with respect to the different descriptors elucidates in which category the models at hand belong. The difference in TPR between CP descriptors and ECFPs exhibit the usual trends. Within the hpa the CP data excels and ECFPs data excels within its niche of lpa. However, upon closer inspection, twelve assays are detected that score a higher TPR within the lpa using CP features. The average difference in figure 6.5a in the lpa is $-7.1\,\%$ and for hpa the average difference is $27\,\%$. Qualitatively, those trends are expected from prior performance metrics.

figure 6.5b shows very similar observations, qualitativelye and quantitatively as well. Within the lpa fourteen assay outperform ECFPs by using SF. On average the TPR difference between ECFPs and SF for lpa is $-7.6\,\%$. The SF outperform the ECFPs for all hpa but one (AID 743012). The improvement that comes with feature engineering amounts to $26.8\,\%$ on average.

Two different domains must be observed when comparing the TPR for SF and CP. Firstly, hpa perform better using CP some perform worse. The average of difference $-0.4\,\%$, i.e. in favor of CP descriptors, is standard when compared with corresponding averages (see table 6.1). On the contrary, the lpa for figure 6.5 shows a novel trend. Instead of clear predicitive predominance of the SF the TPRs for the assays are more or less balanced. The average difference TPR between SF and CP for lpa is $-0.5\,\%$. Therefore, the TPR is the only metric where SF are being outperformed by CP descriptors within lpa.

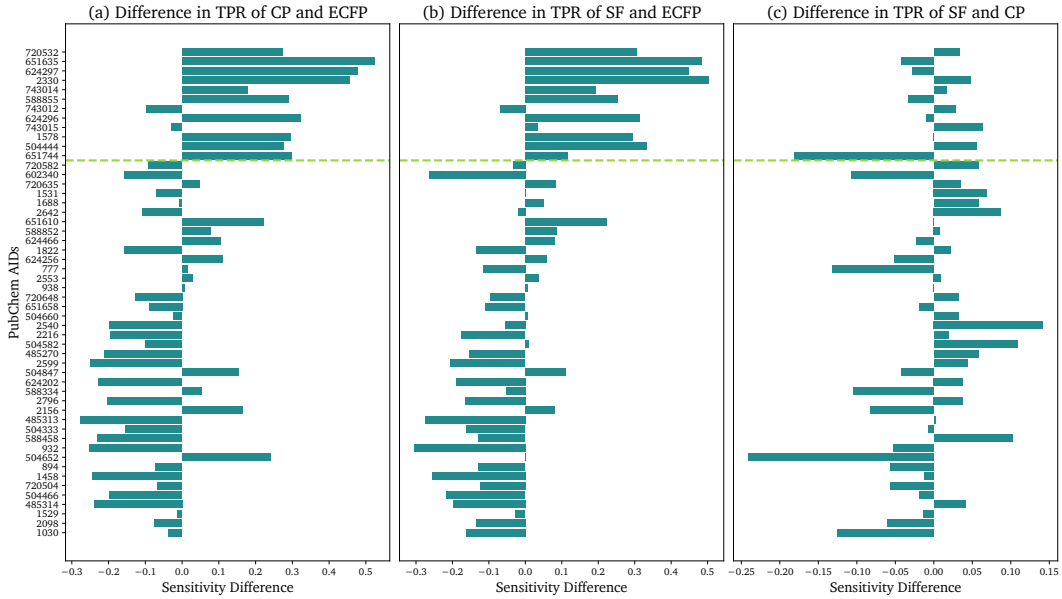


Figure 6.5: Difference in true positive rate between the CP, ECFP and SF for all 52 PubChem assays. The dashed green line separates hpa and lpa.

For low performing assays the TPR does not benefit from feature engineering. Upon feature selection and combination the TPR does not change significantly, which can be seen in figure 6.5. This could mean, that ECFPs do not contribute to the TPR when CP descriptors are applied. However, in that case CP data should outperform ECFPs within lpa which is not the case. Thus, the feature engineering is bound to be the reason for this discrepancy. ~~Upon inspection of~~ figure 6.5 it can be seen that within the hpa the TPR from CP descriptors exceeds ECFPs by up to 50 % and 27 % on average. This implies that within assays that can be well categorized by CP data, the positive samples are particularly well classified.

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 CP descriptors are less specific compared to the ECFPs within the high performing assays. Comparing CP and ECFP descriptors the high performing assays exhibits 2.5 % higher TNR on average which is vanishingly small compared to the difference in TPR (27 %) or other metrics discussed before. On average, the combined features perform 2 % better than 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 average 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.

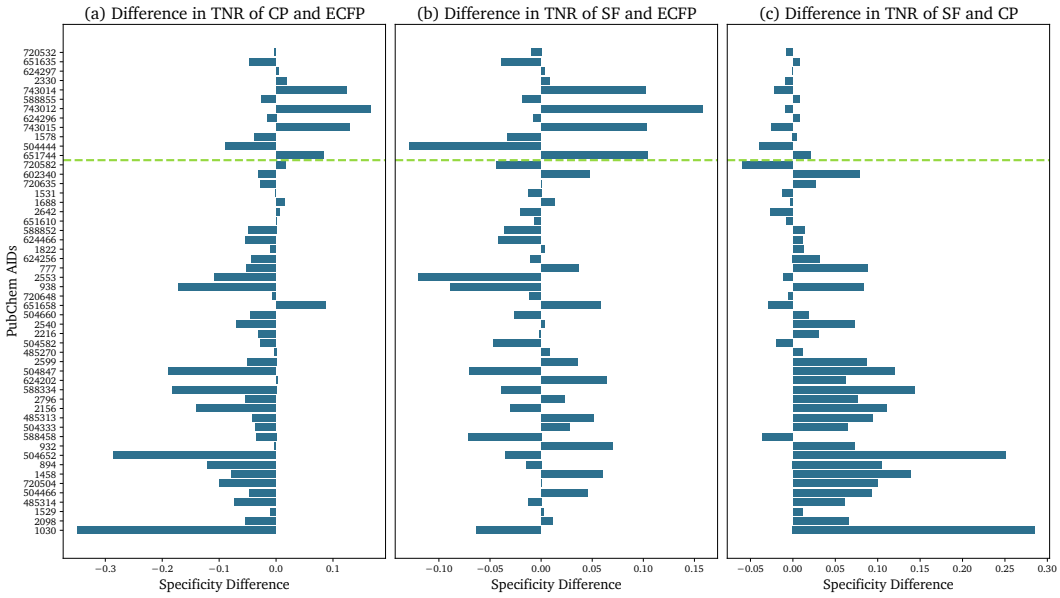


Figure 6.6: Difference in TNR between the CP, ECFP and the combined prediction run

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	0.1155	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

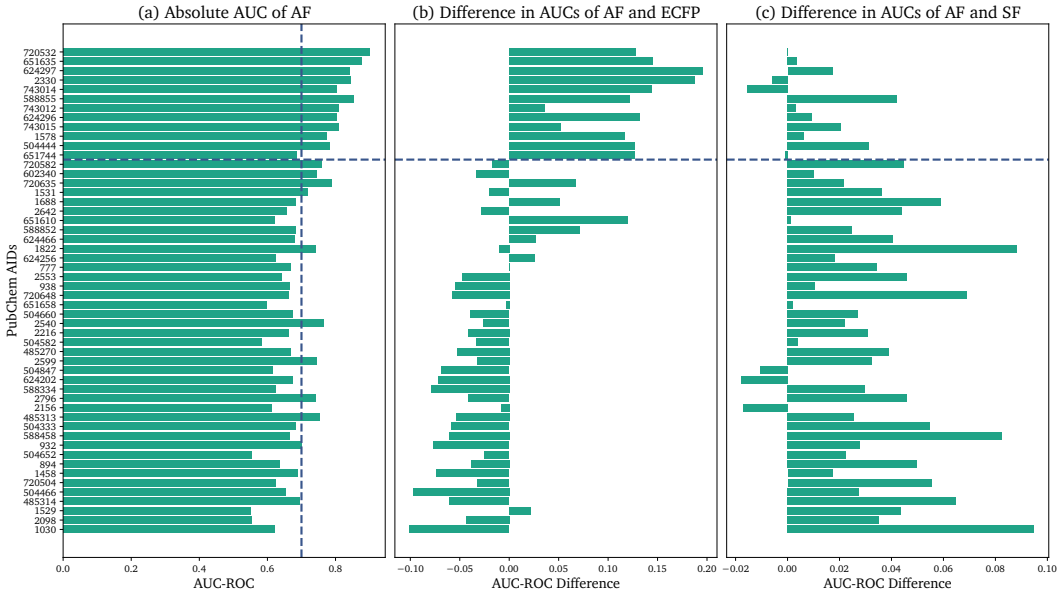


Figure 6.7: Results from the modelling with all features.

6.3 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, in this section the notation of high performing assays and low performing assays introduced in section 6.1 is still in use.

The features in the CP data set are experimentally obtained by fluorescence microscopy. The staining and image generation process described in section 4.4 utilizes six fluorescent dyes and thereby measures five different 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 specific channels should be more critical. Depending on the cellular processes that correspond to the bioassay, a particular channel could be most important. 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 ~~as well as their corresponding channels~~.
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 f_c are counted, their frequencies $\nu_c^{(a)}$ are calculated by dividing f_c by the total number of important features of the corresponding assay $N_f^{(a)}$.

$$\nu_c^{(a)} = \frac{f_c}{N_f^{(a)}} \tag{6.1}$$

Afterwards, the frequencies are normalized for each channel with respect to all bioassays for easier comparison. The normalization was performed using the standard scaler from scikit-learn's preprocessing library.[21] This scaler transforms each channel's frequencies to adopt an average of zero and a standard deviation of 1. Afterwards, every value is multiplied by 100 for easier visual interpretation, resulting in $\widetilde{\nu}_c^{(a)}$.

$$\widetilde{\nu}_c^{(a)} = \text{transform}(\nu_c^{(a)}) \cdot 100 \tag{6.2}$$

The channel-wise average standard deviation within high performing assays, $\widetilde{\sigma}_c^{\text{hpa}}$, and low performing assays, $\widetilde{\sigma}_c^{\text{lpa}}$, is calculated by the standard formula given in equation (6.3).

$$\widetilde{\sigma}_c^{\text{hpa}} = \sqrt{\frac{1}{N_{\text{hpa}}} \cdot \sum_a^{hpa} \left(\widetilde{\nu}_c^{(a)} - \langle \widetilde{\nu}_c \rangle_{\text{hpa}}\right)^2} \qquad \widetilde{\sigma}_c^{\text{lpa}} = \sqrt{\frac{1}{N_{\text{lpa}}} \cdot \sum_a^{lpa} \left(\widetilde{\nu}_c^{(a)} - \langle \widetilde{\nu}_c \rangle_{\text{lpa}}\right)^2} \tag{6.3}$$

For easier ~~quotation~~ $\widetilde{\sigma}_c^{\text{hpa}}$ and $\widetilde{\sigma}_c^{\text{lpa}}$ are referred to as the assay-group standard deviations. The results for each channel are shown graphically in figure 6.8 and also in table 6.2 with their ratio for better comparison. 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.

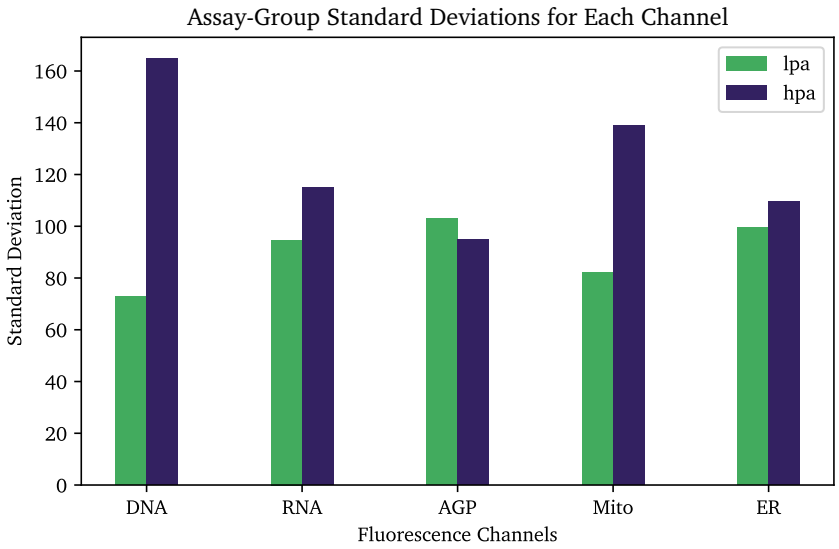


Figure 6.8: Assay-group standard deviations for low and high performing assays. For each fluorescence channel the standard deviations are compared. The low performing assays are shown in green and high performing assays are shown in purple.

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
$\widetilde{\sigma}_c^{\text{lpa}}$	72.826	94.684	103.124	82.02	99.702
$\widetilde{\sigma}_c^{\text{hpa}}$	164.78	115.08	95.022	139.055	109.517
$\widetilde{\sigma}_c^{\text{hpa}}/\widetilde{\sigma}_c^{\text{lpa}}$	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 respective assay's predictive capabilities.

6.4 In Depth Analysis of High Performing PubChem Assays

In an attempt to 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 relating bioassay endpoints to cellular morphology and cytotoxicity. The terms that could be filtered out are shown in table 6.3. The presented term is not exhaustive, but the terms refer to phenomena that are most likely visible on a cellular scale and might be modifiable by small compounds.

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/.[53]

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 figure 6.9 the PubChem assays with their annotations are shown. White squares imply that the phenotypic term is related to the corresponding PubChem assay, and a blue square is assigned if the term cannot be associated with confidence. The decision-making process is intuitive to a certain amount. 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 white square at 'Genome' and a blue square at 'C-Death' even though the two are inseparable in practice.

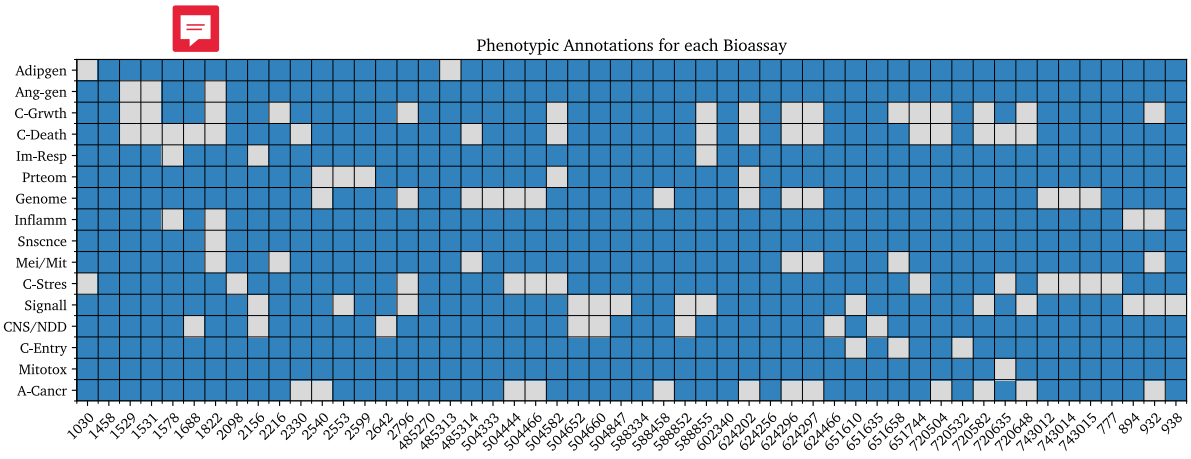


Figure 6.9: Phenotypic terms that can be associated with individual PubChem assays. These terms were manually filtered from the descriptions of the PubChem assays available at https://pubchem.ncbi.nlm.nih.gov/. White fields correspond to positive occurences and blue entries correspond to negative entries.

Most importantly, the complete annotation matrix was created before the prediction performances were recorded and therefore considered unbiased. This matrix allows testing for enriched annotations within high performing assays and low performing assays respectively. For that purpose, all annotations within each assay group are summed up to yield each term's abundances for high performing assays and low performing assays respectively. The partial abundances are divided by the total abundance of a phenotypic term to yield each assay group's frequencies. 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 6.4 the enrichment for the high performing assays and low performing assays 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.

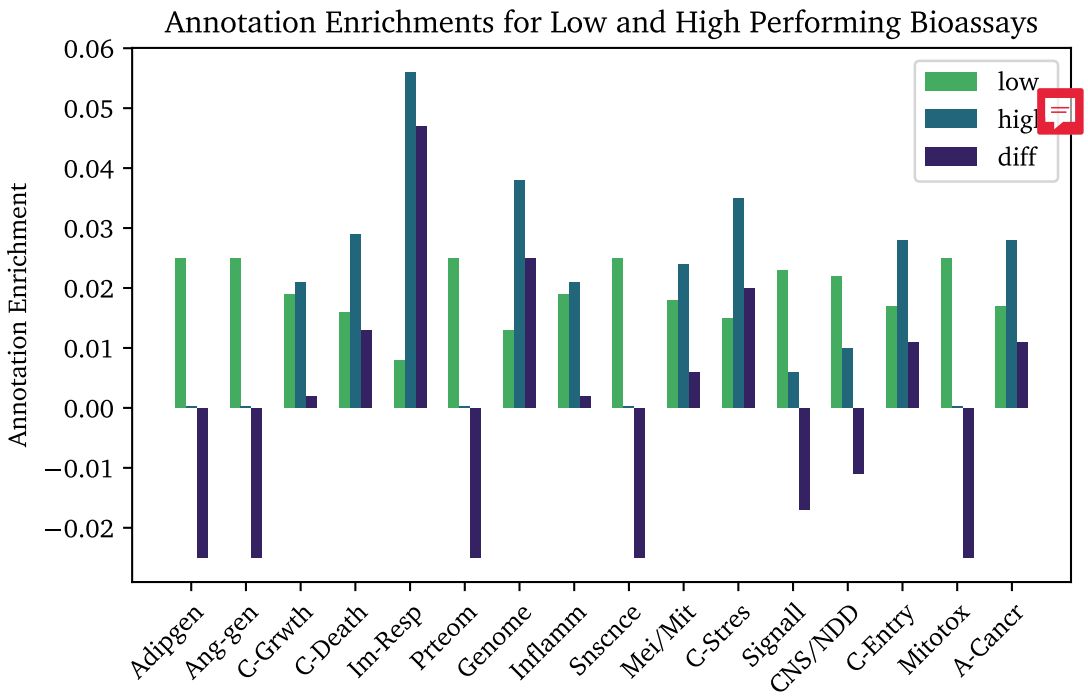


Figure 6.10: Annotations Enrichment in high performing assays and low performing assays and the difference of the two.

Table 6.4: 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 shown in table 6.2, the DNA channel has the highest ratio among all five.

7 Conclusion and Outlook

This work aimed to explore the capabilities of the CP assay by Bray et. al[3] 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.

ML machine learning

CP cell-painting

MLSMR Molecular Libraries Small Molecule Repository

MLP Molecular Libraries Program

ER endoplasmatic reticulum

WGA wheat germ agglutinin

SCC single-concentration-compound

MCC multi-concentration-compound

DMSO dimethyl sulfoxide

SMILES simplified molecular input line entry specification

AID assay identifier

CID compound identifier

InChI-key international chemical identifier key

PubChem PubChem

MBS Metadata_broad_sample

prid preprocessed raw image data

cmrds combined ML-ready data set

ECFP extended-connectivity fingerprint

ECFP-set ECFP-set

RFC random forest classifier

CV cross-validation

KFCV k-fold cross validation

TPR true positive rate

TNR true negative rate

BA balanced accuracy

MCC Matthews correlation coefficient

AUC-ROC area under the ROC curve

ROC-curve receiver operating characteristic curve

TP true positive

TN true negative

FP false positive

FN false negative

FPR false positive rate

PCA principal component analysis

MRMR minimal-redundancy-maximal-relevance criterion

gi gini impurity

SMOTE synthetic minority oversampling technique

hpa high performing assays

lpa low performing assays

VSV vesicular stomatitis virus

ATXN2 Ataxin-2 gene

SCA2 spinocerebellar ataxia type 2

MoA mechanism of action

GCR glucocorticoid receptor

HTS high-throughput-screening

IUPAC International Union of Pure and Applied Chemistry

GO gene ontology

SF selected features

Bibliography

[1] Singh, S.; Khanna, V. K.; Pant, A. B. *In Vitro Toxicology*; Elsevier, 2018; pp 1–19.

[2] 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.

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

[4] 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.

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

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

[7] 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.

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

[9] 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.

[10] 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*.

[11] 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.

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

[13] 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.

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

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

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

[17] Subramanian, A. et al. *Cell* **2017**, *171*, 1437–1452.e17.

[18] Simm, J. et al. *Cell Chemical Biology* **2018**, *25*, 611–618.e3.

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

[20] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. *Nucleic Acids Research* **2020**, *49*, D1388–D1395.

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

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

[23] *Pure and Applied Chemistry*; De Gruyter, 2014.

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

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

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

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

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

[29] 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.

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

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Kolokoltsov, A. A.; Deniger, D.; Fleming, E. H.; Roberts, N. J.; Karpilow, J. M.; Davey, R. A. *Journal of Virology* **2007**, *81*, 7786–7800.

[37] Hofmann-Winkler, H.; Kaup, F.; Pöhlmann, S. *Viruses* **2012**, *4*, 3336–3362.

[38] Kolokoltsov, A. A.; Saeed, M. F.; Freiberg, A. N.; Holbrook, M. R.; Davey, R. A. *Drug Development Research* **2009**, *70*, 255–265.

[39] National Center for Biotechnology Information (2021), A. PubChem Bioassay Record for AID 720532. Source: National Center for Advancing Translational Sciences (NCATS), 2021; https://pubchem.ncbi.nlm.nih.gov/bioassay/720532.

[40] Pulst, S.-M. et al. *Nature Genetics* **1996**, *14*, 269–276.

[41] for Biotechnology Information, N. C. PubChem Bioassay Record for AID 651635, qHTS for Inhibitors of ATXN expression. Source: National Center for Advancing Translational Sciences (NCATS), 2012; https://pubchem.ncbi.nlm.nih.gov/bioassay/651635, Last accessed Mar. 8, 2021.

[42] Biedler, J. L.; Helson, L.; Spengler, B. A. *Cancer research* **1973**, *33*, 2643–2652.

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

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

[45] 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.

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

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

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

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

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

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

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

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

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

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

[56] 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.

8 Appendix

Table 8.1: Lists of GO terms more abundant in each assay group.

GO-Terms More Abundant in Low Performing Assays	
'cellular_response_to_stimulus'	'anterograde_trans-synaptic_signaling'
'organic_cyclic_compound_binding'	'nervous_system_process'
'heterocyclic_compound_binding'	'cation_transmembrane_transport'
'DNA_binding'	'cellular_response_to_nitrogen_compound'
'cellular_protein_modification_process'	'cellular_response_to_oxygen-containing_compound'
'transferase_activity'	'trans-synaptic_signaling'
'macromolecule_modification'	'connected_anatomical_structure'
'catalytic_activity'	'ion_transmembrane_transport'
'organonitrogen_compound_metabolic_process'	'regulation_of_transcription_by_RNA_polymerase_II'
'protein_metabolic_process'	'postsynaptic_neurotransmitter_receptor_activity'
'signal_transduction'	'response_to_nitrogen_compound'
'catalytic_activity_acting_on_a_protein'	'inorganic_cation_transmembrane_transport'
'cell_communication'	'inorganic_ion_transmembrane_transport'
'DNA-binding_transcription_factor_activity'	'cell_surface_receptor_signaling_pathway_involved_in_cell-cell_signaling'
'ion_binding'	'system_process'
'cellular_protein_metabolic_process'	'postsynaptic_membrane'
'protein_phosphopantetheinylation'	'passive_transmembrane_transporter_activity'
'nucleic_acid_binding'	'synaptic_membrane'
'transcription_regulator_activity'	'response_to_organonitrogen_compound'
'signaling'	'chromatin'
'protein_modification_process'	'response_to_dopamine'
'cellular_component'	'organic_substance_transport'
'G_protein-coupled_receptor_signaling_pathway'	'catalytic_activity_acting_on_DNA'
'G_protein-coupled_receptor_activity'	'response_to_catecholamine'
'molecular_transducer_activity'	'cellular_response_to_catecholamine_stimulus'
'signaling_receptor_activity'	'non-membrane-bounded_organelle'
'establishment_of_localization'	'metalloendopeptidase_activity'
'transmembrane_signaling_receptor_activity'	'response_to_organic_cyclic_compound'
'hydrolase_activity'	'cysteine-type_peptidase_activity'
'ion_transport'	'adenylate_cyclase-modulating_G_protein-coupled_receptor_signaling_pathway'
'peptidase_activity'	'metallopeptidase_activity'
'metal_ion_binding 'cation_binding'	'DNA_repair'
'proteolysis'	'potassium_channel_activity'
'transition_metal_ion_binding'	'cellular_response_to_monoamine_stimulus'
'intracellular_anatomical_structure'	'DNA-binding_transcription_factor_activity_RNA_polymerase_II-specific'
'zinc_ion_binding'	'dopamine_neurotransmitter_receptor_activity'
'transporter_activity'	'response_to_monoamine'
'cell_periphery'	'cellular_response_to_dopamine'
'response_to_oxygen-containing_compound'	'gated_channel_activity'
'chemical_synaptic_transmission'	'chromosome'
'transmembrane_transporter_activity'	'voltage-gated_channel_activity'
'multicellular_organismal_process'	'adenylate_cyclase-activating_G_protein-coupled_receptor_signaling_pathway'
'cation_transport'	'organelle'
'inorganic_cation_transmembrane_transporter_activity'	'sequence-specific_DNA_binding'
'plasma_membrane_region'	'cellular_response_to_DNA_damage_stimulus'
'oxidoreductase_activity'	'cellular_response_to_organic_cyclic_compound'
'ion_channel_activity'	'protein_dimerization_activity'
'ion_transmembrane_transporter_activity'	'intracellular_organelle'
'plasma_membrane'	'potassium_ion_transmembrane_transport'
'synaptic_signaling'	'voltage-gated_potassium_channel_activity'
'inorganic_molecular_entity_transmembrane_transporter_activity'	'macromolecule_localization'
'cell_junction'	'synaptic_transmission_dopaminergic'
'channel_activity'	'cellular_response_to_organonitrogen_compound'
'neurotransmitter_receptor_activity'	'voltage-gated_ion_channel_activity'
'cation_transmembrane_transporter_activity'	'potassium_ion_transmembrane_transporter_activity'
'membrane'	'potassium_ion_transport'
'metal_ion_transport'	'dopamine_receptor_signaling_pathway'
'cation_channel_activity'	'regulation_of_molecular_function'
'metal_ion_transmembrane_transporter_activity'	'endopeptidase_activity'
'DNA_metabolic_process'	'intracellular_non-membrane-bounded_organelle'
'synapse'	'voltage-gated_cation_channel_activity'
'transmembrane_transport'	'anion_transport'
'transcription_by_RNA_polymerase_II'	'regulation_of_biological_quality'
'cell-cell_signaling'	'G_protein-coupled_amine_receptor_activity'
'chemical_synaptic_transmission_postsynaptic'	
'postsynapse'	

GO-Terms More Abundant in High Performing Assays	
'cellular_process'	'cellular_response_to_chemical_stimulus'
'cellular_macromolecule_metabolic_process'	'negative_regulation_of_DNA_replication'
'regulation_of_cellular_process'	'regulation_of_cell_cycle'
'regulation_of_cellular_biosynthetic_process'	'negative_regulation_of_macromolecule_metabolic_process'
'regulation_of_cellular_macromolecule_biosynthetic_process'	'regulation_of_nucleobase-containing_compound_metabolic_process'
'cellular_macromolecule_biosynthetic_process'	'protein_binding'
'biosynthetic_process'	'aromatic_compound_biosynthetic_process'
'regulation_of_cellular_metabolic_process'	'cellular_nitrogen_compound_metabolic_process'
'macromolecule_biosynthetic_process'	'regulation_of_transcription_DNA-templated'
'organic_substance_biosynthetic_process'	'DNA_replication'
'regulation_of_biosynthetic_process'	'cellular_aromatic_compound_metabolic_process'
'regulation_of_macromolecule_biosynthetic_process'	'cellular_nitrogen_compound_biosynthetic_process'
'cellular_biosynthetic_process'	'negative_regulation_of_cellular_biosynthetic_process'
'regulation_of_macromolecule_metabolic_process'	'regulation_of_nitrogen_compound_metabolic_process'
'nucleobase-containing_compound_biosynthetic_process'	'negative_regulation_of_macromolecule_biosynthetic_process'
'organic_cyclic_compound_metabolic_process'	'nucleobase-containing_compound_metabolic_process'
'negative_regulation_of_cellular_process'	'RNA_metabolic_process'
'regulation_of_primary_metabolic_process'	'heterocycle_metabolic_process'
'RNA_biosynthetic_process'	'nucleic_acid-templated_transcription'
'regulation_of_nucleic_acid-templated_transcription'	'negative_regulation_of_cell_cycle'
'negative_regulation_of_cellular_metabolic_process'	'organic_cyclic_compound_biosynthetic_process'
'regulation_of_DNA_replication'	'heterocycle_biosynthetic_process'
'negative_regulation_of_cellular_macromolecule_biosynthetic_process'	'regulation_of_RNA_biosynthetic_process'
'negative_regulation_of_metabolic_process'	'transcription_DNA-templated'
'negative_regulation_of_biological_process'	'regulation_of_RNA_metabolic_process'
'gene_expression'	'negative_regulation_of_biosynthetic_process'
'cell_cycle'	'regulation_of_gene_expression'
'nucleic_acid_metabolic_process'	

List of Figures

List of Tables