SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

REGISTRATION NUMBER: FCHPT-5414-91815

DESIGN, IMPLEMENTATION, AND OPTIMIZATION OF CLASSIFICATION ALGORITHM FOR IDENTIFICATION OF SMALL MOLECULES USING ANNOTATED SPECTRAL TREES

DIPLOMA THESIS

2022 Bc. Marek Wadinger

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

REGISTRATION NUMBER: FCHPT-5414-91815

DESIGN, IMPLEMENTATION, AND OPTIMIZATION OF CLASSIFICATION ALGORITHM FOR IDENTIFICATION OF SMALL MOLECULES USING ANNOTATED SPECTRAL TREES

DIPLOMA THESIS

Study programme:	Automation and Information Engineering in Chemistry and Food Industry
Study field:	Cybernetics
Workplace:	Department of Information Engineering and Process Control (FCHPT STU), Thermo Fisher Bratislava
Thesis supervisor:	prof. Ing. Michal Kvasnica, PhD.
Consultant:	Ing. Michal Raab

2022 Bc. Marek Wadinger

Slovak University of Technology in Bratislava
Department of Information Engineering and Process Control

Faculty of Chemical and Food Technology
Academic year: 2021/2022
Reg. No.: FCHPT-5414-91815

STU FCHPT

MASTER THESIS TOPIC

Student:	**Bc. Marek Wadinger**
Student's ID:	91815
Study programme:	Automation and Information Engineering in Chemistry and Food Industry
Study field:	Cybernetics
Thesis supervisor:	prof. Ing. Michal Kvasnica, PhD.
Head of department:	Ing. Martin Klaučo, PhD.
Workplace:	Department of Information Engineering and Process Control (FCHPT STU), Thermo Fisher Bratislava

Topic: **Design, implementation, and optimization of classification algorithm for identification of small molecules using annotated spectral trees**

Language of thesis: English

Specification of Assignment:

Main aim of this master's thesis is design, implementation, and optimization of the classification algorithm for identification of small molecules using annotated spectral trees. Probability-based searching and scoring classification algorithm that assigns confidence score, is primarily developed to reduce the probability of a false positive identification of a chemical compound. In essence, it specifies a likelihood of correct compound identification calculated from a Bayesian network trained on mzCloud reference database. The process of training involves continuous comparison of the compound against the rest of the library and extraction of the specifications of the matches. The parameters extracted from the spectral pairs library are, for example, spectral similarity (expressed as spectral match value), relative difference in collision energies, polarity of the spectra, number of peaks, relative energy of the query spectra and whether compounds that produced these spectra are identical. Data from over 100M such compounds were collected in the database, available for calculating the coefficients of the Bayesian network. The final scoring is calculated as a likelihood of two spectra belonging to the same compound, given the observed spectral match, collision energy of the unknown, the number of peaks in the spectra, polarity, and the difference in the relative collision energies, etc. The performance of the developed algorithm is verified through selectivity and sensitivity on the provided data set using the classification metrics (ROC, AUC, etc.).

Length of thesis: 50

Deadline for submission of Master thesis: 15. 05. 2022

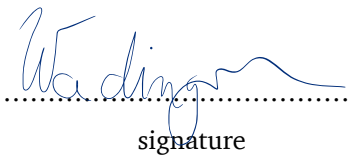
Approval of assignment of Master thesis: 03. 03. 2022

Assignment of Master thesis approved by: prof. Ing. Miroslav Fikar, DrSc. – Study programme supervisor

Honour Declaration

I declare that the submitted diploma thesis was completed on my own, in cooperation with my supervisor, with the help of professional literature and other information sources, which are cited in my thesis in the reference section.

As the author of my diploma thesis, I declare that I did not break any third-party copyrights.



signature

Acknowledgment

I would like to express my gratitude to my teacher and thesis supervisor, prof. Ing. Michal Kvasnica, PhD., who sparked my interest in cutting-edge automation and information solutions and gave me numerous opportunities to challenge my critical thinking and broaden my horizons.
Secondly, I would like to thank Ing. Michal Raab for sharing his deep insight into the domain and his sincere interest in improving my experience from the collaboration with Thermo Fisher Scientific. Following that, I want to thank my colleagues Ingrid and Maria.
Thirdly, I would also like to thank my parents, who supported me in achieving my academic goals.

Abstract

The main aim of this master's thesis is design, implementation, and optimization of the classification algorithm for the identification of small molecules using annotated spectral trees. Probability-based searching and scoring classification algorithm that assigns a confidence score is primarily developed to reduce the probability of a false-positive identification of a chemical compound. In essence, it specifies a likelihood of correct compound identification calculated from a Bayesian network trained on the mzCloud reference database. The training process involves continuous comparison of the compound against the rest of the library and extraction of the specifications of the matches. The parameters extracted from the spectral pairs library are, for example, spectral similarity (expressed as spectral match value), the relative difference in collision energies, the polarity of the spectra, number of peaks, relative energy of the query spectra, and whether compounds that are produced these spectra are identical. Data from over 100M such compounds were collected in the database, available for calculating the coefficients of the Bayesian network. The final scoring is calculated as a likelihood of two spectra belonging to the same compound, given the observed spectral match, collision energy of the unknown, the number of peaks in the spectra, polarity, and the difference in the relative collision energies, etc. The performance of the developed algorithm is verified through selectivity and sensitivity on the provided data set using the classification metrics (ROC, AUC, etc.).

Keywords: mass spectrometry, machine learning, bayesian network, identification, algorithm, validation of similarity scoring, filtering false positives, reducing false detection rate, library search identification confidence

Abstrakt

Cieľom diplomovej práce je návrh, implementácia a optimalizácia klasifikačného algoritmu na identifikáciu malých molekúl použitím anotovaných spektrálnych stromov. Vyhľadávací a vyhodnocovací klasifikačný algoritmus, založený na pravdepodobnosti, prirad'uje skóre spoľahlivosti určeniu a je vyvinutý primárne na redukovanie pravdepodobnosti falošne pozitívnej identifikácie chemickej látky. Základom algoritmu je stanovenie vierohodnosti správnej identifikácie látky vypočítanej Bayesovskou sieť ou trénovanou na referenčnej databáze mzCloud. Proces trénovania zahŕňa nepretržité porovnávanie látky voči zvyšku knižnice a extrahovanie parametrov zhodných látok. Parametre extrahované z knižnice spektrálnych párov sú napríklad spektrálna podobnosť (vyjadrená hodnotou spektrálnej zhody), relatívny rozdiel kolíznych energií, polarita spektra, počet píkov, relatívna energia vyžiadaného spektra a to, či sú látky tvoriace tieto spektrá identické. Dáta pre viac ako 100 mil. takýchto látok sú uložené v databáze, dostupné na výpočet koeficientov Bayesovskej siete. Finálne skóre spoľahlivosti je vypočítané ako pravdepodobnosť príslušnosti dvoch spektier tej istej látke, vzhľadom na pozorovanú spektrálnu zhodu, kolíznu energiu neznámej látky, počet píkov v spektre, polaritu, rozdiel medzi relatívnymi kolíznymi energiami a pod. Výkon vyvinutého algoritmu sa overí prostredníctvom selektivity a citlivosti na poskytnutom dátovom súbore využitím klasifikačných metrík (ROC, AUC a pod.).

Kľúčové slová: hmotnostná spektrometria, strojové učenie, bayesovské siete, identifikácia, algoritmus, validácia skóre podobnosti spektier, filtrovanie falošne pozitívnych výsledkov, zníženie miery falošných detekcií, spoľahlivosť vyhľadávania v spektrálnej knižnici

Contents

List of Figures

thresholds. Scores are given by; cosine similarity scoring (red), random scoring (dark blue), Random Forest re-scoring (light blue), Random Forest re-scoring on categorical features (green), Bayesian Network with computed structure (yellow), and Bayesian Network with manually selected DAG (orange). 81

List of Tables

1. Introduction

1.1. Motivation

Countless areas of chemistry, biology, and pharmacology depend on the identification and characterization of small molecules. Whether it is product quality or its obedience to regulations, the environmental impact of fertilizers, metabolism of drugs, recognition of harmful substances, reconstruction of reaction pathways, or groundbreaking research, the successful identification is subject to precise measuring tools and state-of-art algorithms. Mass spectrometry and its derivatives proved to be a central technology of this interest. The capabilities of modern devices to craft high-resolution spectra are a fundamental ingredient to success in chemical analysis. Nowadays, cloud libraries contain millions of spectra obtained by mass spectrometry experiments. Carefully curated, they serve for search and comparison with the queried analyte. Current searching algorithms can swiftly compare thousands of pairs returning an ordered list of candidate compounds with the highest confidence of true match between query and library spectra based on their similarity. Nevertheless, this approach faces challenges resulting from its very nature since only one molecule in the list of candidates can be a match. In addition, the library cannot guarantee the presence of analyzed compounds within, which commonly results in false-positive identification. This problem has further implications based on knowledge about possible constituents of the analyzed sample. Molecules might be known to us but not to the library, unknown for us but present in the library, or unknown in both cases. Above all, at the intersection lies a vital question of other factors influencing the possibility of true-positive identification. Therefore, the analysts must manually explore the candidate's list from the library search.

The type of the device and experimental conditions are crucial aspects to consider when matching the library spectra with the query sample. However, to filter library spectra based on matching conditions with the query as a requirement, the library database would have to contain billions of combinations of the precursor and experimental conditions, which is not the case. Therefore, to minimize the probability of false-positive identification, the deviations in the conditions shall be explored. Experimental conditions are stored as metadata and can be readily used to filter potential candidates based on the classification of true and false positives on annotated data. All classification algorithms based on machine learning are capable of such tasks. The trained algorithm returns the model with the capability to find patterns in metadata and forge a new ranking system that will give further

support or diminish the reliance on candidates. The bottleneck of such a data-driven approach is that such models are usually not based on causal reasoning, unlike similarity scoring algorithms. Therefore, the quality and diversity of training data, hypotheses testing, and thorough evaluation determine the success and credibility of an algorithm.

Bayesian Networks represent directed acyclic graphs, where relationships between features symbolized by edges show dependencies that best describe contributing factors to our decisions. The Bayesian network mimics conditional dependencies between variables. With the capability to build itself, train its parameters, and make inferences of true-positive and false-positive likeliness, it offers the same framework as machine learning algorithms. Based on the evidence given as a feature list of experimental conditions, it can update the probability of correct sample identification. The building of the network is analogous to the training of any classification model in general, yet it gives a tremendous benefit of a complete explanation of its decisions. Interpretability introduces a competitive advantage in the model's credibility over other models. However, extensive data preprocessing, such as continuous feature discretization, must take place to work with Bayesian Network. In general, this process reduces the information content of the features and raises the need for a systematic way to discretize variables.

2. Current state of the research

Identification of molecules is the fundamental part of analytical chemistry and the central topic of scientific advancement in the field. Among all analytical techniques, mass spectrometry and its derivatives proved to be a state-of-art approach for analysis of the content of a chemical sample. From one stage of the mass spectrometry analysis up to the complex pipelines of combined forces of various chromatography techniques and mass spectrometry analysis in tandem resulting in fragmentation trees, our aim to maximize the probability of correct identification is straightforward. The more accurate the detection, the better our chances of differentiating the molecules.

The most straightforward way to identify a query spectrum is to assess all combinations of atoms and construct a molecule that gives the same Daltonic mass as the molecule in the sample or within the range specified by the device's mass error. Theoretically, this approach allows unknown-unknowns (*de novo*) analysis. Practically, combinatorics gets computationally unreasonable for larger molecules. Moreover, if possibly constituting atoms are not inferred by an analyst, the number of combinations is tremendous, and the matching molecule is impossible to select.

In the last decade, many research groups published papers in the computational mass spectrometry domain focusing on automated methods development for processing, visualization, analysis, and identification of molecules [1, 2, 3, 4]. These methods aim to approach the identification of molecules systematically.

Spectral library searching is *a modus operandi* for molecule identification. This method uses a reference library of spectra obtained by *in vitro* analysis of samples. The tremendous growth of spectral databases in the last decade made many molecules available for identification. However, the lack of many metabolites in such databases led to many *in silico* methods development. We will discuss two families of such methods relevant as a reference for our thesis.

The strategies that predict candidate spectra computationally from InChI or Smiles are called compound-to-MS matching (C2MS). These methods, in general, are rule-based algorithms that employ a library of chemical structure identifiers for MS spectra prediction [1]. This process results in a list of predicted candidates with similar MS. Amidst the most

cited C2MS algorithms are CFM-ID, approaching identification using combinatorial fragmentation and neural network. The version CFM-ID 3.0 introduced the idea of metadata usage to enhance identification capabilities, which in the works of authors, significantly improved the prediction accuracy [2]. Version 4.0 introduced in the 2021 year further increased the prediction F1 score of 0.4253 for the Metlin 2015 database (compared to an F1 score of 0.3269 for version 3.0) and F1 score of 0.3110 for the Metlin 2015 database (compared to and F1 score of 0.2796 for version 3,0). [1] "All experiments were carried out under a 10-fold cross-validation framework. " [1] Another C2MS algorithm is Mass Frontier (HighChem Ltd., Bratislava, Slovakia), which uses a "set of general ionization, fragmentation, and rearrangement rules" given by observation of experimentally studied reaction pathways to generate possible spectra. Ruttkies et al. in their study report, augmentation of MetFrag algorithm version 2.2 with the capability of metadata usage [3].

Another family of methods is called MS-to-compound matching (MS2C), which elucidates characteristics of the queried sample as unique identifiers and compares them with the library of compounds. Among the most cited is the CSI: FingerID. The method builds on fragmentation tree computation and machine learning. First, the MS/MS peaks are annotated with formulas of the respective fragments and connected according to assumed losses. A Support Vector Machine (SVM) predicts the molecular fingerprints and estimates the probabilities. Next, the algorithm performs a fingerprint comparison with structures in the database. The number of correct identifications (i.e., true-positive samples found in the first place of the output list of candidates) reached 31.8% on the PubChem database [4]. The Supporting information document sheds light on the details of the experiment, data curation, and baseline rate, which is the identification rate of randomly ordered candidates. In 2018 Bayesian network was used to model dependencies of CSI: FingerID to offer statistical interpretability and a better identification rate, with a reported improvement of 2.85% [5]. CANOPUS algorithm based on a deep neural network developed in 2021 promises better performance over CSI: FingerID without the need for spectral or structural reference data. However, this method aims only for molecular class identification [7].

A review of software tools from 2018 provides a broader overview of the leading techniques [8]. Therefore, we will not discuss them further in our thesis.

This thesis section will discuss innovative approaches to library searching and scoring algorithms, which are central to our focus.

In 2007, Käll et al. introduced the Percolator. This algorithm, used as a software post-processor to alter sensitivity, is based on semi-supervised machine learning to discriminate between correct and decoy spectrum identifications. However, it is impossible to construct decoy spectra for non-peptides (e.g., small molecules spectra), which restricts the method only for peptide identification [9].
In 2021 two game-changing methods were published.
Spec2Vec, an unsupervised machine learning approach for the computation of spectrum similarities. The basis of the algorithm builds on learned relationships between peaks across large training datasets. The algorithm employs machine learning techniques common to natural language processing. In this study, Huber et al. neatly presents the performance comparison against cosine similarity score, showing improved true-to-false-positive ratios and accuracy vs. retrieval graph in library matching for various similarity thresholds [10]. Moreover, the method produced scores that correlate with structural similarity more strongly than cosine scoring.
The MS2DeepScore represents a novel similarity assessment mechanism based on **unsupervised** deep learning. The tool predicts Tanimoto scores for query-library pairs based on peak m/z positions and intensities without further spectrum information. The algorithm shows its prediction through the ability to infer structural similarity directly from MS/MS spectrum. Unlike the Percolator, this approach is independent of spectral quality and the origin in structural similarity comparison. As the authors denoted in the discussion, the neural network lacks the possibility of results interpretation, and accuracy can change for various training sets. However, combining the scores for the same molecule and setting a threshold for prediction uncertainty reduced the prediction error. Huber et al. underline the ability to predict structural similarity by comparing the approach with the state-of-the-art techniques Spec2Vac and modified Cosine scoring [11].

From the previous paragraph, we shall keep in mind that an automated way to match query and library samples is to use a similarity scoring algorithm. Its purpose is to evaluate the spectral match of the combination of the two. Confidence in the correct identification is a crucial parameter of such an algorithm [12]. The quality and resolution of the spectra and the similarity scoring function affect the confidence. Kind et al . discuss the experimental conditions with critical influence on overconfidence of the prediction [13]. A usual result of library matching in such a process is a list of library matches. Some papers focus on the prediction of whether the queried molecule is in the list of hits or whether it is present in the library database at all (requires reference). We will not cover those in our thesis. Our focus will be mainly on false positive rate reduction. The following studies provide deeper insight

into the topic [12, 14, 15]. Works emphasizing identification confidence of library search in proteomics include algorithms like INFERYS, Calibr, pMatch, and SEQUEST [16, 17, 18].

3. Thesis Goal

The importance of the correct molecule identification is central to countless fields of chemistry and biology. Mass spectrometry is the leading tool for sample analysis. The classification of molecules in the sample is usually performed automatically by comparison with the reference spectra library or computed spectra for known molecular structures. Although many spectra for countless molecules and various experimental conditions are not present in the libraries yet, library search has robust performance and the highest accuracy compared with the alternative methods. In the 21st century, data collection is a rapid and inevitable process that favors the library searching approach and makes it as exigent as ever.

First, we review and analyze current advancements in small-molecule identification and list the state-of-art algorithms that tackle this problem using both *in vitro* and *in silico* experimental techniques.

Second, we introduce a novel classification algorithm for small molecule identification. The developed algorithm uses an annotated list of query-library spectrum matches based on rigorous dot product score on the input. The algorithm's output is a new confidence score for each sample in the dataset based on pertained data-based machine learning model. A confidence score serves as evidence to reduce the probability of false-positive identification of chemical compounds based on consideration of deviation between experimental parameters. Further, we provide in-depth information on each step of the algorithm development based on the data sciences life cycle *CRISP-DM* framework.

Third, we confront the problem with the interpretability of the data-based machine learning model and present probability-based searching and scoring algorithm based on the Bayesian Network.

Lastly, we verify the performance of the developed algorithm through selectivity and sensitivity in the provided data set using the classification metrics (e.g., ROC, AUC, F1 score). Lastly, we test the hypotheses to support our statements.

4. Theoretical Background

This chapter delivers theoretical principles behind readily employed techniques in our work and the instruments used to obtain data. The concepts drawn in this section shall be sufficient for a comprehensive understanding of the practical aspect of this thesis among the whole academic community, regardless of the major.

4.1. Principle of Mass Spectrometry

Mass spectrometry over the last century represented superlative to analytical methods with its unrivaled sensitivity and low detection limits. Rapid progression of technology, the instruments innovation, and the combination of various analyzers made it especially hard to generalize. Yet the main principle is best described by four stages drawn in the following lines [20].

4.1.1. Ion Acquisition - Ionization

A first step is to produce ions from the sample using a beam of electrons in **the ion source**, which alters its charge and possibly results in fragmentation. Ions yield information about the nature and structure of their precursor molecule. The selection of ionization techniques may vary. The character of the analyte, e.g., **the polarity** of the solution and proton **affinity** in the gas phase, and the experimenter's intentions are only two factors to consider [23]. Two families of ionization techniques are discussed based on internal energy transferred:

- **Hard ionization** - resulting in extensive fragmentation due to a high quantity of residual energy:
    1. Electron ionization (EI) - fragmentation results from interaction with electron [24].
- **Soft ionization** - resulting in little fragmentation given little residual energy:
    2. Electrospray ionization (ESI) - the strong electric field's application on liquid samples to produce droplets. Few or no productions - identify the molecular mass of the analytes [20];
    3. Desorption Electrospray Ionization (DESI) - modified ESI for samples under ambient conditions [20];
    4. Matrix-assisted laser desorption/ionization (MALDI) - allows selective ionization of solid samples based on laser light absorption; [21]
    5. Atmospheric-pressure chemical ionization (APCI) - produces ions by adduct formation or proton abstraction. [21]

4.1.2. Ion separation - Acceleration

Ions are further separated in **a mass analyzer** by their mass-to-charge ratio utilizing acceleration through a magnetic or an electric field in a vacuum. Separation is based on the

deflection of the ion as a subject to interaction with the magnetic field or sped up as a result of the electrical field's influence. The selection of a mass analyzer significantly impacts the quality of mass spectra information. The second most important is a selection of **fragmentation techniques** - inducing the formation of product ions [23].

4.1.3. Mass Selection - Deflection

The ion deflection is the process resulting from the kinetic energy and the momentum of the ions. The electrical field governs the deflection on the y-axis. Cations move in the ascending angle, and anions in the opposite. The deviation from the trajectory is higher for slower ions and smaller for the faster ions. Therefore, the turning on the y-axis allows us to infer the ion's kinetic energy.

The magnetic field distorts the beam of the ions horizontally. The angle corresponds to the m/z and the electric charge. The lighter the ion is, the more it is deflected [21].
Two physical laws govern the dynamics of the charged particles:

- Lorentz force law:

$$F = Q * (E + v \times B) \tag{1}$$

- Newton's second law:

$$F = m * a \tag{2}$$

where

F - force applied to the ion

m - mass of ion

a - acceleration

Q - ion charge

E - electric field

$v \times B$ - vector cross product of ion velocity and magnetic field

Resulting in a differential equation of motion for charged particles

$$(m/Q) * a = E + v \times B \tag{3}$$

4.1.4. Ion Detection

The resulting ions pass through an ion detection system and are analyzed using various methods, e.g., electron multiplier, Faraday cups, and image current detection, and transformed into a signal. The system generates electrical current. **Signal amplification** and **digitalization** of the ion packages are usually needed [21].

4.1.5. Tandem Mass Spectrometry

Tandem mass spectrometry, or MS/MS, denotes any technique where the precursor sample is subject to more than one stage of mass spectrometry [20]. Molecule fragmentation, chemical reaction, or dissociation usually separate the consequent steps. The distribution of the steps can be relative to the time or space, thus called tandem-in-time and tandem-in-space, respectively [21]. MS/MS enables the engineering of a variety of experimental sequences. It is common to employ separation techniques such as chromatography. Gas chromatography/ mass spectrometry (GC/MS) is the most widespread. The liquid chromatography/mass spectrometry (LC/MS) is used based on the properties of the analytes.

4.2. Analysis in Mass Spectrometry

Among possible visual representations of mass spectrometry analysis, the most common is the **mass spectrum**.
Other data representation techniques include **mass chromatogram** and **three-dimensional contour map**.

4.2.1. Mass Spectrum

Mass spectrum represents the two-dimensional distribution of ions by their mass-to-charge ratio in a sample. It can illustrate both **fragments** and **intact molecular masses**. The position on the x-axis depicts the mass-to-charge ratio of an ion resulting from the sample fragmentation. The y-axis represents the signal intensity of the ions or the measure of the abundance of the ions. The intensity representation is possible in several ways:

- Number of Ions
- Counts per second (cps) for particle counting detector
- Volts for analog detector
- Squared amplitude for analysis in the frequency domain

The peak of the ion with the highest abundance is usually the peak of an intact ionized molecule, the base peak. The intensity on the y-axis is usually normalized concerning the highest peak value of ions further from the y-axis, thus with lower m/z representing the fragment ions [21].

4.2.2. Ion chromatogram

The chromatography techniques, coupled with mass chromatography, are sometimes used to separate components of the mixture. The individual segments leave the chromatography column at the specific retention times. The analysis of the separate compounds happens one by one in the tandem-in-time MS/MS. The ion chromatogram represents the abundance of the ion as a function of time [21]. The x-axis, in this case, represents retention time. The y-axis can represent the total ion current (TIC), the most intense peak in each spectrum (base peak chromatogram). Usually, the graph contains information about mass tolerance (dependent on mass accuracy and resolution).

4.2.3. Three Dimensional Contour

Represents a modified form of either one of the previous techniques, where the z-axis displays a piece of additional information about experimental parameters.

4.2.4. Data analysis

A computer is a compulsory component of mass spectrometry analysis. It performs three principal tasks:

- mass spectrometer control and setup
- data acquisition and preprocessing
- data visualization and interpretation

The computer works with the information in the binary format while the mass spectrometer produces continuous electrical impulses. Therefore, the need to convert the data arises. For this reason, the analog-to-digital converter (ADC) lies in the interface of information flow into the computer, and the digital-to-analog converter (DAC) on the opposite side [20].

The computer can analyze the sample and provide insight into its origin and nature or identify it using specialized algorithms. More importantly, it can move the data in the database for later service. The commonly employed database solutions offer swift writing properties and allow cloud backup.

The first strategy for identifying an unknown compound is to compare its experimental mass spectrum with a mass spectra stored in the reference library. The analyst has two possible search strategies at his disposal. Firstly, he can compare a new analyte with the library and explore the best matches. Alternatively, he can query the database and check for the possibility of the presence of the compound in the database [20].

The process of a compound's chemical structure identification is called structural elucidation. The workstation, in this case, tries to assess all possible structures that are compatible with the queried analyte using the candidate's generation or search.

4.3. Small molecule identification

Strategy selection and success of sample identification depend on our previous knowledge of its molecules - analytes. Analytes can be grouped based on two binary classifications:

1. Identified, therefore present in a library database/knowledge-base;
2. Expected based on a priori knowledge of the sample and the conditions.

Based on the two, we can stratify four groups of analytes:

1. “Known” (both (1) and (2)) - analyte is confirmed or quantified in the sample;
2. “Known not expected” ((1) and not (2)) - common contaminants, matched with library;
3. “Unknown but expected” ((2) and not (1)) - analytes resulting from metabolism or chemical side reactions;
4. “Unknown and unexpected” (neither (1) nor (2)) - no *a priori* knowledge of the analyte (“*de novo*” identification)

4.3.1. Strategies of identification

Two main groups of identification strategies exist:

- Targeted - looks for certain chemicals only (based on knowledge of their mass spectra and retention times). For instance, selected reaction monitoring (SRM) increases both selectivity and sensitivity by limiting the amount of data.
- Non-targeted - looks for any chemicals that are detectable in the sample. A typical example is a time-of-flight. The problem of data processing rises.

Another categorization concerns the source of candidates, whether being a product of *in vitro* or *in silico* creation:

- Library Searching
- De Novo Sequencing

4.3.2. Impact of Disturbance

The signal of an ion can be related to the **compound of an interest** or **background disturbance**. The disturbance's treatment poses a task of significant importance. The possibility and extent of **background subtraction** usage depend on the previously selected

identification strategy. The following lines give an overview of various classes of possible contamination sources.

Disturbance due to background ions:

- impurities,
- contaminants,
- degradation of the LC column,
- cross-contamination,
- carryover of previous samples [25]

Disturbance due to background noise:

- electrical noise,
- artifacts in the transformation of the data [26]

The interesting molecule's signal distribution spans multiple entities in the mass spectrum. Besides the distribution over the different isotopes, other factors influence the mass spectrum:

- the concentration of the precursor,
- no ionization of compound given the conditions
- multiply charged ions,
- absence of the molecule of the interest
- formation of adducts,
- in-source fragmentation,
- creation of dimers.

4.3.3. Reporting standards for metabolomics analysis

The previous sections briefly discussed several variables that influence the qualitative characteristics of the retrieved mass spectrometry data. Whether it is instrumentation or possible disturbances due to various conditions, collective behavioral patterns categorization is possible and crucial. The success of this task lies inconsistent practices in meta-data reporting the chemical analysis.

The mass spectrometry analysts community asks for the application of minimum guidelines for metadata reporting. The common reporting standards could allow the analysts to explore the context of the experimentation conducted in the past and stored in the library, look for common patterns, and replicate the experiments when needed. The benefit of metadata's

availability rises with the consistent effort concerning its collection. The reporting standards postulate the guidelines of the experimental conditions storage and communication, therefore, maximizing the utility of the data for further analysis and comparison. However, they do not pose restrictions on obtaining the spectra [27].

4.4. Similarity Scoring Algorithm

4.4.1. Methodology

Library searching algorithms design represents a classification problem. The classification problem aims to find a possible query's representation in the library of possible classes based on the input data. The inputs are regularly an evaluation of similarity between spectra on the query side and the library side. The extent of the correspondence, which is subject to the selected scoring method, gives rise to a confidential assessment. The resulting list of the candidates represents a set of pairs with ordering defined by the score's value. A broadly used method for similarity assessment is cosine similarity scoring. Many modifications exist that employ various filtration strategies in series with the scoring algorithm, with the aim of false-positive candidate reduction. This false-positive candidate represents any candidate in the list that scored higher than any predefined threshold.

Cosine similarity scoring algorithm:

$$S_C = cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

4.4.2. Selection - Filtration of candidate library compounds

The most commonly used filtration technique is the precursor mass comparison and selection of the candidates with the same precursor mass or the mass in the range of the designed resolution of the device.

The first condition of the precursor matching says that, at last, one precursor peak must overlap in each spectral pair. This approach poses a very effective and narrow filtering method, which implicitly raises the confidence in the library matching because it directly relates to the compound. Yet, there are events at which precursor information may not be

present. The filter removes many correct matches where the precursor peak is different due to the adduct formation and in-source fragmentation.

Second, we may filter out candidates based on the highest peaks number that overlap. This approach retrieves candidates for which the n-highest peaks rule applies. The n-highest peaks algorithm represents a generic filter that removes false-positive candidate spectra with higher similarity overall. The drawbacks of this approach are that the n-highest peaks do not relate to the actual compound identity, only to spectral similarity. Moreover, the false-negative may not indicate low identification probability, only the inability to distinguish based on spectra alone.

Thirdly, the downstream filtering options remove the candidates based on a specified threshold on the dot product values over the spectral pairs. This approach is an easy to calculate, optimistic approach.

4.5. Background to Data Science

In the 21st century, we can confidently say that nearly anyone interacted with data science products or performed data science himself [30]. It represents a systematic discovery procedure illuminating valuable patterns and relationships in statistical or factual quantities [31]. Various approaches toward data science are used daily by myriad scientists around the World. Collectively, typical stages of the data science procedure identify many of them. However, the naming convention and categorization of the phases can differ considerably. One of the most widely used structures is the Cross-Industry Standard Process for Data Mining (CRISP-DM) [32]. CRISP-DM proposes six components of the cyclic framework - business understanding, data understanding, data preparation, modeling, model evaluation, and deployment.

Introducing such a standardized pipeline creates a common ground for communication and understanding between scientists and stakeholders. It models the whole data science process. However, the model is simplified and may result in a false assumption that the individual stages are independent. In truth, they are strongly related. The unresolved challenges of one step may show its implications in the following [32]. This sheds some light on the intertwined nature of the whole procedure. Moreover, each phase is sensitive to the internal and external attributes of the others. In the lines to follow, we address the steps and explain their role in the CRISP-DM framework.

4.5.1. Business Understanding

Understanding the market opportunity for the developed data analysis/machine learning algorithm is a vital action. It connects the research with the business and creates a possibility to move the idea from scratch to a monetizable product. When designing any Software as a Service (SaaS) model, a comprehensive understanding of a business perspective is crucial. The knowledge of a business allows a clear goals setup with tangible implications and gathering all decisive information about the business background and current opportunities and threats. The detailed situation assessment is a determining step toward success with a central purpose of resource identification. The resources could be any relevant requirements, including personnel, data, hardware, and software resources. Evaluation of competition helps to mitigate any conflicting patents [33].

4.5.2. Data Understanding

Thorough exploration and preparation allow a deep insight into the provided data. The ability to generate the data, describe them, and act on the obtained discernment has vast importance. Data generation requires an understanding of data streams and the storage models employed in our use case.

Database Management Systems (DBMSs) represent a traditional storage system with optimized writing properties. This property makes it a robust tool for recording logs at high speed. The record represents a row in relational tables which build the DBMSs [34]. Querying those systems, in contrast, occurs less often. By design, DBMSs are not read optimized and not suitable for rapid inquiries fulfillment (continuous queries).

Join queries are the prime class of data extraction mechanisms. They allow for collective extraction of rows from two and more tables and their combination based on their relation. Aggregate queries serve as a bridge between extraction and description of the data, performing a mathematical operation on the dataset in the queried database records. It returns a final result of an action, which may decrease the quantity of communicated data out of the database.

4.5.3. Data Preparation

In-depth exploration of the data provides the building stone for our success and possible assessment of the potential. Exploration and descriptive analysis of the data is the first stage of preprocessing and feature engineering. Management of the quantities molds them and adjusts them to fit our data science task. Moreover, it helps to expose useful features and apply knowledge-based insight to them.

Preprocessing the data consists of:

- Data Parsing
- Data Cleansing
- Data Types conversion
- Data Transformation
- Outlier Handling
- Feature Selection
- Data Sampling

Data parsing represents a procedure of data format conversion into another, more suitable format. The input data source may come in different kinds and flavors, e.g., plain text format, column separated values (CSV), database, or JSON [30]. Once the data parsing is complete, transformation into a more favorable structure, given a technical solution, occurs.

Data cleansing consists of two overlapping parts: missing values and duplicate removal. Both elements should be respectively comprehended when occurring in the data set. One of the common frameworks is to track down the provenance of the data source [31]. Upstream tracking can reveal systemic issues of data retrieval method, storage system, and miscellaneous parsing. After careful consideration, the treatment of the data occurs. Duplicate samples can be, in many cases, safely removed. Missing attribute values may be discarded or replaced by systematically and artificially manufactured values.

Individual features of the data set may take the form of four types: nominal, ordinal, discrete, or continuous. Nominal attributes possess no mutual relationship with the other and, therefore, no order. They provide no quantitative value and troublesome automatic grouping. The interrelation of ordinal values lies in the possibility of being ordered. Discrete features represent a finite set of possible values. Continuous features denote an infinite continuum where an unbounded number of values separates any two values.

4.5.4. Feature Engineering

The representation of the observables may pose a significant factor for the statistical learning performance and interpretation. This part of the procedure is called feature engineering. It is usually the most time-consuming and consequential task of the data scientist. Whether conversion or modification of existing features, their combination, or refinement based on domain knowledge, this step requires a highly experienced data scientist's intervention [35].

Feature engineering is usually performed based on the best practices given the nature of the data and domain expertise. The central idea is to spread the training space by introducing new variables. The engineered features shall be independent of other observables, which is usually impossible. Therefore, a correlation is non-zero.

A common problem of feature engineering is the situation when collinearity occurs. This occurrence refers to a state when two variables are closely related together. Related features in the means of correlation can pose a problem regarding the model's predictive power. During feature importance assessment, it is hard to separate distinctive effects of the correlated features. To illustrate the problem, imagine a case where more correlated features with a little predictive power over the response occur. We can consider them as one predictor with the weighting respectful to their absolute number. The statistical learning methods requiring normalized data could interpret those as a more important feature. A simple, yet not complete, assessment of the collinearity between two variables is performed by computation of the correlation matrix and [36].

4.6. Background to Probability

Probability theory gives a mathematical basis to analyze experiments that produce different outcomes. From the perspective of probability, a well-defined trial has a defined set of results. The number of results can be infinite. Each experiment results in one sample. The group of all experiments is called the sample space. Mathematically, a sample space is a set, and the outcomes are the "elements of the set."

4.6.1. Notions of the Probabilistic Beliefs

Let's introduce a set of variables X_i where $i \in \mathbb{N} = \{1,2,3,\ldots,\infty\}$. Each variable X_i represents a finite set of possible outcomes x_i generally denoted by ω. All possible effects collectively will be called **sample space** and described as Ω. **An event**, marked by letters of a Greek alphabet, is a set of outcomes of an experiment to which a probability is assigned. We denote that the sample space represents a finite space in this case.
Event is, for example, all outcomes of dice throw equal to three or all throws less than 4. This example suggests that outcome might be an element of many different events. Different events usually hold different likelihoods. [37]

4.6.2. Probability

Represents a numerical description of how likely an event is to take place or how likely it is that a proposition is true. Number in the range <0,1> where zero indicates an impossible event and one certainty. A proportion of desired outcomes to all outcomes.
The book Learning Bayesian Networks by Richard Neapolitan defines conditions that a probability function must satisfy as follows [38]:

1. $0 \leq P(\{e_i\}) \leq 1$ for $1 \leq i \leq n$
2. $P(\{e_1\}) + \ldots + P(\{e_n\}) = 1$
3. For each event $E = \{e_{i1}, e_{i2}, \ldots, e_{ik}\}$ that is not an elementary event,
   $P(E) = P(\{e_{i1}\}) + P(\{e_{i2}\}) + \ldots + P(\{e_{ik}\})$

The pair (Ω, P) is called a probability space.

Two probability interpretations exist out there:

- **Objectivists** (**frequentists**): the probability of a random event denotes the relative frequency of occurrence of an experiment's outcome when repeated indefinitely. Therefore, the relative frequency does relate to the whole sequence of trials and not to a specific trial.
- **Subjectivists** (**Bayesian**): degree of belief. Includes expert knowledge as well as experimental data to produce probabilities - prior probability distribution

$$P(p) = \frac{p}{p+q};\ P(A) = \frac{A}{A+\neg A} \tag{5}$$

Where

p - set of desired outcomes (event)

q - set of all other outcomes (a complementary event also denoted as A', A^C, $\neg A$)

Concerning normal distribution, it represents the area under the curve for a specific event given the mean and standard deviation of the fixed distribution.

4.6.3. Likelihood

Concerning normal distribution, it represents the ratio of occurrences of a given known event in distribution with varying mean and standard deviation. Likelihood deals with fitting models given some known data.

4.6.4. Conditional probability

It is a measure of the probability of an event occurring, given that another event (by assumption, presumption, assertion, or evidence) has already happened. In statistical inference, it can be seen as an update of the probability of an event based on new information (suppose that α is an event of interest and β a new event). We can say "α given β" or "probability of α under condition β".

The book Learning Bayesian Networks by Richard Neapolitan defines conditional probability as follows [2]: Let E and F be events such that $P(F) \neq 0$. Then the conditional probability of E given F, denoted $P(E \mid F)$, is given by

$$P(\alpha \mid \beta) = \frac{P(\alpha \cap \beta)}{P(\beta)} \tag{6}$$

4.6.5. Statistical independence

Two events are independent, statistically independent, or stochastically independent if the occurrence does not affect the probability of occurrence of the other. In the collection of events, we distinguish pairwise independence (between two events) and mutual (collective, strong) independence when each event is independent of any group of other events.

The book Learning Bayesian Networks by Richard Neapolitan defines two conditions out of which one must hold if two events E and F are independent [38]:

1. $P(E \mid F) = P(E)$ and $P(E) \neq 0, P(F) \neq 0$
2. $P(E) = 0$ or $P(F) = 0$

4.6.6. Conditional Independence

Describes the relevance of observation toward the evaluation of the hypothesis. Independent observation is redundant and uninformative in the sense of no influence over prior probability. It can be denoted as follows:

$$P(A \mid B, C) = P(A \mid C) \tag{7}$$

"Where $P(\alpha \mid B, C)$ denotes probability of an event A given both event B and event C and $P(C) > 0$.

A and B are said to be conditionally independent given C, written symbolically as:

$$(A \perp\!\!\!\perp B \mid C) \tag{8}$$

The book Learning Bayesian Networks by Richard Neapolitan defines two conditions out of which one must hold if two events E and F are independent [38]:

1. $P(E \mid F \cap G) = P(E \mid F)$ and $P(E) \neq 0, P(F) \neq 0$
2. $P(E) = 0$ or $P(F) = 0$

4.6.7. Bayes Theorem

Describes the probability of an event, based on *a priori* knowledge of conditions that might be related to the event can be derived from conditional probability. Bayes Theorem serves as a unified framework for updating our beliefs.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{9}$$

Where $P(A)$ is a prior probability, $P(B)$ is a marginalization, $P(B \mid A)$ is a likelihood, and $P(A \mid B)$ is a posterior probability.

4.6.8. Law of total probability

The law of total probability is a theorem that, in its discrete case, states if $\{B_n : n = 1,2,3,...\}$ is a finite or countably infinite partition of a sample space (in other words, a set of pairwise disjoint events whose union is the entire sample space) and each event B_n is measurable, then for any event A of the same probability space:

$$P(A) = \sum_n P(A \cap B_n) \tag{10}$$

Or:

$$P(A) = \sum_n P(A \mid B_n)P(B_n) \tag{11}$$

For conditional probabilities

$$P(A \mid C) = \sum_n P(A \mid C \cap B_n)P(B_n \mid C) \tag{12}$$

4.7. Machine Learning

Machine Learning, abbreviated as ML, represents an application of statistical learning in silico. Any ML has a set of quantitative or qualitative observations used to predict an outcome that may be quantitative (continuous, ordinal) or qualitative (categorical, discrete, nominal). In this process, we use training data, which represents the set of available observations as an input (predictors, independent variables). This data may contain information about the outcome, or the outcome may be something that sheds insight into the relationship between observations utilizing clustering or organization, the output (response, dependent variables). The former is called supervised learning or the process of predictive model creation. The latter is denoted as unsupervised learning or the descriptive or generative model creation. The model is also called a learner. The process of learner training makes it able to perform prediction of the desired outcome on previously unseen data for which the outcome is unknown. Based on the desired effect, quantitative or categorical, we will discuss regression for the former and categorization for the latter. Both tasks can be viewed as function approximation problems. For a deeper insight into the machine learning theoretical background, check the book by Hastie et al. [39]. To better understand a practical implementation, check the text from Mülle and Guido [40].

4.7.1. Algorithm

Many machine learning techniques that tackle prediction and description problems exist up to date. To discuss each, we would have to dedicate tremendous space and time for which the extent of this thesis is not sufficient. Therefore we will leave only the reference to papers and books.

We will discuss the typical pattern found in the wast majority of the models. Machine learning models can differentiate as parametric and non-parametric [36].

The parametric model's creation consists of two phases. The first is the model's functional form selection. This approach, therefore, tries to assume and approximate the function resulting in training data. Second, we use the training data to find parameter values. Most commonly, this process is called training or fitting.

Non-parametric modeling estimates the real functional form. Except for looking for a possibly smaller number of parameters, the non-parametric model has to store all the observations used for its training to make any estimation for previously unseen data. This process is memory-consuming, yet the model creation usually requires lesser processing power.

4.7.2. Evaluation

Correct performance assessment is essential for model selection and quality evaluation. There is no single fit-for-all algorithm in the machine learning pipeline, and model selection is usually subject to the collection of presumptions. Performance comparison between various models is therefore needed. The stereotypical way of model's performance evaluation for the quantitative output is the mean squared error evaluation between the ground truth and the model output. The error on a test set can be called generalization error, or the measure of the inability to make the correct prediction on independent data set. In other words, the capacity of generalization relates to a performance on a completely separate data set. If any model's correction, like the loss function minimization on test data during hyperparameter tuning (check 4.8.3 Tuning), was made based on the data, they are no longer independent.

The most widely used method for prediction error estimation is cross-validation. K-fold cross-validation mitigates the problem of prediction error assessment on scarce data set by splitting the training set into k fold and then evaluating the performance on a given fold using the model trained on the other ones.

4.7.3. Tuning

How to get the best out of your model?
Once the analyst decides the best machine learning approach, given the generalization of the task, we can tweak its parameters to improve its qualitative measures further. We denote that those parameters are not the same as those introduced for parametric models in subsection 4.8.1 (check 4.8.1 Algorithm). Those parameters are external parameters of the model or are more often called hyperparameters. Hyperparameters are model specific and relate to the model's complexity and the generalization ability. The standard method for hyper-parameter tuning is the grid search. The optimization method tries to find a combination of hyperparameter values that minimizes the loss function by exhaustively training and validating models for each [3]. The development of methods without optimality guarantees addresses the time complexity problem by introducing successive halving. This approach represents an optimization strategy that evaluates all the combinations of parameters on a small portion of resources and promotes the best ones for training on a more extensive segment of the training data. The process repeats until the last candidate

[41]. Hyperparameter optimization is usually combined with cross-validation introduced in 4.8.2 (check 4.8.2 Evaluation).

Hyperparameter tuning adapts the complexity of the model to a training data set. To a certain extent, it helps the model with generalization. However, at a specific point, the further complexity increases, raises the expected test error too. The model learned underlying structures particular to the training set but not the whole dataset from the domain. Two characteristics of the model's quality are contained yet not displayed in the MSE without comparative analysis.
The variance deals with the changes of functional form estimate as subject to varying proportions and batch of training data set. Alternations in model performance are typical. However, the variance shall not be too high. The bias represents an error of the functional form approximation; along with the increasing complexity of the model, the variance increases, and the bias decreases. Therefore the typical U-curve of the MSE can be observed when running the hyperparameter tuning.
We shall pay attention to one aspect of hyperparameter optimization, which is the actual generalization ability, which cannot be assessed using a test set. Since the prediction error rate on the test set served as a stopping criterion, the value no longer represents independent data [40].

4.7.4. Interpretability

Except for the prediction discussed more extensively, inference is a vastly important area of machine learning. Sometimes great importance is given to understanding the influence of the features on output. It sheds light on how the model makes its decisions and tells us which predictors are associated with the response and consequently which measurements to track, what are the relationships between the response and predictor, and how can we effectively approximate the model with a simpler version [36]. However separated the prediction and inference might seem, we find them interconnected in various settings. Such an example is a model deployed in practical settings where the need to be understood by the stakeholders is high or critical to identify blind spots in the medical setup. Therefore the tradeoff between interpretability and performance is needed. Simple, more restrictive models provide simple interpretation but usually yield poorer performance. The more complex models offer state-of-the-art performance but are interpretable only partially or not at all.

4.8. Bayesian networks

This subsection will further develop the concepts drawn in 4.7 (check 4.7 Background to Probability). Bayes Theorem gives a robust framework for updating our prior beliefs based on new probabilistic evidence. Its basis lies in the computation of a conditional probability. The theory allows performing probabilistic inference based on an exact algorithm. The drawback of this algorithm is the exponential complexity of conditional probability computation concerning the number of prior events. Bayesian Networks offer a partial solution for this problem.

Bayesian networks belong to a class of graphical models [42]. In general, graphical models involve two components: qualitative and quantitative.
The qualitative component of the Bayesian Network is a directed acyclic graph (DAG). The mathematical representation of DAG is G = (V, E), where V represents a set of random variables, unknown parameters, observables, or hypotheses (nodes), and E symbolizes direct influence between them (edges or paths). The DAGs edges are the representation of the probabilistic relationship among variables. The qualitative element of a Bayesian Network takes the form of conditional probability tables [37].

The practical element of Bayesian Networks lies in conditional independence usage, which allows performing a probabilistic inference amidst the features in a lesser amount of time. Independence represents the disconnection of nodes of the DAG. Moreover, it makes up for a compact representation of the probability distribution exponential in size [37].

The second aspect of Bayesian Network favorability is its usage as an inference model. Based on given evidence, or known features, answers questions about the cause. The probability distribution of an unobserved variable is called inverse probability. This aspect builds on a specific creation procedure. By manipulating random variables in systematic order and observing changes in the others, we can draw edges between random variables if one is the direct cause of the other. The stronger interpretation of the observation is that the influence is causal [37].

4.8.1. Learning Bayesian Network

Domain knowledge and data usage are viable approaches toward BN model fitting [45]. Learning of Bayesian Networks is composed of two stages.

In the first, we identify the topology of the DAG in structure learning. Finding the best structure among all possible arrangements is an NP-hard problem [44]. Therefore, great importance lies in the development of heuristic search strategies. They formulate optimization problems based on model selection criteria as the goodness of fit or network's complexity [43].

Second, we compute the conditional probability tables in parametric learning. This step represents a computation of the joint probability table.

4.8.2. Causal models

Machine learning models became a common technique to obtain valuable insights and make predictions. Using numerous methods, good performances are achieved in various areas of classification, clustering, regression, or dimensionality reduction. On the other hand, ML usually does not give straightforward answers to causal relationships between variables.

4.8.3. Correlation and association

Although being used interchangeably, correlation and association denote two sets of relationships. Correlation is related to linear relationships, whereas association is any sort of relationship. Neither is sufficient to deduce causation because statistical relation does not uniquely constrain causal ties [46].

4.8.4. Causation

Causation is the influence of the independent variable so-called predictor, on the dependent variable (outcome variable). Formulation by Reichenbach:
If two random variables X and Y are statistically dependent (X / Y), then either X causes Y, Y causes X, or there exists a third variable Z that causes both X and Y. Further, X and Y become independent given Z, i.e., $X \perp Y / Z$. This definition is incorporated in Bayesian graphical models.
The causality is difficult to prove. It requires a high level of statistical rigor and carefully collected data. There are other ways to decide based on strong correlations that maximize our chances of making the “best” decision.

5. Methodology and Research Methods

This section provides information about the research objective and procedure applied to achieve the thesis goals. We will provide a framework for our data handling, introduce data sources and discuss methods to evaluate and interpret the results of our research.

Throughout the section, we follow a set of guidelines widely used in the data science community called The Cross-Industry Standard Process for Data Mining (*CRISP-DM*). *CRISP-DM* describes the process model of the data sciences life cycle.

Six stages depict the *CRISP-DM:*

1. Business understanding – What are the goals and objectives?
2. Data understanding – What is the nature of the data?
3. Data preparation – How do we manage the data?
4. Modeling – How do we build and assess the ML model?
5. Evaluation – Which model generalizes the best?
6. Deployment – How does the business use the results?

5.1. Object of Research

In the words of the data science life cycle, this part represents a business understanding. We will discuss objectives, assess the risks, specify the technical requirements, and plan the project.

The description of our research objective is in section 2 (check 2. Thesis Goal). In section 1, we provided an overview of existing methods, which address the same goal. The correct small-molecule identification, in general, also implies the reduction of the false-positive rate (check 1. Current state of research). The success criterion is to select the correct matches from a list of candidates based on experimental metadata.

The tools proposed in 5.2 make up crucial components of our novel algorithm for small-molecule identification (check 5.2 Procedures (pipeline)). The algorithm based on a data-driven approach broadens the family of computational methods in mass spectrometry. By data-driven, we mean any technique for making deliberate decisions based on data analysis and interpretation. We will present a data curation and feature engineering pipeline developed on knowledge about data retrieved from exploratory data analysis and visualization.

By data, we will further denote an annotated list of query-library spectrum matches retrieved from the mzCloud database and used to demonstrate the accuracy of our algorithm. The retrieval of the data and its sources will be discussed in 5.3 (check 5.3 Usage of data and data sources).

We acknowledge that our data-based model has limitations. Though extensively trained on consistently and transparently curated metabolomics data, it may produce slightly different results when trained and evaluated on the other databases while following the pipeline proposed in 5.2 (check 5.2 Procedures (pipeline)).

Python, an open-source high-level programming language with a broad community of commercial and private users, served as a development platform throughout the pipeline [47]. Prototyping of the machine learning pipeline was performed in the Google Colaboratory environment based on the interactive web computing platform Jupiter Notebook. Google Colaboratory requires no configuration of the environment and allows

collaboration on python-based projects, as the name suggests. Moreover, it provides cloud resources rich in rapid access memory. JetBrains' DataSpell served for local interaction and execution of tasks, with high processing power consumption.

Library NumPy aided the mathematical computation throughout the code [48]. Pandas, an open-source tool, administered data manipulation and description [49]. Scikit-learn and Pomegranate served for modeling and evaluation [50, 51]. Matplotlib assisted with an easy visualization [52].

5.2. Usage of data and data sources

Our objective drives the need to identify, collect and analyze the metabolomics data set. The data used during our research derive from a subset of the mzCloud reference database. SQL query combined with pandas toolbox secured the data loading into our analytical tools and creation of further used .csv file.

Data understanding is a crucial step toward the pipeline's design. Figure 1 shows the essential properties of the structure and relationships in the input data. Our dataset represents a collection of MS Experiments. Each experimental sample contains numerous compounds identified in the mzCloud reference library of compounds. The quality of spectra that describe every compound is of critical importance in the identification success. The similarity scoring algorithm provides a single measure of the match between unknown and library spectra, from which the correct compound can be elucidated.

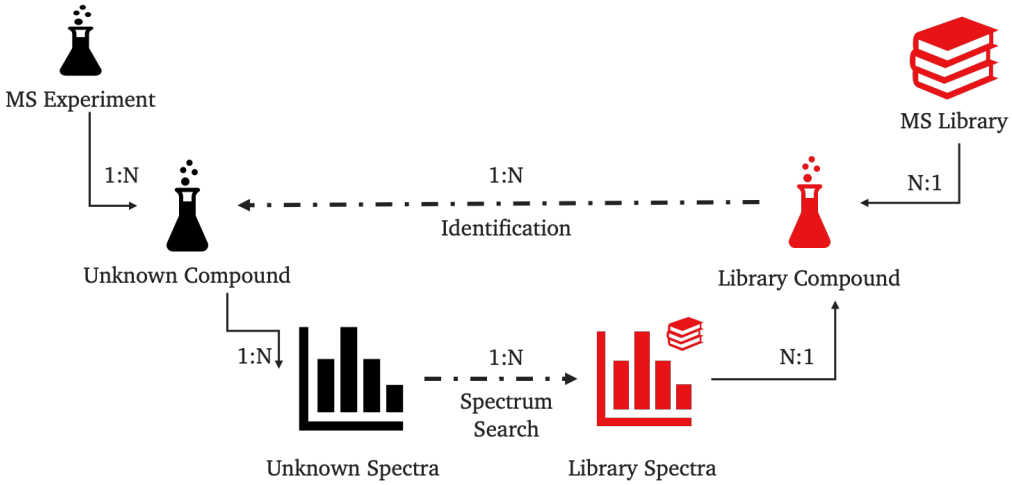


Fig. 1 Structure of the data and its aggregation. Unknown represented black, and the reference library depicted red from the left side. MS experiment performed on any sample yields N compounds. Analysis of the unknown compounds in the tandem MS/MS results in N spectra. Each spectrum is subject to similarity scoring against the library spectra. Likewise, more library spectra belong to one of many compounds stored in the library.

Two datasets combined, one for authentic compounds and one for blood group identification using HILIC/Reversed Phase, represented our data. For those information about the queried compound identities Table 1 gives an overview of quantitative aspects of the input data following the former description. The retrieved values have the missing values removed.

Tab. 1 Overview of quantitative aspects of the input data. Count results from dataset observation without knowing the identity of individual ions. The number of unique unknown values derives from the identity-based grouping of individual ions. The deviation between count and number of unique on the library side (red) results from the fact that one library spectrum/compound can be retrieved more than once for various unknowns.

	Count	No. Unique
MS Experiments	249	249
Unknown Compounds	1701	731
Unknown Spectra	34682	34682
Library Spectra	1189678	23353
Library Compounds	11980	1839
MS Library	2	2

The matches are ranked based on similarity scores from the best to the worst. Table 1 shows that for 34682 unknown (queried) spectra, the Cosine algorithm paired 1189678 candidates from the reference library. Ground truth information about queried compounds allowed for elucidation of whether the hit was true-positive or false-positive. It is vital to restate that the InChIKey is a unique compound identifier describing chemical formula with a fixed length of 27 characters. Based on common knowledge of mass spectrometry's limitation to differentiate between isomers, we will further denote InChIKey as the first 14 characters of the identifier and call it compound indistinguishably. The database did offer references for all experiments InChIKeys. However, the cosine similarity scoring did not retrieve the correct candidate for every spectrum. Table 2 gives an insight into the identification accuracy of algorithms without any precursor matching.

Tab. 2 Identification accuracy by similarity scoring candidates ranking first and all candidates. 1st rank represents the highest scoring candidate.

	Spectra	Compounds
Cosine Scoring, 1st rank	33.91%	15.93%
Cosine Scoring, any rank	51.38%	27.69%

5.2.1. Description of queried metadata

The database offered a range of metadata for library and query spectra. Since our objective was to assess the confidence of correct matching between query and library based on experimental conditions, we used features that could be found in the database for both. This way, we created the annotation and described the deviation between experimental conditions, as described in section 5.2, under which the spectra were obtained (check 5.2 Procedures (pipeline)). Here we present metadata that we queried from the database and referred to throughout the thesis.

- **RawFileName**

  Represents the name of Thermo Fisher raw files storing unknown compounds with spectra and metadata (e.g., HILIC_NEG_LIQ_1_MS3_CID.raw).

- **QueryScanNumber**

  The number represents the order of spectra in the raw file defined by the order in which corresponding substances are eluted from chromatography (e.g., 256).

- **QueryInChIKey**

  InChIKey represents the compact version of the IUPAC International Chemical Identifier. It has a fixed length of 27 characters, making it better suited for searching and indexing (e.g., NKBWMBRPILTCRD-UHFFFAOYSA-N).

- **QueryScanFilter**

  Metadata of queried spectra in form of, e.g., FTMS + p ESI d Full ms2 198.19@cid35.00 [50.00-210.00], where:
  FTMS = mass analyzer (Fourier Transform MS)
  '+' '-' = ionization mode (positive or negative)
  ESI = ionization technique (electrospray ionization)
  Full = scan mode
  ms2 = first MS stage
  198.19 = m/z of precursor ion
  cid35 = fragmentation technique (collision induced decay)
  35 = NCE (intensity of 35%)

[50.00-210.00] = mass range (fragments are shown in a mass range from m/z 50 to 210)

- **LibrarySpectrumNCE**

  Normalized collision energy used to produce fragmented ions.

- **LibrarySpectrumIonActivation**

  Represents an abbreviation for a mass spectrometry technique used to incite ion fragmentation (e.g., HCD, CID).

- **LibraryMSStage**

  The first stage at which fragmentation was performed (e.g., 2).

- **LibraryType**

  Specifies the source of the library spectra. Two options are possible. "Autoprocessed" refers to automatically preprocessed spectra, and "Reference" presents spectra manually curated with high precision.

- **Score**

  Represents the cosine score in percentile value which evaluates the similarity of two spectra.

5.2.2. Exploratory data analysis

Visual exploration provides important clues about the distributions of values given the feature. Figure 2 depicts the prevalence of true and false positive hits on the left side, which indicates the imbalanced nature of the dataset. The other two graphs represent the cosine similarity score distribution for true-positive candidates (middle) and false-positive hits (right). The similarity scoring retrieved approximately five false-positive candidates for each true-positive candidate. We observed expected negative skewness for the true-positive candidates. Nevertheless, false-positive candidates are negatively skewed too. Denver similarity scoring used to retrieve the input data provides less confident decisions.

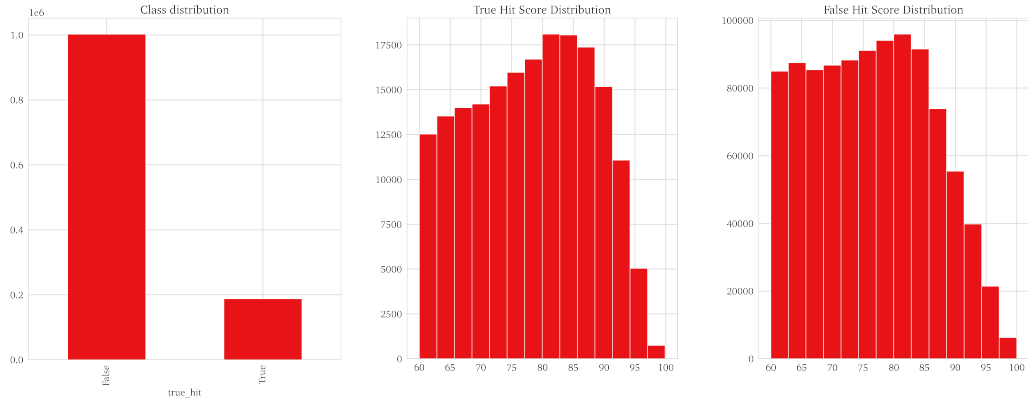


Fig. 2 Histograms depicting the distribution of; the true-positive and false-positive candidates (left), the similarity scores for true-positive candidates (middle), and the similarity scores for false-positive candidates (right).

Figure 3 shows the distribution of features available for both unknown and library spectra. The left side of the graph depicts NCE. The library collects spectra in a broader range of NCE. The trend could mean the presence of more fragments and smaller peaks in some candidates, and a greater chance of false positives. However, we should be careful about the generalization of this trend since new queries may contain spectra obtained at higher NCEs as well. MS stage 2 is the only observed stage in unknown spectra. Similarity scoring retrieved candidates of various MSStages, which resulted in false-positive identification in the vast majority of the cases. In general, hits that do not match the MS stage are false positives or randomly assessed true-positive candidates. The right side of the graph illustrates the presence of two ion activation techniques and their relative abundance in recorded experiments. We can observe the domination of Higher-energy C-trap dissociation (HCD) usage over Collision-induced dissociation (CID). Both are fragmentation techniques; however, the former is specific to the orbitrap analyzer.

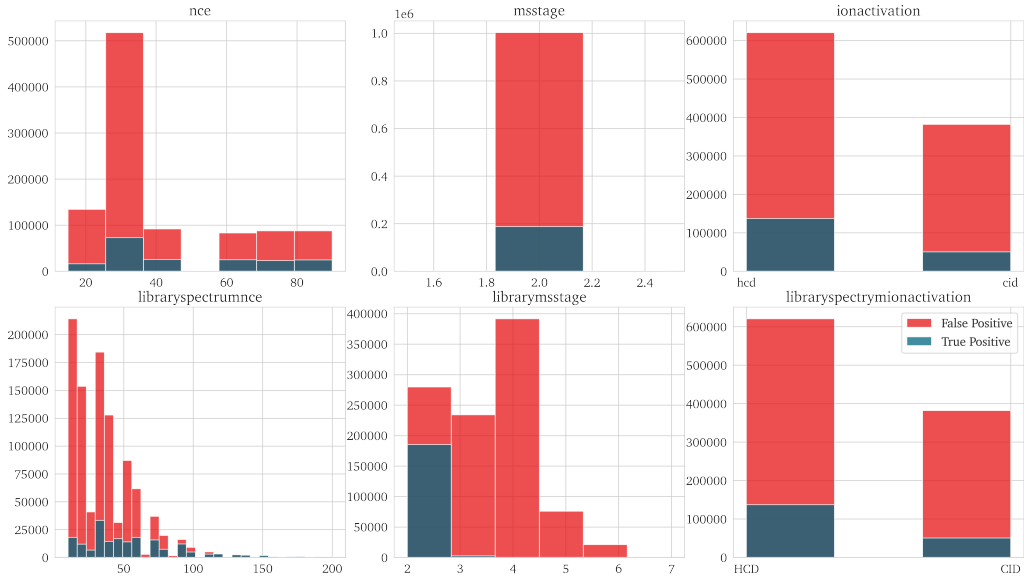


Fig. 3 Histograms depicting the distribution of; the NCEs for; the unknown compound spectra (top left), the known library compound spectra (bottom left), MS stage for; the unknown (top middle), the library candidates (bottom center), and ion activation technique for; the unknown compound spectra (top right), the available library compound spectra (bottom right).

5.3. Procedures (pipeline)

This section provides an in-depth view into the construction of the sequence that orchestrates the data flow from the input to its processed form and output from a machine learning model. The procedure consists of three major parts: data preparation, feature engineering, and machine learning model training (modeling).

5.3.1. Data Preparation

As the first step of the data preprocessing procedure, we treat the missing values (NaNs). The NaNs were initially present in the tables as well as introduced to the data frame after querying the database. Missing values were removed as the visual inspection using histogram did not prove any interesting pattern concerning the subset of the data with the NaNs. Removal of missing values eliminated the representation of some query InChIKeys. A great variety of query spectra is crucial for the training step. Nonetheless, we will predict the binary value of whether the hit was true-positive or false-positive for new query spectra. Therefore, we consider this loss concerning the size of subsets of binary values.

Second, we extracted the first part of the InChIKey (14 characters) describing the chemical formula, number of hydrogen atoms, and charge. The LC-MS/MS is insufficient for isomerism differentiation as written in section 4.2.5 (check 4.2.5 Known problems). Since some queries could have been identified falsely as different isomers, this produces some deviation from the dataset description that was denoted after slash for affected values in 5.2 (check 5.2 Usage of data and data sources).

Third, we used regex to extract metadata from the QueryScanFilter column too. The column's content was described in 5.2.1 (check 5.2.1 Description of queried metadata). Regex provided us with a robust and reusable extraction tool. That searches for the first occurrence of a particular pattern in string based on predefined conditions and not particularly to the preceding number of characters.

5.3.2. Feature engineering

In a feature engineering procedure, we focused mainly on significant information extraction. The engineered features have been selected both on domain knowledge and visual inspection of the distribution of the values.

As of knowledge-based engineering, we established two groups of possible sample aggregations.

At the first level, we introduced the aggregation based on query spectra. One query is represented multiple times based on the number of library hits identified in the list by the similarity scoring algorithm. This information was not available from the point of view of a single dataset record. Similarly, new query spectra will come after the similarity scoring and ranking with the list of candidates.

At the second level, we introduced the aggregation of the library spectra that matched against one query spectrum. We familiarized that in the reference library, N spectra can correspond to one InChIKey. Having one library compound compared with a query spectrum by more than one library spectra can contain valuable information about the confidence of a true hit. Likewise, the algorithm will make a prediction on a list of hits for one query that may involve identical compounds with different library spectra.

Figure 1 gives insight into the structure of the obtained data that serves as an overview of aggregations used and possible ones.

We forged new features by applying the functions over both levels of grouping, namely:

- **The rank of the hit** (rank_query / rank_library)
  Position of the match concerning its score value relative to other samples in a given group. The ranked order provides a metric that is comparable between groups.
- **Size of the group** (num_same_query / num_same_library):
  How many hits belong to a given group. A high value may yield valuable information about the quality of the spectra and their similarity to reference.
- **Mean value of the score** (mean_score_query / mean_score_library):
  Mean score of the hits in a given group. A high mean score could provide a convenient baseline for our confidence re-scoring.
- **Standards deviation of the score** (std_score_query / std_score_library):
  Expresses how much the scores diverge from their mean value among the group.
- **Weighted score by the size of the group** (score*num_query / score*num_library):

Multiplication of the score by the size of the cluster may, in the case of a second-level aggregation, result in enforcement of the score for highly represented compound matches.

Computation introduced other features by calculating the deviation between values of numeric metadata, i.e.

- **The deviation between the mean scores** (dev_mean_score):
  Subtract the mean score of a candidate compound from the query-based baseline.
- **The distance of sample score from mean scores** (dev_score_sample):
  Subtract the score of a sample from the query-based baseline.
- **The proportion of the same candidate hits in the whole list** (div_num):
  The ratio of candidate spectra representing the same compound in the list of candidates given the query spectrum
- **Weighted distance of the score from baseline by the number given by the former proportion** (dev_score*div_num):
- **Weighted score by the ratio** (score*div_num):
- **Same ion activation technique** (same_ion):
  Boolean value that represents match/mismatch between methods used to induce the ion fragmentation
- **The deviation between NCE** (dev_nce):
  Specifies the distance of the experimental conditions that may determine the number of fragments
- **Mean deviation from the NCE** (mean_dev_nce_query):
  The mean value of the distances between NCEs under which we raised the experiments
- **Distance of the NCE deviation from mean NCE given queried compound** (dev_mean_nce):

The feature engineering procedure produced mainly continuous features essential to increase the evaluation score of the trained model. They introduce an infinite space of possible values. Their distributions are presented in Figure 4, respectively.

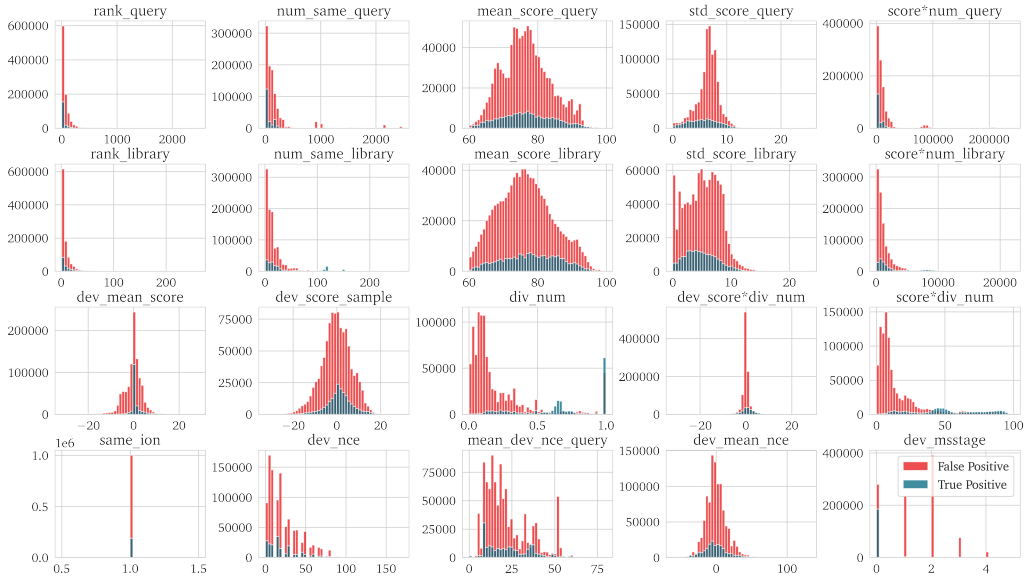


Fig. 4 Histograms that show the balanced distribution of continuous features. Subplot titles represent the acronyms of the features. Each histogram displays false-positive hits (red) and true-positive hits (dark blue) overlapping the former. The x-axis represents the intervals of values, and the y-axis reflects the frequency of samples' occurrence in bins.

However useful it may seem, real-valued properties are harder to interpret and usually must be discretized in the case of a decision tree or Bayesian network.

Value discretization is an essential step toward training some machine learning models. A Bayesian network implementation in pomegranate works natively with discrete values. This fact raises a crucial need to modify continuous features to support future steps of the pipeline. Two approaches approaching categorization are available and used broadly in our research. We denote that the methods can only lead to a loss of information from the continuous features. In a bin-based discretization, we create categories according to the equal distance between the values. Quantile-based discretization forms categories based on the specified frequency of occurrence and favors the uniform distribution.

Bin-based discretization is particularly useful when only a small number of outliers is present. The quantile-based method handles the effect of sparseness and a high number of outliers that do not follow any specific distribution. It results in a uniform distribution of values.

We developed an automated procedure to select a transformation method between continuous-valued variables and discrete-valued. The basis of a routine lies in a simple algorithm shown in Figure 2. This procedure runs over the whole list of features. Firstly, we evaluate the accuracy of a simple decision tree on samples grouped by six bins of values using the cut function (equal-width) and the qcut function (equal-frequency). We select the method providing a higher score. Secondly, we loop through a list of the number of bins incrementing by two. Lastly, we compare the results of new binning with the previous one using the results of action in the first step, updating the number of bins and stopping if the difference is smaller than the termination criterion.

```
def discretize_df(df, features_list, init_bins = 6, stop_criterion = 0.01):

  for each feature in features_list:

    for each discretization_function in [“equal-width”, “equal-frequency”]:
      1. discretization_function(df[feature], init_bins)
      2. score = tree_score(1.)

    if score[equal-width] > score[equal-frequency]:
      discretization_function = “equal-width”
    else:
      discretization_function = “equal-frequency”

    for number_of_bins in bin_list:
      1. discretization_function(df[feature], init_bins)
      2. score = tree_score(1.)

      if new_score - old_score < stop_criterion:
          break
      else
          loop

  return discrete_df
```

Fig. 5 Illustration of an algorithm for automatic feature discretization.

The automation provided a simple framework with two tunable parameters - the termination criterion and the initial number of bins by which the algorithm selects the binning technique. It is necessary to illuminate that the data distribution modulates an algorithm's decision. Yet so influences any of our assumptions based on visualization. We backed the default bin value by an exhaustive search of binning combinations for both functions cut and qcut. Figure 5 draws distributions after feature discretization.

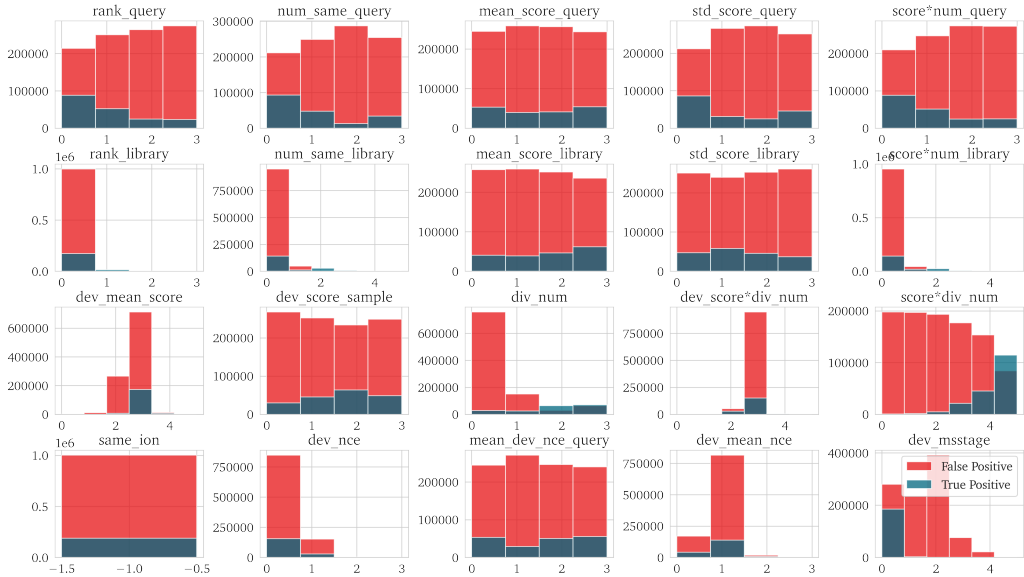


Fig. 6 Histograms that display the distribution of categorical features. Subplot titles represent the acronyms of the features. Each histogram shows the distribution of false-positive hits (red) and true-positive hits (dark blue) overlapping the former. The x-axis represents the intervals of values, and the y-axis reflects the frequency of samples' occurrence in bins.

5.3.3. Modeling

This section examines the creation of an algorithm to classify true-positive and false-positive candidates. The algorithm predicts the label with a certain probability which we assess as the confidence of the prediction. Numerous data-based modeling techniques offer classification properties. In our research, we focused on competitive performance and high interpretability. Therefore we selected two machine learning models for training - Random Forest classifier and Bayesian network. Performance and interpretability are success criteria that generally go against each other. Consequently, the finest tradeoff will depend on fictional weight matrices that we set on the two objects of our triumph.

The correct design of models' testing required domain knowledge and a good understanding of the data. Each sample in the dataset is classified by unknown sample, compound, and experiment from which derived. Random split of data into training and testing sets would pose leakage of candidates from the same experiment into both, and validation would only

provide naive assumptions about the models' ability to generalize. Therefore, the splitting algorithm shall be group aware.

As we presented in Figure 2, the classes are misbalanced. We employ sample stratification to keep the proportion of categories similar to the original dataset in both subsets created.
We built a custom setup for the test design. Based on a function for a randomized split of the data, implemented in the scikit-learn library, we created a splitting algorithm that considers experiments and unbalanced classes when performing separation into train and test split. To the former, the algorithm randomly assigned 75% of the samples.

The features created and presented in section 5.3.2 and the similarity score served as predictors. However, not all the features favored the reduction of estimated prediction error. We performed filtering of the predictors based on the permutation importance on a held-out testing set. This inspection technique can be used for any estimator. By randomly shuffling one predictor at a time and evaluating the decrease in the model's score, one can derive its importance by its extent.

Random Forest Classifier

A random forest classifier implemented using a scikit-learn library in Python offers effortless training and validation of the machine learning algorithm. The highly competitive performance of the method makes it very popular. Moreover, the mean decrease of impurity, procedure of the random forest's construction, can directly illustrate the importance of the used features. It is not the same thing as the explainability of the Bayesian network. Therefore, this aspect cannot be compared. However, the Random Forest classifier can help us assess the possible score we can achieve using the engineered features.

We kept the random forests attributes at default first and changed only the number of estimators from 100 to 10 to improve the execution time. 10-fold cross-validation reckoned the performance of the random forest classifier on the training set.

Firstly, we assessed the feature importance of the trained model by permuting the features one at a time and predicting the F1 score on the testing dataset. This step favors the model's generalization. Moreover, it dramatically affects the training and evaluation speed of the

models employed later. We performed further modeling on the selected subset of the features, namely by their abbreviation (check 5.3.2 Feature Engineering):

- ‘dev_msstage';
- ‘div_num’;
- ‘num_same_query’;
- ‘score*div_num’;
- ‘score*num_query’;
- ‘mean_score_library’;
- ‘mean_score_query’.

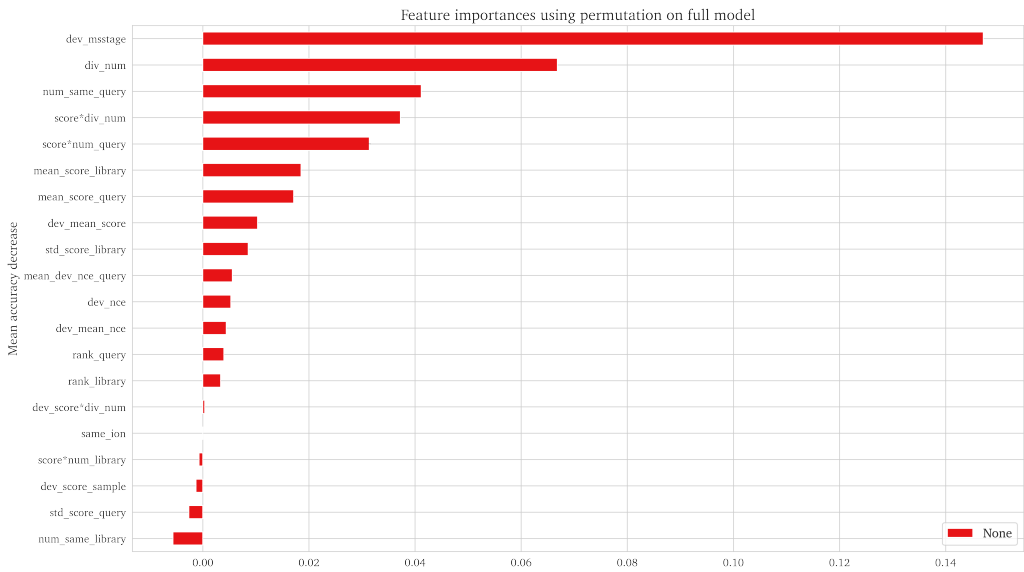


Fig. 7 Horizontal histogram of feature importance assessed by permutation. Positive deviation represents to what extent a randomly shuffled feature decreases the prediction F1 score.

Model Tuning

We tuned the Random Forest classifier’s hyperparameters to improve generalization and adjust the model's complexity. A vast number of hyperparameters define its structure and learning. We selected the number of trees in the forest, the maximum depth of each tree, the minimum number of samples required to spit the tree's nodes, the maximum number of features to consider, the criterion to measure split quality, and class weighting. The Halving Random Search Cross-Validation algorithm tuned the hyperparameters utilizing the following strategy:

1. It trained the model on the training dataset for each consecutive fold;
2. Evaluated the unweighted mean F1 score of each test set of 5-folds cross-validation;
3. Selected half of the best candidates for training on more resources based on the highest score.

The candidate selection is depicted in Figure 8

Tab. 3 Search Space of the Random Forests' hyperparameters

Name of the hyper parameter	Options Grid
Max depth of the trees	[None, 2, 5, 10, 20, 50, 100, 200]
Min no. samples in node	[2, 5, 10, 20, 50]
The number of trees in the forest	range(10, 100, 20]
The number of features to consider for best split	["auto", "log2"]
Criterion to measure quality of the split	["gini", "entropy"]
Class Weights	[{False: 1, True: 5}, {False: 1, True: 2}, {False: 1, True: 1}, {False: 2, True: 1}]

Hyperparameter tuning evaluated a 5-fold Cross-Validation score, starting with 361 candidate models and selecting the best ones over five iterations. The tuning took 4min 24s. A detailed insight into the iterations would not be provided. However, a brief observation follows. The most competitive candidates had a higher weight on true-positive candidates, in general. This made the model more complex and time-consuming to train and evaluate. We highlight that Halving Random Search Cross-Validation is a suboptimal algorithm with no optimality guarantees with the main benefit of a significant decrease in hyperparameter tuning time.

Selected model:

```
RandomForestClassifier(class_weight={False: 1, True: 2},
                       criterion='entropy',
                       max_depth=2,
                       min_samples_split=50,
                       n_estimators=50)
```

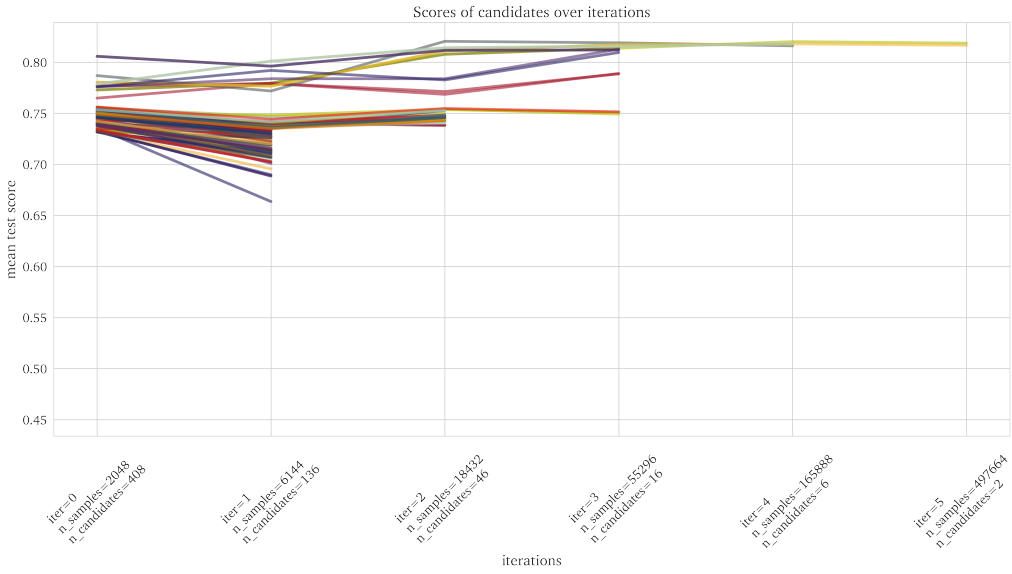


Fig. 8 Hyperparameter tuning candidate scores over iterations. HalvingRandomSearchCV removes one-third of the candidates in each iteration and increases the resources available.

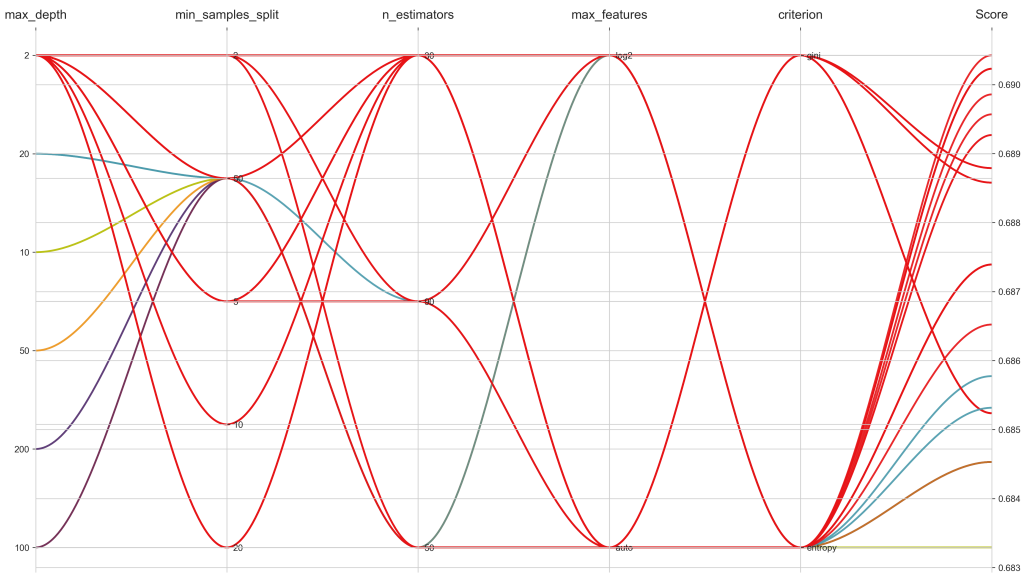


Fig. 9 Parallel plot of the parameters for the models in second iteration.

Bayesian Network

As we learned in section 4.8.1, the learning Bayesian Network represents the structure selection and computation of conditional probabilities. We use the features marked as the most important by the Random Forest classifier. It is crucial to declare that the feature importance is model specific, and the dropped features may yield important insight into the underlying patterns for the Bayesian Network. However, we were bound to train on the shrunk feature space to reduce resource consumption and achieve feasible training times. We selected five best predictors based on their permutation importance (Figure 7), namely: ‘dev_msstage'; ‘div_num'; ‘num_same_query'; ‘score*div_num'; ‘score*num_query' (check 5.3.2 Feature Engineering). Firstly, we perform the structure selection manually and evaluate it on the data.

The knowledge about the predictors’ sources gave basis to the structure of the model. The features retrieved from the database had no parent nodes and directly resulted in the true-positive/false-positive assessment. The engineered features had parent nodes corresponding to the predictors they were composed of. Figure 10 depicts the structure of the manually created model’s diagram.

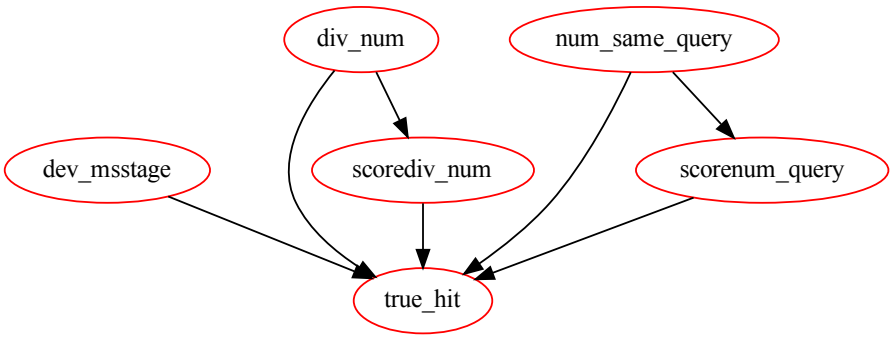


Fig. 10 Manually created directed acyclic graph of Bayesian Network. Blocks represent nodes of the diagram. Root nodes represent a discrete distribution; other nodes represent conditional probabilities of the predictor given to the parents. The direction of the arrow depicts conditional dependence.

Secondly, we automatically assessed the structure as well and compared the performance. The exact computerized data-driven approach required exponential time for the number of

variables. The pomegranate library implements a novel greedy algorithm to mitigate this problem, significantly reducing fitting time. This procedure renders the desired outputs as evidence, which is inherently wrong. However, it does not restrict the prediction of the true-false hit and confidence assessment.

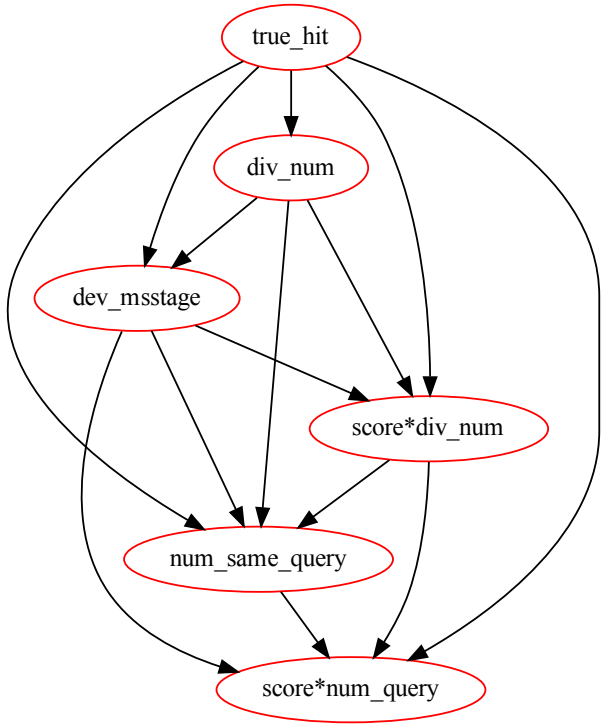


Fig. 11 Automatically initialized directed acyclic graph of Bayesian Network. Blocks represent nodes of the diagram. Root nodes represent a discrete distribution; other nodes represent conditional probabilities of the predictor given to the parents. The direction of the arrow depicts conditional dependence.

5.4. Methods used to evaluate and interpret the results

This section looks broadly at the model's performance assessment according to the business needs. Several evaluation metrics exist that deliver a single-valued score representing the model's ability to identify a class that was known but hidden during prediction.

High awareness should be present during metric selection. The data science community considers the F1 score the suavest, combining information about the model's precision and sensitivity with robust mitigation of the imbalanced classes effect.

$$F1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} \tag{13}$$

Where:

$$PPV = \frac{TP}{TP + FP} \tag{12}$$

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

TP – true positive; FP – false positive; FN – false negative

Different training data subsets, used for learning over several iterations, assess the model's generalization to an independent data set. This procedure is called cross-validation. We perform a 10-fold cross-validation to support or reject the generalization hypothesis. Further, we evaluated the performance on a withheld data set for validation.

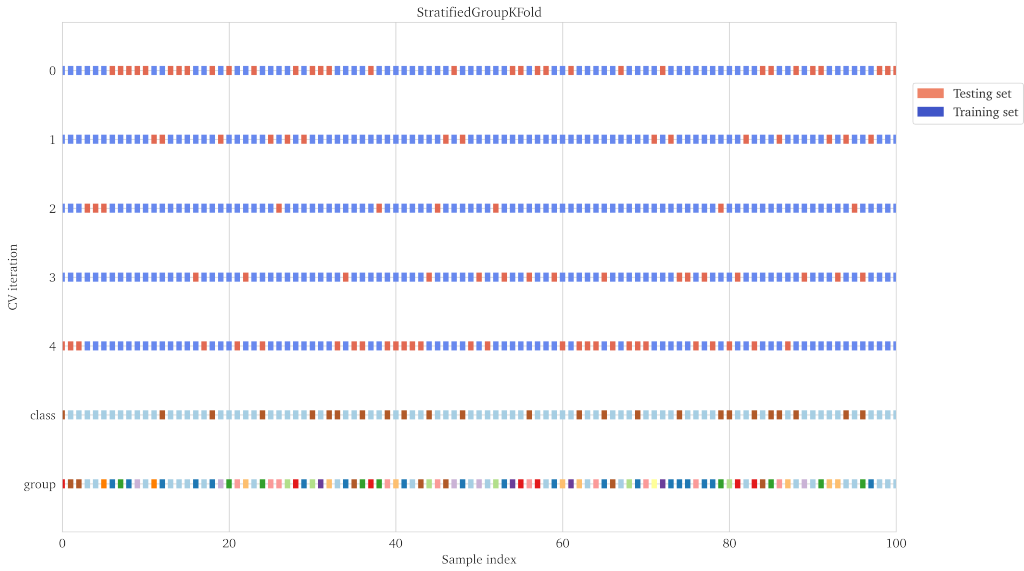


Fig. 12 Illustration of stratified grouped K-fold cross-validation on 5-folds

The confusion matrix allows better correct and incorrect predictions assessment. It represents a contingency table that plots the model's predictions that are:

- True-positive hits (TP) - correctly assess the candidate as being the right candidate;
- True-negative hits (TN) - accurately indicate the candidate as wrong;
- False-positive hits (FP) - incorrectly mark the candidate as being the right candidate although being wrong in reality;
- False-negative hits (FN) - incorrectly classify the candidate as being wrong.

Tab. 4 Illustration of the confusion matrix

		Predicted Label [%]	
Number of Samples P+N		True Hit	False Hit
True Label	True Hit	TP	FN
	False Hit	FP	TN

5.4.1. Visualization

The crucial step of any performance examination is its good visualization. Graphical interpretation sheds light on any strengths and weaknesses of the trained model.
In general, we used three plots to present and compare the results.

The Rank vs. Accuracy line plot sheds light on an essential property of library scoring and candidate retrieval. The rank on the x-axis represents the candidates' hierarchy in the list, sorted either by the confidence of the prediction or the similarity score of the cosine algorithm. The percent of correct hits represents the number of true-positive in all retrieved candidates for a given rank and better, given by Equation 14. The practical benefit of this evaluation is a direct assessment of the rescoring's quality.

$$ACC = \frac{TP + TN}{TP + TN + TP + FN} \tag{14}$$

The Precision vs. Recall line plot shows the model's performance from the perspective of various thresholds based on which the model made its decisions. It iterates through scores incrementing their value in every loop. Every iteration computes precision, the ratio between true-positive and all-positive values, given by Equation 12. This metric gives us a clue about how many relevant samples were retrieved given the threshold on the score. On the x-axis, we draw the recall, the measure of how many correctly identified samples model marked as relevant, provided by Equation 13.

The ROC-AUC curve represents a line plot that depicts the model's ability to discriminate true-positive hits from false-positive hits for various decision probability thresholds. We consider the AUC (**Area Under The Curve**) as the degree of separability between classes and the ROC (**Receiver Operating Characteristics**) curve to reckon the probability of pinpointing the candidate correctly. The x-axis shows the false-positive rate given by Equation 15. It represents the proportion of candidates wrongly assessed as correct. The y-axis represents the true-positive rate (TPR) or sensitivity, provided by Equation 13. TPR gives a clue about the proportion of right candidates retrieved from the sample space. The interpretation of separability by AUC in a probabilistic notion is that randomly selected spectra, one for each class, will result in the correct candidate ranking higher than the incorrect.

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

$$AUC = \int_{x=0}^{1} TPR(FPR^{-1}(x))\, dx \tag{16}$$

6. Results

The results obtained in our thesis emerge from the pipelines employed on the combined dataset of Schymanski's authentic compounds and blood groups identified using the HILIC/ Reversed-Phase. We described the dataset in section 5.2. Usage of data and data sources and defined the data processing steps in the preceding sections. Any part of the pipeline that employs randomness was set to a random state of 0 for the reproducibility of our findings. Models were trained following the best practices, drawn from the literature, and briefly collected in section 4.

Three Random Forest classifiers were introduced in subsection 5.3.3. The unbalanced mean of the F1 score over correct candidates and incorrect candidates was computed and evaluated in Table 5. The first two columns belong to the Random Forest classifier trained on the whole space of predictors, in both continuous and discrete fashion. We can see that both models have high F1 scores on training data. However, they do not generalize well.

By introducing only the essential features to the models selected by their permutation importance (check 5.3.3 Modeling), we sought to alter the generalization of the models. The feature selection slightly reduced the training score of the model trained on categorical features.

We were tuning the model with the aim of cross-validation improvement and, therefore, increased generalization capacity. This process introduced a boost in cross-validation scores. Nevertheless, in the case of the model trained on the entire space of predictors, a significant decrease in the validation and train scores occurred. The training and evaluation time raised significantly. Overall, the models trained on categorical data showed better validation scores. This observation led us to conclude that the model overfits training data and learns the underlying relationships central to the experiments it saw. The discretization of predictors reduces their information content and helps to mitigate overfitting.

Tab. 5 Unweighted mean F1 Score of all classes evaluated for all models. The column headers represent the tags of different classifiers. The first two columns represent the default model trained on the whole predictors' space.

	Random Forest All	Random Forest Cat. All	Random Forest	Random Forest Cat.	Random Forest Tuned	Random Forest Cat. Tuned
Train Score [%]	99.90	93.17	98.86	85.75	45.28	80.08
10-Fold CV Score [%]	61.49	65.38	62.80	63.46	69.34	68.73
Validation Score [%]	71.25	72.91	68.49	72.49	46.42	82.61
Train time [s]	5.31	0.92	2.28	0.69	11.1	4.94
Validation Time [s]	0.57	0.53	0.38	0.39	2.67	5.3

* Cat. abbreviates the usage of categorical features.

The confusion matrix in Table 6 assessed the candidates' identification performance of the best setup. The significantly unbalanced nature of the data is evident in this graph. The model mislabeled around one-fourth of the correct candidates. 4.68% of the incorrect candidate spectra were labeled correct, representing one-third of all positively labeled data.

Tab. 6 Confusion matrix of validation performance of tuned Random Forest classifier trained on selected categorical features

		Predicted Label [%]	
Number of Samples 449 961		True Hit	False Hit
True Label	True Hit	9.73	3.64
	False Hit	4.68	81.95

Bayesian Network was used as a classifier aiming for the improved explainability of the decision. We developed two models of Bayesian Network as depicted in section 5.3.3 (check 5.3.3 Modeling). Table 7 displays the evaluation of the trained model on data. Enormous validation time due to the maximum likelihood estimation evaluated for each sample makes this approach non-deployable and drives the need for an explicit solution. Moreover, the automatically assessed causality of the Bayesian Network raises suspicion and requires an in-depth examination of novel greedy search methods.

Tab. 7 Unweighted mean F1 Score of all classes evaluated for all models. The column headers represent the tags of different classifiers. The first two columns represent the default model trained on the whole predictors' space.

	BayesianNetwork Tuned	Bayesian Network Manual
Train Score [%]	81.45	45.25*
Validation Score [%]	73.40	59.31
Train time [s]	1.77	0.00
Validation Time [s]	3977	4099

* 10 000 samples of training dataset

The confusion matrix, depicted in Table 8, points to the model's inability to correctly label the valid candidates more than half of the time.

Tab. 8 Confusion matrix of tuned Bayesian Network trained on selected categorical features.

		Predicted Label [%]	
Number of Samples 449 961		True Hit	False Hit
True Label	True Hit	6.49	6.87
	False Hit	4.51	82.12

6.1.1. Visualization

Firstly, we established a rank vs. accuracy graph (check Figure 13). The accuracy is drawn concerning the unknown candidates and not individual spectra (check Figure 1). We can see that both Random Forest models perform the task of re-scoring the candidates correctly and give us the correct candidate, if present, as the first ranking more often. A slight positive deviation of Random Forest trained on categorical data within the first ranks provides us with a clue about the information loss that favors the model's generalization. Tuned Bayesian Network with computed structure improved the accuracy of the first ranking candidates. Bayesian Network with a design based on feature creation flow could not correctly re-score the candidates. A dark blue line represents the impact of randomly shuffling candidates. All rescoring models could potentially increase the confidence in the correct classification by the first ranking candidate to a much greater extent.

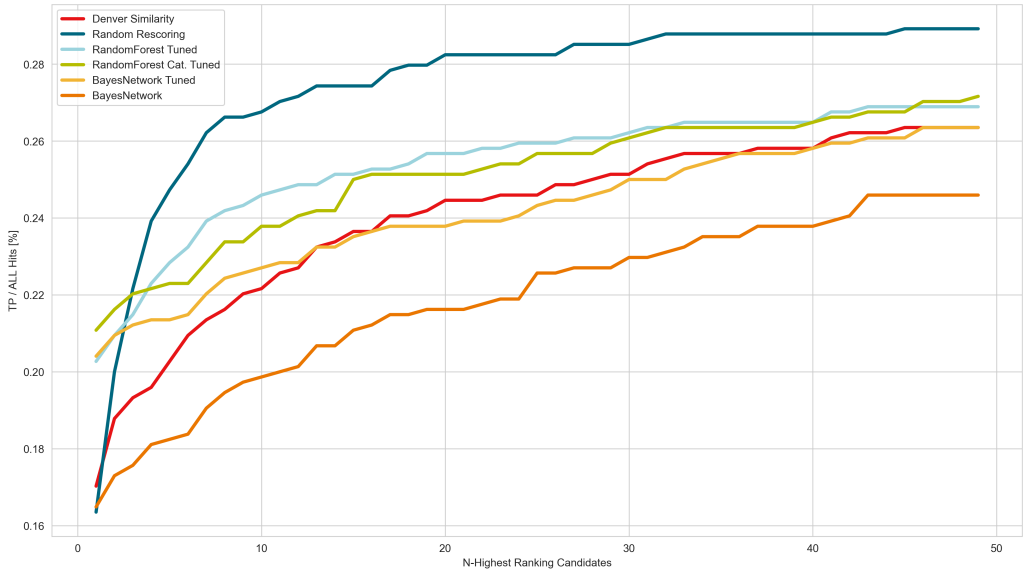


Fig. 13 Line plot where the x-axis represents n-highest ranking candidates selected to evaluate the scoring accuracy on the y-axis (TP / (TP+FP+TN+FN)). Rank means the hierarchy of candidates by; cosine similarity scoring (red), random scoring (dark blue), Random Forest re-scoring (light blue), Random Forest re-scoring on categorical features (green), Bayesian Network with computed structure (yellow), and Bayesian Network with manually selected DAG (orange). The x-axis was truncated to the first 50-highest ranking candidates (max. 302). For all candidates, the value approaches 80.39%.

Secondly, we evaluated precision vs. recall regarding the sample score threshold. Figure 14 shows that while the highest Denver scoring recalls a tiny portion of all-positive candidates with a precision of around 15%, both Random Forest classifier and Random Forest classifier on continuous features achieve higher precision-recall for the best scoring candidates. Bayesian Network cannot confidently (with a high score) assess the candidates, and the model's precision is low for the high scores. Bayesian Network from Figure 10 decreases the precision of the best candidates by rescoring. Random rescoring shows nearly constant precision over the whole recall and various scoring thresholds.

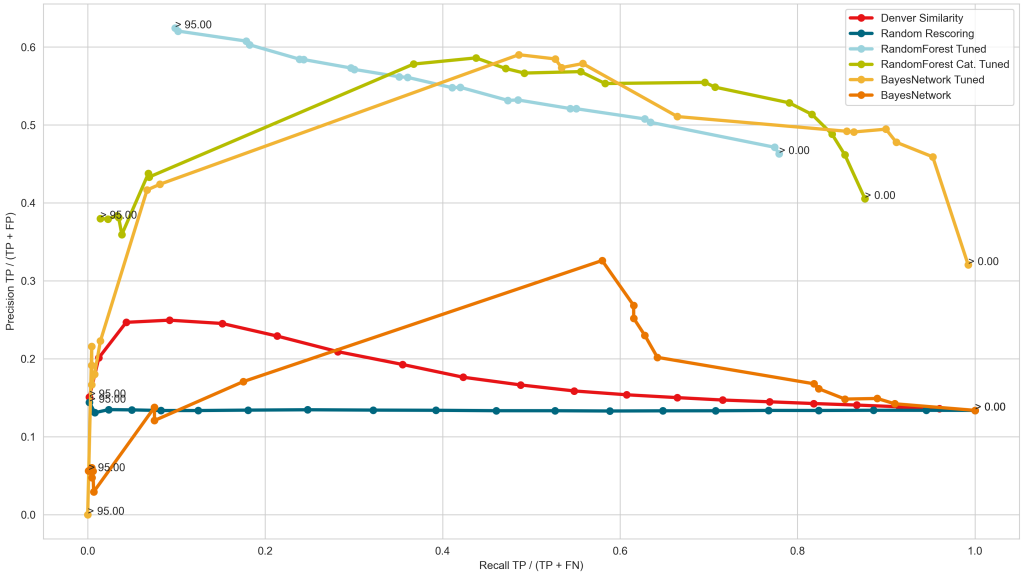


Fig. 14 Line plot where the x-axis represents recall or the proportion of true-positives retrieved from all-positive hits. The y-axis depicts the precision, or how many retrieved candidates are relevant. Dots represent the score thresholds decrementing by 5% from left to right, for which precision-recall was obtained. Scores are given by; cosine similarity scoring (red), random scoring (dark blue), Random Forest re-scoring (light blue), Random Forest re-scoring on categorical features (green), Bayesian Network with computed structure (yellow), and Bayesian Network with manually selected DAG (orange).

Thirdly, we evaluate the false-positive rate vs. the true-positive rate regarding the similarity score and AUC. All the models except the Bayes Network from Figure 10 increase the true-positive rate to the false-positive rate proportion and render a higher AUC value. In other words, by randomly selecting two candidates, one labeled correct and one incorrect, we would increase the probability of correct judgment by former classifiers.

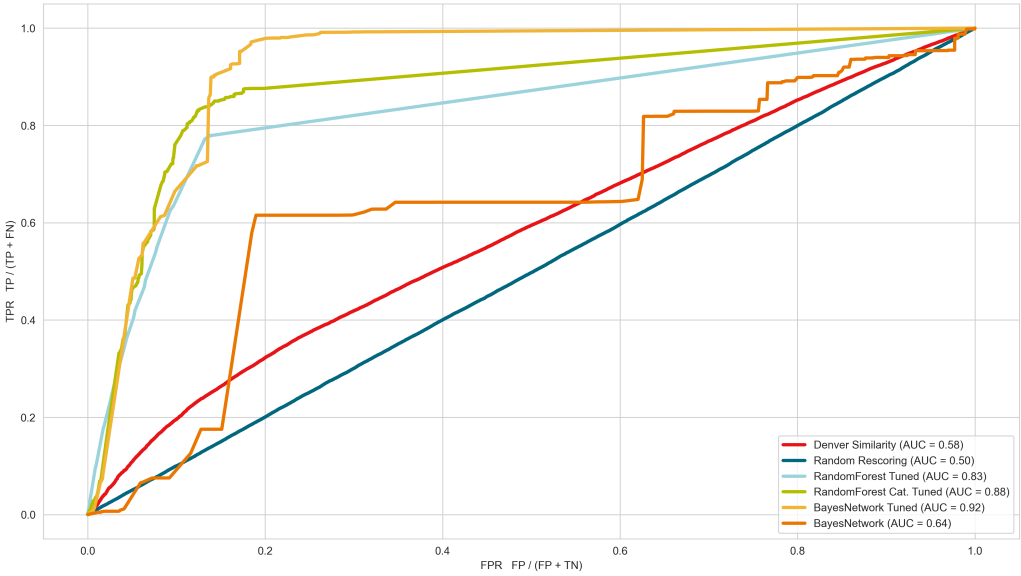


Fig. 15 Line plot where the x-axis represents the false-positive rate (FPR). The y-axis depicts the true-positive rate (TPR). The ROC AUC computes FPR-TPR for variable score thresholds. Scores are given by; cosine similarity scoring (red), random scoring (dark blue), Random Forest re-scoring (light blue), Random Forest re-scoring on categorical features (green), Bayesian Network with computed structure (yellow), and Bayesian Network with manually selected DAG (orange).

7. Summary and Discussion

In the thesis, we aimed to design, implement and optimize a classification algorithm for small molecule identification using annotated spectra trees. Section 3 presented three separate components of our focus. In general, we aimed to perform automatic re-scoring of the candidates retrieved by the similarity scoring algorithm based on the available experimental metadata. The summary will briefly explain the progress made in each area, respectively.

Firstly we explored the current state of research in small molecule identification and structural elucidation based on spectra similarity. Section 2 provided a brief overview of existing state-of-the-art methods approaching this problem *in silico* and *in vitro*. We found out that the focus and development reach far and wide. Although *in silico* methods seem to be central to any research progress, we can assume that there are many open endings. Alternative approaches to the most common *in vitro* identification method, namely, cosine similarity searching against the library of carefully curated spectra, expose some bottlenecks. The Spec2Vec, as the most promising library searching and scoring algorithm, proved to be very time-consuming and non-deployable on the vast datasets exclusively available to Thermo Fisher Scientific.

The availability of a great dataset of curated and annotated spectra gives the owner a competitive advantage and allows the potential of rigorous similarity scoring to be fully engaged. We learned that the similarity scoring algorithm deployed on the vast library at our disposal retrieves a list of candidate spectra spanning from one up to hundreds for each unknown spectrum. Currently, the analyst experimenting has to carefully select the candidate while considering its similarity score and other readily available parameters. Such parameters contain, for instance, a match of experimental conditions of unknown spectrum and library spectra acquisition and several candidate spectra that belong to one compound on a library side. Statistical machine learning conducted on a database of countless experiments can yield a definition of the underlying relationships that differentiate correct candidates from incorrect ones. Our novel classification algorithm for small molecule identification presents an updated scoring and ranking system that considers the annotation of query-library spectra matches. New scoring can retrieve a higher proportion of correct candidates at the first rank thanks to the combination of rigorous similarity scoring and data-driven model stacking. Two attributes determined the data-driven model's selection.

The first was the competitive performance of the model; the second was the possibility of explaining its decision. We employed Random Forest Classifier as a model of reference. It showed higher evaluation scores and the opportunity to explain the importance of provided features based on permutation and the mean decrease in impurity.
We used Bayesian Networks as classifiers that offer a better tradeoff between the two requirements. The Bayesian Network solves a common problem of the system's decision questioning by introducing an utterly explainable framework, utilizing evidence-based reasoning. However, greedy usage of resources and extended execution time for testing and prediction poses an unresolved critical bottleneck. Moreover, great awareness should be present when selecting the model's design. As we learned, the structure assessment based on the data understanding does not always yield good accuracy.

Resolvement of this problem creates a potential for further development in the field. The availability of numerous libraries offers a wide range of possible baselines. We consider the prediction time as the most critical problem. Caching the optimization solutions behind prediction results can significantly improve prediction times. The proposed explicit solution will allocate the memory block to store the solution, which could be further used to evaluate new query-library candidate spectra swiftly. The size of the memory block will depend on the number of predictors and discrete categories.

Above all, exploring other available metadata and exhaustive feature engineering may be a critical step toward further improvement in the domain of library searching and scoring. The ultimate aim is to select the correct candidate and, therefore, identify the unknown compound with confidence, completely removing the need for further analysts' intervention. High confidence in identification will lower the false-positive rate having a critical impact on various fields of chemistry and biology. The former will be central to further improvement in the area. This goal can only be achieved where analytical chemistry and informatics cooperate.

8. Resumé

V práci sme sa zamerali na návrh, implementáciu a optimalizáciu klasifikačného algoritmu na identifikáciu malých molekúl pomocou anotovaných stromov spektier. Ciele práce pozostávali na najvyššej úrovni z troch konkrétnych krokov.

Prvým bolo skúmanie a analýza súčasného pokroku v identifikácii malých molekúl a uvedenie najmodernejších algoritmov, ktoré riešia tento problém pomocou experimentálnych techník in vitro aj in silico.

Po druhé, sme predstavili nový klasifikačný algoritmus na identifikáciu malých molekúl. Vyvinutý algoritmus používa anotovaný zoznam zhôd neznámeho spektra a spektier z referenčnej knižnice na základe rigorózneho skóre podobnosti. Výstupom nami vyvinutého algoritmu je nové skóre spoľahlivosti pre každú vzorku v súbore údajov na základe príslušného modelu strojového učenia. Nové skóre spoľahlivosti slúži zníženiu pravdepodobnosti falošne pozitívnej identifikácie chemických zlúčenín. Zvažuje pri tom odchýlku medzi experimentálnymi parametrami spektier. Ďalej poskytujeme podrobné informácie o každom kroku vývoja algoritmu na základe postupu CRISP-DM, široko využívaného v sfére dátových vied.

Po tretie, konfrontujeme problém s interpretovateľnosťou modelu strojového učenia založeného na dátach a predstavujeme algoritmus vyhľadávania a hodnotenia založený na Bayesovej sieti. Vvýkonnosť vyvinutého algoritmu sme overili prostredníctvom selektivity a citlivosti v poskytnutom súbore údajov pomocou klasifikačných metrík (napr. ROC, AUC, F1 skóre).

V práci sme uviedli stručný prehľad existujúcich najmodernejších metód, ktoré k tomuto problému pristupujú in silico a in vitro. Hoci sa metódy in silico zdajú byť kľúčové pre akýkoľvek pokrok vo výskume, predpokladáme veľké množstvo výziev, ktorým toto smerovanie výskumu čelí. Alternatívne prístupy k najbežnejšej metóde identifikácie in vitro, konkrétne k hľadaniu podobnosti neznámeho spektra oproti knižnici starostlivo upravených spektier, odhaľujú niektoré prekážky. Spec2Vec, ako najsľubnejší algoritmus na vyhľadávanie a skórovanie knižníc, sa ukázal ako veľmi časovo náročný a nenasaditeľný na rozsiahlych súboroch údajov, ktoré má spoločnosť Thermo Fisher Scientific exkluzívne k dispozícii.

Spoznaním kontextu vyhľadávania v referenčnej knižnici sme sa dozvedeli, že algoritmus hodnotenia podobnosti nasadený v rozsiahlej knižnici, ktorú máme k dispozícii, získava zoznam kandidátskych spektier v rozsahu od jednej do stoviek pre každé neznáme spektrum. V súčasnosti musí analytik, ktorý experimentuje, starostlivo vybrať kandidáta, pričom musí zvážiť jeho skóre podobnosti a ďalšie ľahko dostupné parametre. Takéto parametre obsahujú napríklad zhodu experimentálnych podmienok neznámeho spektra a získavanie spektier knižnice a niekoľko kandidátskych spektier, ktoré patria jednej zlúčenine na strane knižnice. Štatistické strojové učenie vykonávané na databáze nespočetných experimentov môže poskytnúť definíciu základných vzťahov, ktoré odlišujú správnych kandidátov od nesprávnych. Náš nový klasifikačný algoritmus na identifikáciu malých molekúl predstavuje aktualizovaný systém hodnotenia a hodnotenia, ktorý zohľadňuje anotáciu zhôd spektier dopytov a knižníc. Nové bodovanie môže získať vyšší podiel správnych kandidátov na prvom mieste vďaka kombinácii prísneho bodovania podobnosti a skladania modelov na základe údajov. Výber modelu založeného na údajoch určovali dva atribúty. Prvým bol výkon modelu; druhým bola možnosť vysvetliť svoje rozhodnutie. Ako referenčný model sme použili Random Forest Classifier. Preukázal vyššie hodnotiace skóre a možnosť vysvetliť dôležitosť poskytovaných funkcií na základe permutácie a priemerného poklesu nečistôt, zle klasifikovaných vzoriek.
Bayesovské siete sme použili ako klasifikátory, ktoré ponúkajú lepší kompromis medzi týmito dvoma požiadavkami. Bayesiánska sieť rieši bežný problém spochybňovania rozhodnutí dátovo založeného systému zavedením úplne vysvetliteľného rámca, ktorý využíva argumentáciu založenú na štatistických dôkazoch. Nenásytné využívanie zdrojov a predĺžený čas testovania a vyhodnocovania vzoriek však predstavujú kritické miesto. Okrem toho by sa pri výbere dizajnu modelu malo dbať na veľkú pozornosť. Ako sme sa dozvedeli, posúdenie štruktúry založené na pochopení údajov neprináša vždy dobrú presnosť.

Vyriešenie tohto problému vytvára potenciál pre ďalší rozvoj v odbore. Dostupnosť mnohých programových balíkov ponúka dobrý štart do ďalšieho vývoja. Za najkritickejší problém považujeme čas predikcie. Ukladanie optimálnych riešení výsledkov predikcie môže predstavovať obrovské zlepšenie jej časov. Navrhované explicitné riešenie pridelí pamäťový blok na uloženie riešenia, ktoré by sa mohlo ďalej použiť na rýchle vyhodnotenie nových kandidátskych spektier z referenčnej knižnice. Veľkosť pamäťového bloku bude závisieť od počtu prediktorov a diskrétnych kategórií.

Predovšetkým, skúmanie ďalších dostupných metadát a vyčerpávajúce inžinierstvo prediktorov môže byť kritickým krokom k ďalšiemu zlepšeniu v oblasti vyhľadávania a

hodnotenia neznámych látok pomocou referenčnej knižnice. Konečným cieľom je vybrať správneho kandidáta, a teda s istotou identifikovať neznámu zlúčeninu, čím sa úplne odstráni potreba d'alšieho zásahu analytikov. Vysoká dôvera v identifikáciu zníži mieru falošne pozitívnych výsledkov, čo má kritický vplyv na rôzne oblasti chémie a biológie. Tento cieľ možno dosiahnuť len v spolupráci analytickej chémie a informačných technológií.

References

1. WANG, F. – LIIGAND, J. – Tian, S. et al. CFM-ID 4.0: More Accurate ESI-MS/MS Spectral Prediction and Compound Identification. Anal. Chem. 2021, 93, 34, 11692–11700. 2021 [CrossRef]

2. DJOUMBOU-FEUNANG, Y. – PON, A. – KARU, N. et al. CFM-ID 3.0: Significantly Improved ESI-MS/MS Prediction and Compound Identification. *Metabolites* 2019, *9*, 72. [CrossRef]

3. RUTTKIES, C. – SCHYMANSKI, E.L. – WOLF, S. et al. MetFrag relaunched: Incorporating strategies beyond in silico fragmentation. J. Cheminform 8. 2016. [CrossRef]

4. DÜHRKOP, K. – SHEN, H. – MEUSEL, M. et al. Searching molecular structure databases with tandem mass spectra using CSI: FingerID. Proc Natl Acad Sci U S A. 2015 Oct 13;112(41):12580-5. [CrossRef]

5. LUDWIG, M. – DÜHRKOP, K. – BÖCKER, S. Bayesian networks for mass spectrometric metabolite identification via molecular fingerprints. Bioinformatics (Oxford, England). 34. i333-i340. 2018. [CrossRef]

6. SCHEUBERT, K. – HUFSKY, F. – BÖCKER, S. Computational mass spectrometry for small molecules. *J Cheminform* 5, 12. 2013. [CrossRef]

7. DÜHRKOP, K. – NOTHIAS, LF. – FLEISCHAUER, M. et al. Systematic classification of unknown metabolites using high-resolution fragmentation mass spectra. *Nat Biotechnol* 39, pp. 462–471 (2021). [CrossRef]

8. BLAŽENOVIĆ, I. – KIND, T. JI, J. – FIEHN, O. Software Tools and Approaches for Compound Identification of LC-MS/MS Data in Metabolomics. *Metabolites* 2018, *8*, 31. [CrossRef]

9. KÄLL, L. – CANTERBURY, J. – WESTON, J. et al. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat Methods* 4, 923–925. 2007. [CrossRef]

10. HUBER, F. – RIDDER, L. – VERHOEVEN, S. et al. Spec2Vec: Improved mass spectral similarity scoring by learning structural relationships. PLoS Comput Biol 17(2): e1008724. 2021. [CrossRef]

11. HUBER, F. – VAN DER BURG, S. – VAN DER HOOFT, J.J.J. et al. MS2DeepScore: a novel deep learning similarity measure to compare tandem mass spectra. *J Cheminform* 13, 84 (2021). [CrossRef]

12. STEIN, S. Mass Spectral Reference Libraries: An Ever-Expanding Resource for Chemical Identification. *Analytical Chemistry*, vol. 84, no. 17, Sept. 2012, pp. 7274–82. [CrossRef]

13. KIND, T, et al. Identification of Small Molecules Using Accurate Mass MS/MS Search. *Mass Spectrometry Reviews*, vol. 37, no. 4, July 2018, pp. 513–32. [CrossRef]

14. SPARKMAN, O.D. Evaluating Electron Ionization Mass Spectral Library Search Results. *Journal of the American Society for Mass Spectrometry*, vol. 7, no. 4, Apr. 1996, pp. 313–18. [CrossRef]

15. SCHEUBERT, K. et al. Computational Mass Spectrometry for Small Molecules. *Journal of Cheminformatics*, vol. 5, no. 1, Mar. 2013, p. 12. [CrossRef]

16. ZOLG, D. P. et al. INFERYS Rescoring: Boosting Peptide Identifications and Scoring Confidence of Database Search Results. *Rapid Communications in Mass Spectrometry: RCM*, May 2021, p. e9128. [CrossRef]

17. WANG, J. et al. Calibr Improves Spectral Library Search for Spectrum-Centric Analysis of Data Independent Acquisition Proteomics. *Scientific Reports*, vol. 12, no. 1, Feb. 2022, p. 2045. [CrossRef]

18. YE, D. et al. Open MS/MS Spectral Library Search to Identify Unanticipated Post-Translational Modifications and Increase Spectral Identification Rate. *Bioinformatics (Oxford, England)*, vol. 26, no. 12, June 2010, pp. i399-406. [CrossRef]

19. ENG, J. K. et al. An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *Journal of the American Society for Mass Spectrometry*, vol. 5, no. 11, Nov. 1994, pp. 976–89. [CrossRef]

20. HOFFMANN, E. – STROOBANT, V. Mass Spectrometry: Principles and Applications, 3rd Edition. Wiley Online Library, 2007. ISBN: 978-0-470-03310-4.

21. GROSS, J. H. Introduction. In: Mass Spectrometry. Springer, Cham, 2017. ISBN: 978-3-319-54398-7. [CrossRef]

22. HIRAOKA, K. Fundamentals of Mass Spectrometry. Springer, New York, NY, 2013. {pp. X-X}. ISBN: 978-1-4614-7233-9. [CrossRef]

23. DE VIJLDER, T. – VALKENBORG, D. – LEMIÈRE, F. et al. A tutorial in small molecule identification via electrospray ionization-mass spectrometry: The practical art of structural elucidation. *Mass Spec Rev*. 2018; 37: 607– 629. [CrossRef]

24. MCNAUGHT, A. D. – WILKINSON, A. IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"). Blackwell Scientific Publications, Oxford (1997). Online version (2019-) created by S. J. Chalk. ISBN 0-9678550-9-8. [CrossRef]

25. KELLER, B. O. et al. Interferences and Contaminants Encountered in Modern Mass Spectrometry. *Analytica Chimica Acta*, vol. 627, no. 1, Oct. 2008, pp. 71–81. [CrossRef]

26. MATHUR, R. – OCONNOR, P. B. Artifacts in Fourier Transform Mass Spectrometry. *Rapid Communications in Mass Spectrometry: RCM*, vol. 23, no. 4, Feb. 2009, pp. 523–29. [CrossRef]

27. SUMNER, L. W. et al. Proposed Minimum Reporting Standards for Chemical Analysis. *Metabolomics*, vol. 3, no. 3, Sept. 2007, pp. 211–21. [CrossRef]

28. BRISTOW, A. W. T. et al. Reproducible Product-Ion Tandem Mass Spectra on Various Liquid Chromatography/Mass Spectrometry Instruments for the Development of Spectral Libraries. *Rapid Communications in Mass Spectrometry*, vol. 18, no. 13, July 2004, pp. 1447–54. [CrossRef]

29. THE, M. 2018. Statistical and machine learning methods to analyze large-scale mass spectrometry data. Doctoral Thesis. ISBN 978-91-7729-967-7.

30. IGUAL, L. SEGUÍ, S. (2017). Introduction to Data Science. In: Introduction to Data Science. Undergraduate Topics in Computer Science. Springer, Cham. [CrossRef]

31. VIJAY, K. – BALA D. Chapter 2 – Data Science Process. In: Data Science (Second Edition), 2019. Pages 19-37. ISBN 9780128147610. [CrossRef]

32. TRIPATHI, S. – Muhr, D. – Brunner, M. et al. Ensuring the Robustness and Reliability of Data-Driven Knowledge Discovery Models in Production and Manufacturing. Frontiers in Artificial Intelligence. 2021. [CrossRef]

33. NINO, M. et al. Business Understanding, Challenges and Issues of Big Data Analytics for the Servitization of a Capital Equipment Manufacturer. 2015, pp. 1368-1377. [CrossRef]

34. When things matter: A survey on data-centric internet of things. [CrossRef]

35. MÜLLER, A. C. & GUIDO, S. *Introduction to machine learning with Python: a guide for data scientists*. ISBN: 9781449369415. 2017.[CrossRef]

36. JAMES, G. – WITTEN, D. HASTIE, T. et al. 2017. An Introduction to Statistical Learning: with Applications in R. Springer New York, 2014. ISBN 1461471370.

37. DARWICHE, A. (2008). Chapter 11 Bayesian Networks. F. van Harmelen, V. Lifschitz, & B. Porter, Handbook of Knowledge Representation. pp. 467–509. [CrossRef]

38. NEAPOLITAN. R.E. (2003). Chapter 1 BASICS OF PROBABILITY THEORY. R. E. Neapolitan, Learning Bayesian Networks. (pp. 3–64). ISBN: 9780123704771

39. HASTIE, T. – TIBSHIRANI, R. – FRIEDMAN, J. The Elements of Statistical Learning. *Data Mining, Inference, and Prediction.* 2009. [CrossRef]

40. JAMES, G. WITTEN, D. HASTIE, T. TIBSHIRANI, R. (2013). Introduction. In: An Introduction to Statistical Learning. Springer Texts in Statistics, vol 103. Springer, New York, NY. [CrossRef]

41. JAMIESON, K. – Talwalkar, A.. (2016). Non-stochastic Best Arm Identification and Hyperparameter Optimization. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, in *Proceedings of Machine Learning Research* 51:240-248. [CrossRef]

42. SCUTARI, M. Who Learns Better Bayesian Network Structures: Accuracy and Speed of Structure Learning Algorithms. 2019. pp. 235-253, ISSN 0888-613X. [CrossRef]

43. YUHONG, G. – RUSS, G. Discriminative model selection for belief net structures. In *Proceedings of the 20th national conference on Artificial intelligence – Volume 2*(*AAAI05*). AAAI Press, pp. 770–776. 2005.

44. Chickering, David & Meek, Christopher & Heckerman, David. (2012). Large-Sample Learning of Bayesian Networks is NP-Hard. Journal of Machine Learning Research. 5.

45. LARRAÑAGA, P. – CALVO, B. – SANTANA, R. et al. Machine learning in bioinformatics, *Briefings in Bioinformatics*, Volume 7, Issue 1, March 2006. (pp. 86–112). [CrossRef]

46. TASKESEN, E. 2021. A Step-by-Step Guide in detecting causal relationships using Bayesian Structure Learning in Python. Hands-on Tutorials. Available online at: [CrossRef]. Accessed: 2022-03-01

47. VAN ROSSUM, G. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam. 1995.

48. HARRIS, C. R.; MILLMAN, K. J.; van der WALT, S. J. et al. Array programming with NumPy. Nature. 2020, 585, 357–362

49. MCKINNEY, W. et al. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference. 2010, 445, pp. 51–56.

50. PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. 2011. JMLR 12, pp. 2825–2830. [CrossRef]

51. SCHREIBER J. Pomegranate: fast and flexible probabilistic modeling in python. Journal of Machine Learning Research, 18(164), pp. 1–6. 2018.

52. HUNTER, J. 2007. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), pp. 90–95.