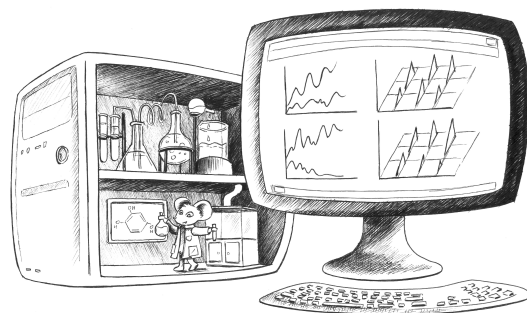


# ALGORITHMS AND COMPUTATIONAL MODELS IN CHEMICAL ANALYSIS

GRZEGORZ SKORACZYŃSKI



Faculty of Mathematics, Informatics, and Mechanics  
University of Warsaw

January 2023

Grzegorz Skoraczyński: *Algorithms and computational models in chemical analysis*, © January 2023

**SUPERVISOR:**  
Błażej Miasojedow

**LOCATION:**  
Warsaw

To Anna, Michał, and Gamma

— GS





## DECLARATIONS

---

I hereby declare that this dissertation is my own work.

*Warsaw, January 2023*

---

Grzegorz Skoraczyński

I hereby declare that this dissertation is ready to be reviewed.

*Warsaw, January 2023*

---

Błażej Miasojedow



## ABSTRACT

---

In the present work, we undertake two problems of computational chemistry: retention time alignment and synthetic accessibility scoring. For the former one, we present the Alignstein, an algorithm for LC-MS retention time alignment by feature matching. We show that the algorithm can find the correspondence appropriately even for signals of swapped elution order. We achieve this by taking advantage of the generalization of the Wasserstein distance as mass spectra and feature dissimilarity measure. It allows us to incorporate all signal information and compare features not only by monoisotopic mass but also by their spatial properties or signal distribution. We validate the algorithm on publicly available benchmark datasets obtaining competitive results. Finally, we show that it can detect the information contained in the tandem mass spectrum by the spatial properties of LC-MS chromatograms.

For the latter problem, we design three different synthetic accessibility scores. The first one is based on a manually prepared set of descriptors, computed on molecules from the database. This model uses stochastic gradient descent to model the distribution of descriptors and predict the likelihood of molecule structure. The second model is based on the same set of descriptors but applies supervised learning to predict compound synthetic accessibility. It requires creating a part dataset representing infeasible molecules, for which we use the bootstrap method. The last model is based on semisupervised learning for outliers detection: One Class SVM. It does not require creating part of the dataset corresponding to non-existent molecules. Moreover, we trained it on extended-connectivity fingerprints, which allows for capturing all possible structural patterns. In this work, we discuss their applicability as a preretrosynthesis heuristic, their limitations, as well as verify the correctness of their predictions. One of the challenges of designing new synthetic accessibility scores is their verification with a ground-truth dataset. To this point, we assess if synthetic accessibility scores: SAScore, SCScore, RAScore, SYBA, and previously described OCSVM-based score can reliably predict outcomes and complexity of the retrosynthesis planning performed by the AiZynthFinder tool. Moreover, by in-depth analysis of AiZynthFinder search trees, we assess if synthetic accessibility scores can speed up retrosynthesis planning by better prioritizing partial synthetic routes.

## STRESZCZENIE

---

W niniejszej pracy podejmujemy dwa problemy chemii obliczeniowej: problem uliniowienia czasu retencji w chromatografii cieczowej oraz problem przewidywania synteżowalności cząsteczek. W przypadku pierwszego z nich przedstawiamy Alignsteina, algorytm do uliniowienia czasu retencji metodą mapowania cech. Pokazujemy, że algorytm ten może poprawnie znaleźć odpowiedniość sygnałów nawet o zamienionej kolejności elucji. Aby to osiągnąć, korzystamy z uogólnienia dystansu Wassersteina jako miary podobieństwa widm masowych. Pozwala nam uwzględnić wszystkie informacje o cechach i porównywać je nie tylko na podstawie różnicy masy monoizotopowej, ale także ich właściwości przestrzennych, czy rozkładu sygnału. Weryfikujemy algorytm na publicznie dostępnych zestawach danych porównawczych, uzyskując konkurencyjne wyniki. Na koniec pokazujemy, że może wykryć informacje zawarte w tandemowym widmie masowym za pomocą przestrzennych właściwości chromatogramów.

Dla drugiego problemu projektujemy trzy różne modele oceny synteżowalności cząsteczek. Pierwszy oparty jest na ręcznie przygotowanym zestawie deskryptorów cząsteczek. Model ten wykorzystuje metodę stochastycznego spadku wzdłuż gradientu do modelowania rozkładu deskryptorów i przewidywania prawdopodobieństwa struktury cząsteczki. Drugi model opiera się na tym samym zestawie deskryptorów, ale wykorzystuje uczenie nadzorowane do synteżowalności związków chemicznych. Wymaga on, aby zbiór treningowy zawierał elementy reprezentujące nieistniejące cząsteczki. Tworzymy je stosując metodę bootstrap. Ostatni model oparty jest na uczeniu częściowo nadzorowanym stworzonym celu do wykrywania anomalii w zbiorach treningowych: jednoklasowego SVM. Nie wymaga on tworzenia części zbioru treningowego odpowiadającej nieistniejącym cząsteczkom. Co więcej, wytrenowaliśmy go na ECFP, numerycznej reprezentacji cząsteczek, która pozwala na zakodowanie obecności wszystkich możliwych wzorców strukturalnych. W tej pracy omawiamy poprawność predykcji modeli do przewidywania synteżowalności, a także ich ograniczenia. Jednym z wyzwań związanych z projektowaniem nowych modeli do oceny synteżowalności cząsteczek jest ich weryfikacja na dobrze opisanym zbiorze danych. W tym celu analizujemy, czy modele do oceny synteżowalności: SAscore, SCScore, RAscore, SYBA a także wcześniej opisany model oparty na jednoklasowym SVM mogą wiarygodnie przewidywać wyniki i złożoność planowania retrosyntetycznego. Ponadto dogłębnie analizujemy drzewa przeszukiwania narzędzia AiZynthFinder i oceniamy, czy modele do oceny synteżowalności mogą przyspieszyć planowanie retrosyntetyczne poprzez lepsze priorytetyzowanie częściowych wyników.

#### KEYWORDS

retention time alignment, Wasserstein distance, synthetic accessibility

#### 2012 ACM COMPUTING CLASSIFICATION

Applied computing → Physical sciences and engineering → Chemistry  
Applied computing → Life and medical sciences → Bioinformatics

#### SŁOWA KLUCZOWE

uliniowanie czasu retencji, odległość Wassersteina, synteżowalność

#### TYTUŁ ROZPRAWY W JĘZYKU POLSKIM

Algorytmy i modele obliczeniowe w analizie chemicznej



## ACKNOWLEDGMENTS

---

I would to thank all people who helped me to perform all research required for this thesis. Especially, the PI of my research group prof. Anna Gambin, and my supervisor prof. Błażej Miasojedow. Moreover, I want to thank Anna Skoraczyńska for her endless patience and Michał Skoraczyński for motivating me to eventually finish this work. Finally, I want to thank Gamma for every night watching me during writing this thesis.

Work summarized in this thesis was supported by several Polish National Science grants: Preludium 2019/33/N/ST6/02949, Opus 2018/29/B/ST6/00681, and Opus 2018/31/B/ST1/00253.





# CONTENTS

---

## I COMPUTATIONAL PROBLEMS OF CHEMICAL SCIENCE

- 1 INTRODUCTION 3
  - 1.1 The problem of the retention time alignment 4
  - 1.2 The problem of synthetic accessibility scoring 6
  - 1.3 Author's contribution 8

## II LC-MS RETENTION TIME ALIGNMENT

- 2 SCORING MASS SPECTRA SIMILARITY 13
  - 2.1 The Wasserstein distance 14
  - 2.2 Wasserstein distance application 16
  - 2.3 Generalized Wasserstein distance derivation 17
    - 2.3.1 Entropic penalizing 17
    - 2.3.2 Dealing with noise 22
- 3 THE ALIGNSTEIN 25
  - 3.1 Algorithm formulation 25
  - 3.2 Implementation details 31
- 4 ALIGNSTEIN VERIFICATION 33
  - 4.1 Algorithm benchmarking 33
  - 4.2 Detecting repeating biomarkers 36
    - 4.2.1 Data collection and curation 37
    - 4.2.2 Data analysis 37
  - 4.3 Swaps experiment 39
  - 4.4 Comments on Alignstein's results 40

## III PREDICTION OF SYNTHETIC ACCESSIBILITY

- 5 APPROACHES TO SYNTHETIC ACCESSIBILITY SCORING 45
  - 5.1 Approach by modeling motifs co-occurrences 45
    - 5.1.1 Dataset 45
    - 5.1.2 The model 46
    - 5.1.3 Comments on model accuracy 52
  - 5.2 Approach by supervised learning 52
    - 5.2.1 Descriptors and dataset 53
    - 5.2.2 The model training 55
    - 5.2.3 Model validation 56
    - 5.2.4 Comparison with existing solutions 57
    - 5.2.5 gradient boosting machines (GBM) model predictive efficiency 59
    - 5.2.6 Comments on model accuracy 59
  - 5.3 Approach by semisupervised learning 60
    - 5.3.1 The model and training 60
    - 5.3.2 Model training and verification 63

6	ASSESSMENT OF SYNTHETIC ACCESSIBILITY SCORES IN COMPUTER-ASSISTED SYNTHESIS PLANNING	67
6.1	Analyzed synthetic accessibility scores	67
6.2	AiZynthFinder, the analyzed CASP tool	68
6.3	Evaluation of synthesis planning and scores	69
6.3.1	Dataset	69
6.3.2	Analysis of the search trees	70
6.4	Results and Discussion	71
	BIBLIOGRAPHY	79

## LIST OF FIGURES

---

Figure 1.1	The optimal transport plan between two features. 6
Figure 2.1	Optimal transport between two simple spectra. 16
Figure 2.2	An example of a deconvolution of a simulated human hemoglobin spectrum. 17
Figure 2.3	Impact of $\varepsilon$ parameter on the objective function for the two variable minimization problem. 19
Figure 2.4	Optimal transport plans for various values of parameters $\varepsilon$ and $\lambda$ . 20
Figure 3.1	Example of swapped features representing two peptides. 26
Figure 3.2	The outline of the Alignstein algorithm. 26
Figure 3.3	Flow network for finding the optimal feature matching between features of one chromatogram and features of the rest of chromatograms. 30
Figure 3.4	Flow network for finding the optimal feature matching between two chromatograms. 31
Figure 4.1	Identification recalls calculated for sample replicates. 38
Figure 4.2	Identification recall calculated for all BaP concentrations. 39
Figure 4.3	Histogram of average RT differences between feature pairs annotated with the same identification. 41
Figure 5.1	Motifs distribution in a database. 47
Figure 5.2	Interactions among the subset of motifs. 48
Figure 5.3	Workflow of stochastic gradient descent algorithm for detection of motif interactions. 50
Figure 5.4	Heatmap of $\theta$ coefficients describing motifs interactions 51
Figure 5.5	Distribution of scores over the test set for several $\ell_1$ penalty values of $\lambda$ 52
Figure 5.6	The workflow of the model training and prediction. 54
Figure 5.7	Boxplots and ROC curve for discrimination of real molecules from decoys. 57
Figure 5.8	Comparison of different scores for the set of 40 molecules including Mf-Score 58

- Figure 5.9 Comparison of score distributions for three substance types. 59
- Figure 5.10 Descriptors feature importance. 60
- Figure 5.11 A hyperplane separating blue data points from orange ones. 62
- Figure 5.12 Example of separation two sets of points with RBF kernel. 63
- Figure 5.13 Data points and decision boundaries of OCSVM model on a training dataset. 64
- Figure 5.14 Comparison of different scores for the set of 40 molecules including OC-MF-Score. 65
- Figure 6.1 Types of AiZynthFinder search tree nodes. 69
- Figure 6.2 Configurations of analyzed nodes. 70
- Figure 6.3 ROC curve for synthetic accessibility scores prediction of AiZynthFinder outcomes. 72
- Figure 6.4 Heatmap of correlation between synthetic accessibility scores and complexity search tree parameters. 72
- Figure 6.5 Heatmaps of t-test p-values for hypothesis if synthetic accessibility scores discriminate node types. 73
- Figure 6.6 ROC curves for discrimination of internal and not solved nodes by appropriately scaled synthetic accessibility scores. 74
- Figure 6.7 ROC curves for discrimination of solved and not solved nodes by appropriately scaled synthetic accessibility scores. 75
- Figure 6.8 Heatmap of AUC of discrimination between internal and not solved nodes and solved and not solved nodes. 76

## LIST OF TABLES

---

Table 4.1	Detailed results for P1 set in CAAP comparison. 35
Table 4.2	Detailed results for P2 set in CAAP comparison. 36
Table 4.3	Comparison of alignment precision, alignment recall, and F-score for M1 set in CAAP comparison. 37
Table 5.1	Example motif patterns. 47
Table 5.2	Distances between instances of nonzero motif groups. 55
Table 6.1	Comparison of analyzed synthetic accessibility scores in predicting the AiZynthFinder outcomes. 71

## ACRONYMS

---

AUC	area under the curve
BaP	benzo[a]pyrene
CAAP	Critical Assessment of Alignment Procedures
CADD	computer-assisted drug discovery
CASP	computer-assisted synthesis planning
DL	deep learning
ECFP <sub>4</sub>	extended-connectivity fingerprints of diameter 4
GBM	gradient boosting machines
GWD	generalization of the Wasserstein distance
HPLC	high-performance chromatography
IR	identification recall
LC	liquid chromatography
LC-MS	liquid chromatography-mass spectrometry
LC-MS/MS	liquid chromatography with tandem mass spectrometry
LFQ	label-free quantification
LP	linear programming
MCTS	Monte Carlo tree search
ML	machine learning
MLE	maximum likelihood estimator
MS	mass spectrometry
M/Z	mass-to-charge ratio
OCSVM	One-class Support Vector Machines
OT	optimal transport
PCA	principal component analysis
RBF	radial basis function
ROC	receiver operating characteristic
RT	retention time
SVM	Support Vector Machines
UCB	upper confidence bound
UPLC	ultra-performance chromatography
VS	virtual screening

Part I

COMPUTATIONAL PROBLEMS OF  
CHEMICAL SCIENCE





## INTRODUCTION

---

With the development of high-level programming languages, such as Fortran, and the field of scientific computing, computational chemistry spread in the 1960s. The term chemometrics emerged for the first time in 1971 in Svante Wold's grant proposal and one year later in his article (in Swedish) [1]. Shortly after, in 1974 Kowalski joined him and created an International Chemometrics Society [2–4]. In the beginning, the main areas of chemometrics research were: processing datasets obtained from instrumental analysis [5–7], computer-assisted synthesis planning (CASP) [8, 9], automatic molecular structure prediction [2, 10]. Parallely to chemometrics development, researchers were interested in automated processing of chemical information [11, 12], such as: 2D and 3D quantitative structure-activity relationship [13], chemical structure encoding [14–16], database deposition and searching [17], molecular fingerprints and molecular similarity detection [18–20]. With the development of public and commercial databases (e.g. Cambridge Structural Database [21] or Chemical Abstracts Service [22]) and an increasing amount of processed data, cheminformatics emerged in the 1980s [2]. In the 1990s, it became recognized as a distinct discipline [23]. The term cheminformatics was used for the first time in 1998 by Brown in his work [24]. Recently, with continuously increasing computational power and the development of machine learning (ML) techniques, cheminformatics is also getting focused on other problems such as computer-assisted drug discovery (CADD) or molecular dynamics simulations. This resulted in a spread of various computational models, for example, molecular docking or molecular dynamics, which allow for the design of new, unknown molecular structures [25–28].

In the present work, we undertake two computational problems of chemistry: one originating from automatic analysis mass spectrometry (MS) results and the other one originating from CASP. The former is the problem of retention time (RT) alignment for liquid chromatography-mass spectrometry (LC-MS) experiments. MS is an analytical instrumental method that allows for precise measuring a mass-to-charge ratio (M/Z) of ions. A set of masses and measured mass intensities are stored in a mass spectrum plot. Complex mixtures require, however, prior separation usually done by liquid chromatography (LC). It is less reproducible than MS due to the appearance of RT drift, which needs to be corrected during the data analysis. This correction is named RT alignment (see Section 1.1 for details). Here, we present an Alignstein, a retention time alignment algorithm which

is based on the Wasserstein distance a spectra dissimilarity measure as described in Chapters 2-4.

The latter problem regards **CASP** and computational retrosynthesis. It is a technique for finding synthetic routes of target molecules from simple, available precursors. Although the existence of many retrosynthetic algorithms, still their computation time is too high for screening methods. One of the solutions is computing synthetic accessibility scores which allow for predicting if a given compound is synthesizable (see Section 1.2 for details). Here, we design three different accessibility scores (Chapter 5) and discuss their applicability as a preretrosynthesis heuristic. Moreover, we assess the accuracy of **CASP** tools and compare the available synthetic accessibility scores (Chapter 6).

### 1.1 THE PROBLEM OF THE RETENTION TIME ALIGNMENT

Advances in **LC-MS** have provided a remarkable insight into the functioning of the organisms, ranging from protein level [29], through tissue [30] to environmental networks [31]. All of these research studies benefit from the possibility of separating complex mixtures in the liquid chromatographic column and then measuring the analytes with a high throughput mass spectrometer. Although **LC-MS** systems provide precise answers to both quantitative and qualitative biological and medical questions, designing algorithms for efficient and precise analysis of **LC-MS** datasets remains challenging.

One of these challenges is the correction of errors caused by retention time **RT** drift. It limits the reproducibility of **LC** separation, which is important for experiments usually acquired in many (even hundreds) replicates. **RT** drift became a significant obstacle with the emergence of high-performance chromatography (**HPLC**) and ultra-performance chromatography (**UPLC**) technologies. For example, nanoflow **UPLC** column separation takes a relatively long time, usually up to several hours. For these experiments, the elution time of peptides may vary up to 5 minutes [32] or even 10 minutes [29].

**RT** drift can be corrected by the experimental protocol only to a limited extent [33]. It may change the whole gradient or affect only single peaks. These changes may be caused by various reasons such as the unstable mobile phase, the column change or degradation, sample chemical instability, or imprecise experiment setup [34–36].

**RT** drift requires a correction, usually named the **RT** alignment. It results in the correspondence of signals across runs [37]. For example, in proteomics, the signal correspondence of the same peptides is needed for further applying label-free quantification (**LFQ**) for which samples must be measured separately [38, 39]. Moreover, for **LFQ** techniques, we cannot obtain the correspondence any other way because analytes

do not have any additional information, such as metabolic labels, or chemical tags [40, 41].

RT drift may swap the order of eluting analytes. For example, in a dataset of Marine Mussels' intestinal protein, we analyzed that about 3 % of all identified feature pairs are swapped between two chromatograms (cf. Section 4.3). Although many of the available algorithms properly align most signals, still they fail to resolve swaps.

The vast majority of approaches to RT alignment are so-called warping algorithms, e.g. OpenMS [42], MetAlign [43], MZMine 2 [44], SIMA [45], the solution proposed by Zhang [46], MS-Dial [47], DI-AlignR [48], the solution proposed by Chiung-Ting Wu, et al. [49]. These algorithms consist of applying a warping function that transforms the chromatograms by shifting, stretching, and squeezing. These transformations result in a close distance between corresponding signals. After alignment, however, further feature detection and matching are still required to obtain the signal correspondence. These algorithms' applicability is limited because the warping function is applied under the assumption that ions elute monotonically with RT. Thus, they are not able to deal with elution order swaps.

Alternatively, a rarer implemented approach is feature matching, e.g. OpenMS [42] (both warping and matching algorithm), MassUntangler [50], LWBMatch [51], the solution proposed by Wandy, et al. [52], Quandenser [53]. Algorithms by feature matching find the correspondence between initially detected features of two or more chromatograms. Features are convex sets of peaks representing the signal of a single analyte. Corresponding features represent the same analyte and further will be referred to as consensus features. To the best of the authors' knowledge, all matching algorithms reduce multi-dimensional features to one-dimensional extracted ion chromatogram or a single point with monoisotopic peak  $M/Z$  and average RT value, ignoring the information of isotopic envelope or feature span over the RT dimension. Without feature spatial characteristics and information of coeluting ions, elution order swaps are practically undetectable [36]. The main reason for this simplification lies in the difficulty to find multidimensional feature dissimilarity measures. Typically, Euclidean distance between points or one-dimensional cosine-like spectra similarity scores is applied [54, 55].

In this work, we present a feature matching RT alignment algorithm named Alignstein. It overcomes the limitations of current algorithms and properly resolves the correspondence of analytes of swapped elution order. To achieve this, we take advantage of the generalization of the Wasserstein distance (GWD) [56] to compare multidimensional features as described in Chapter 2. It originates from the optimal transport (OT) theory and has been recently attracting growing attention to various problems of mass spectrometry [54, 57–60]. In brief, the Wasserstein distance describes the cost of the optimal way how to transform

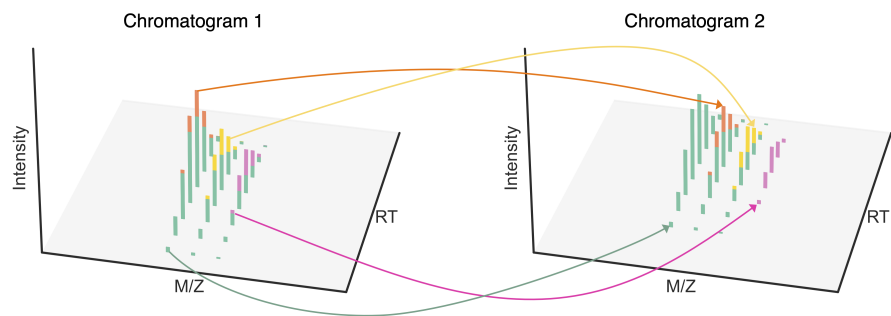


Figure 1.1: The optimal transport plan between two features. The Wasserstein distance captures not only the distance of feature drift along the [RT](#) dimension but also spatial differences between features. Here, the left feature consists of three ions, right feature consists of four ions. To properly capture this difference, part of the signal must be transported between different ions (denoted with arrows) and thus the transport cost (the Wasserstein distance) is higher.

one feature into the other one. The transformations include not only shifting the signal from one feature to another but also splitting or combining the signal between peaks (cf. Figure 1.1). To obtain the most feasible alignment results, Alignstein has formulated a complex optimization signal-matching problem, for which we use clustering and network flow algorithms to achieve a computationally tractable outcome as described in Chapter 3. We evaluated Alignstein on several different datasets and achieved competitive results as described in Chapter 4.

## 1.2 THE PROBLEM OF SYNTHETIC ACCESSIBILITY SCORING

[CASP](#) consists of two tasks: reactions forward planning and retrosynthesis. The former is predicting the outcomes of reaction for given reactants. The latter is a method of planning the synthesis scheme of chemical compounds from simple precursors available in stock, to synthesized intermediates, and the target molecule. Synthesis planning remained a laborious, manual task until the 1960s when Corey [61] formalized the idea of [CASP](#) and then implemented it in LHASA [9] software. Over the years, new solutions were developed that automated subsequent planning elements, required less human intervention, and increased the speed and accuracy of algorithms [62–64]. Over the last decade, several modern, [ML](#)-based [CASP](#) tools were independently developed: from closed vendor software, e.g. Synthia (previously Chematica) [65, 66], to the closed source with the available interface, e.g. IBM RXN [67], and open-source ones, e.g. LillyMol [68], AiZynthFinder [69–71], ASKCOS Tree-builder [72], AutoSynRoute [73]. Currently, a standard [CASP](#) tool [74] consists of three modules: (i) the database of reaction templates and rules on how to apply them to analyzed molecules, (ii) algorithms searching for possible synthetic routes,

(iii) a database of in-stock molecules. The aforementioned tools differ significantly in the design of every module. For example, the database of reaction templates may be manually encoded with a rule-based algorithm for reaction prediction, e.g. Synthia. It may be also automatically extracted and reactions may be predicted with a neural network, e.g. LillyMol, AiZynthFinder, and ASKCOS Tree-builder. Finally, reactions may be predicted using a template-free seq2seq algorithm [75] known from natural language processing as implemented IBM RXN.

Besides CASP tools' strengths, their key bottleneck is computational complexity. During retrosynthesis planning runtime, potentially exponential in size search space of solution candidates (partial synthetic routes) must be traversed. It makes CASP tools non-applicable when numerous molecules need to be immediately checked for synthesizability. One example is a virtual screening (VS) method known in CADD. During VS, even billions of compound candidates are evaluated for desired properties; thus, searching for a synthetic route for each of these candidates is computationally intractable.

This limitation may be overcome by scoring the synthetic accessibility, i.e. by predicting how the molecule of a given structure is synthesizable. Previously, synthetic accessibility scores were based on single molecular properties selected manually by experts [76–79]. With the emergence of machine learning and deep learning (DL) methods, new scores were designed. They can be divided into structure-based and reaction-based approaches. Structure-based approaches evaluate the feasibility of molecular structure, e.g. SAScore [80], SYBA [81], GASA [82]. Reaction-based approaches predict the synthetic accessibility by capturing the similarity of synthetic routes deposited in reaction databases, e.g. SCScore [83], RAScore [84], CMPNN [85].

Here, we pose the question if synthetic accessibility scores can speed up retrosynthesis planning by better prioritizing partial synthetic routes. For this reason, we propose three different structure-based models for scoring synthetic accessibility as described in Chapter 5. The first one is based on a carefully curated set of descriptors, i.e. structural patterns which encode compounds' molecular properties. We discuss whether special descriptors with an appropriately trained model can overcome the limitations of existing synthetic accessibility scores. Here, we use stochastic gradient descent to model the distribution of descriptors in existing molecules and predict molecule likelihood. The second score is based on the same set of descriptors but applies a supervised learning model to predict compound synthetic accessibility. A challenging part of its design is part of the training set representing infeasible molecules. It is created using bootstrap methods, i.e. by randomizing fragments of real molecules' descriptors. The last model is based on semisupervised learning for outliers detection: One-class Support Vector Machines (OCSVM). This approach allows for

omitting to create a negative set and the model predicts if the new test molecule is similar to any molecule from the training set or not.

One of the challenges of designing new synthetic accessibility scores is their verification with a ground-truth dataset. There is a lack of critical assessment of synthetic accessibility scores on the standardized dataset with common test conditions. Moreover, the majority of aforementioned scores are publicly available and documented, but their applicability as a pre-retrosynthesis heuristic is known to a limited extent. To this end, we assess if synthetic accessibility scores can reliably predict outcomes of retrosynthesis planning as described in Chapter 6. We also analyze if synthetic accessibility scores can speed up the retrosynthesis planning by reducing the size of the search space. Specifically, we analyze the outcomes and runtime of the retrosynthetic tool AiZynthFinder on a specially prepared compounds database. We assess if four scores: SAscore, SCScore, RAscore, and SYBA properly predict the results of retrosynthesis planning and the search complexity and compare them with the synthetic accessibility score based on the OCSVM model (see Section 5.3). To do this, we analyze the AiZynthFinder partial solutions search trees. Moreover, by in-depth analysis of these search trees, we assess if synthetic accessibility scores can speed up retrosynthesis planning by better prioritizing partial synthetic routes.

To the best of the authors' knowledge, it is the first of this kind of assessment. Although benchmarks are available in cheminformatics, they focus on the outputs of the CASP tools [86] or synthetic accessibility scores alone [87, 88].

### 1.3 AUTHOR'S CONTRIBUTION

This thesis describes the author's scientific research achieved during Ph. D. studies. Results presented in Part II were published in the article:

Grzegorz Skoraczyński, Anna Gambin, and Błażej Miasojedow. "Alignstein: Optimal Transport for Improved LC-MS Retention Time Alignment." In: *GigaScience* 11 (Nov. 2022), giac101.

The author as the first author of this work, co-worked on the algorithm design, as well as implemented and verified its accuracy. Section 2.2 partially summarizes the article:

Michał Aleksander Ciach, Błażej Miasojedow, Grzegorz Skoraczyński, Szymon Majewski, Michał Startek, Dirk Valkenburg, and Anna Gambin. "Masserstein: Linear Regression of Mass Spectra by Optimal Transport." In: *Rapid Communications in Mass Spectrometry* (Sept. 2020), e8956.

For this article, the author implemented the analysis of human hemoglobin spectra deconvolution, as well as co-worked on algorithm implemen-

tation. Finally, the content of Chapter 6 comes from the article (a preprint):

Grzegorz Skoraczyński, Mateusz Kitlas, Błażej Miasojedow, and Anna Gambin. "Critical Assessment of Synthetic Accessibility Scores in Computer-Assisted Synthesis Planning." In: *Chemrxiv* (Nov. 2022), which was accepted for publication in *Journal of Cheminformatics* recently. The author led this project, and being one of the first authors, performed statistical data analysis, and revised a tool for analysis of synthetic accessibility scores and AiZynthFinder search trees.





## Part II

### LC-MS RETENTION TIME ALIGNMENT



## SCORING MASS SPECTRA SIMILARITY

One of the key bottlenecks of currently developed [RT](#) alignment algorithms is the inability to effectively compare signal sets, e.g. spectra. Typically, cosine-like similarity scores are used [[91](#), [92](#)]. These methods treat mass spectra as vectors in one space and compare by measuring the angle between them using a cosine or a similar statistic (dot-product, correlation [[55](#)]). For this purpose, intensity vectors require to have the same length. It is achieved by prior peak matching between two spectra so that peaks of small [M/Z](#) difference (e.g. [M/Z](#) difference smaller than 0.5 Da [[93](#)]) are treated as corresponding. Usually, it is obtained by binning intensities in [M/Z](#) ranges of constant size or by gathering the signal around the most intense peaks [[93](#), [94](#)]. If  $a_1, \dots, a_n$  and  $b_1, \dots, b_n$  are matched intensities of spectra then the similarity between them is equal to the cosine between them:

$$\frac{\sum_{i=1}^n a_i \cdot b_i}{\sqrt{\sum_{i=1}^n a_i^2} \cdot \sqrt{\sum_{i=1}^n b_i^2}}.$$

These similarity scores have, however, several limitations:

- they rely on the arbitrary peak matching or [M/Z](#) domain unification and thus are limited when comparing spectra of different resolutions;
- they are non-scalable with dimension, e.g. not capable of comparing [LC-MS](#) features;
- they cannot reliably compare similar but highly distanced spectra, e.g. peptide [MS1](#) spectra with a single peptide modification or post-translational modification, metabolomic spectra with different [TMS](#) derivatization;
- they score precisely only highly similar spectra but for nonsimilar spectra their precision drastically falls [[95](#)].

Although these limitations are known and well described, the majority of similarity score improvements focus on details of computed statistics leaving the aforementioned problems unsolved [[55](#), [95](#)]. Recently, Huber et al. developed [Spec2Vec](#) [[95](#)], one of the few conceptually different approaches for spectra similarity. They apply techniques known from [ML](#) and natural language processing. Here, we propose a different approach to measuring spectra dissimilarity, which originates from the [OT](#) theory – the Wasserstein distance [[96](#)] with further generalizations [[56](#), [97](#)]. Its design significantly differs from currently existing

similarity scores and thus it overcomes the majority of their limitations. The Wasserstein distance is also known in computer science literature as an Earth Mover’s Distance [98] because it can be interpreted as a way of transforming one pile of sand into the other with the least used amount of work. We introduce its definition in Section 2.1 and in Section 2.2, we show its applicability to mass spectrometry by the presentation of masserstein package. Then in Section 2.3, we generalize the Wasserstein distance so that it allows us to compute it more effectively and better deal with noise. Finally, we describe the most successful application for algorithms in mass spectrometry, Alignstein, the algorithm for LC-MS retention time alignment. For this algorithm, we applied a Wasserstein distance to compare multidimensional sets of signals and it emerged to be a key to resolving signal correspondence of swapped order.

## 2.1 THE WASSERSTEIN DISTANCE

Recently, the Wasserstein distance has emerged as a practical metric for comparing probability measures. It transfers also naturally to mass spectrometry. The mass spectrum represents the distribution of charged ions in a spectrometer detector, and thus it corresponds to some probability measure. Here, we introduce a formal definition of the Wasserstein distance and provide a handy notation for generalizing the Wasserstein distance in Section 2.3.

Suppose, that we have a spectrum represented by a measure  $\mu$ . We normalize the spectrum by its total ion current without substantial loss of information and thus we assume that  $\mu$  is a probability measure. We evaluate centroided spectra, i.e. spectra in which continuous signal was discretized by finding local signal maxima via peak-picking. Thus, we also assume that the measure  $\mu$  has discrete and finite support.

Further in the text, we denote spectrum and measure  $\mu$  interchangeably. For given measure  $\mu$ , we denote its support as  $\text{Supp}(\mu) = (x_1, \dots, x_n)^T$ , which corresponds to  $M/Z$  values and measure masses  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$  corresponding to intensities located in  $\text{Supp}(\mu)$ .

Suppose we have two measures  $\mu$  and  $\nu$  with supports  $\text{Supp}(\mu) = (x_1, \dots, x_n)^T$  and  $\text{Supp}(\nu) = (y_1, \dots, y_m)^T$  and weights  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$  and  $\boldsymbol{\nu} = (\nu_1, \dots, \nu_m)^T$ . For the Wasserstein distance formulation, we define the cost matrix  $\mathbf{M} \in \mathbb{R}_{\geq 0}^{n,m}$ , so that:

$$M_{i,j} = |x_i - y_j|.$$

Cost matrix  $\mathbf{M}$  describes the cost of transportation of the unit of ion current from peak located in  $M/Z$   $x_i$  of spectrum  $\mu$  to peak located in  $M/Z$   $y_j$  of spectrum  $\nu$ . Here, we use the  $\ell_1$  metric as a cost.

We define also set  $\mathcal{U}$  as a set of all couplings between measures  $\mu$  and  $\nu$  as:

$$\mathcal{U}(\mu, \nu) = \left\{ \mathbf{T} \in \mathbb{R}_{\geq 0}^{n,m} \mid \mathbf{T} \cdot \mathbf{1}_m = \boldsymbol{\mu}, \mathbf{T}^T \cdot \mathbf{1}_n = \boldsymbol{\nu} \right\}.$$

We name the coupling  $\mathbf{T} \in \mathbb{R}_{\geq 0}^{n,m}$  as a transport plan, where  $T_{i,j}$  describes amount of ion current transported from  $x_i$  of spectrum  $\mu$  to  $y_j$  of spectrum  $\nu$ . For given transport plan  $\mathbf{T}$  and cost matrix  $\mathbf{M}$ , we define a transport cost as a sum of multiplied transported amounts of ion current and costs, i.e.:

$$\sum_{i,j} T_{ij} M_{ij}$$

The Wasserstein distance is the cost of the optimal transport plan, i.e. cost of the solution to the OT problem:

$$\text{OT}_{\mathbf{M}}(\mu, \nu) = \min_{\mathbf{T} \in \mathcal{U}(\mu, \nu)} \sum_{i,j} T_{ij} M_{ij}. \quad (2.1)$$

If  $\mathbf{T}^*$  is the optimal transport plan, then the Wasserstein distance is:

$$d^W(\mu, \nu) = \sum_{i,j} T_{ij}^* M_{ij}.$$

The problem of finding optimal  $\mathbf{T}^*$  is an linear programming (LP) problem, but for one-dimensional spectra, it can be computed in linear time [99].

*Example.* Suppose that we want to compute the Wasserstein distance between two spectra depicted in Figure 2.1. Spectra  $\mu$  and  $\nu$  can be defined so that

$$\begin{aligned} \boldsymbol{\mu} &= \left( \frac{1}{3}, \frac{2}{3} \right)^T & \text{and} & & \boldsymbol{\nu} &= \left( \frac{2}{3}, \frac{1}{3} \right)^T \\ \text{with } \text{Supp}(\mu) &= (1, 2)^T & & & \text{with } \text{Supp}(\nu) &= (1, 3)^T. \end{aligned}$$

Then, the cost matrix is equal to:

$$\mathbf{M} = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix}.$$

and the optimal transport plan, depicted with blue lines, is equal to:

$$\mathbf{T}^* = \begin{bmatrix} \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

The cost of optimal transport (the Wasserstein distance) equals:

$$d^W(\mu, \nu) = \frac{1}{3} \cdot 0 + 0 \cdot 2 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}.$$

◦

The Wasserstein distance as spectra dissimilarity measure overcomes limitations of existing similarity scores. It is not based on arbitrary peak matching. Instead, for optimal transformation, the signal from a single peak may be transported to several peaks of the other spectrum.

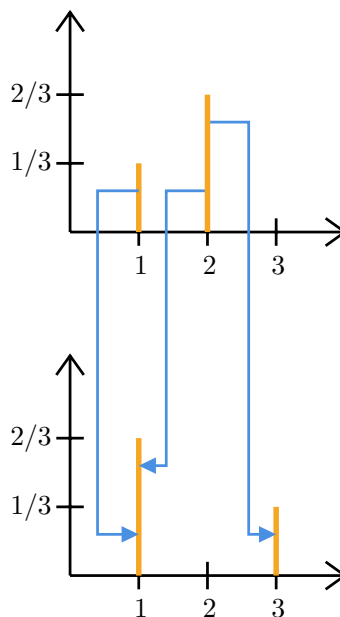


Figure 2.1: Optimal transport between two simple spectra.

Thus, the Wasserstein distance can compare meaningfully even spectra of different resolutions. Moreover, it is capable of transporting signals over larger distances, so it is capable of comparing different spectra giving intermediate values of similarity. Finally, it is easily scalable with dimension so it is capable of comparing multidimensional signal sets as described in Chapter 3.

## 2.2 WASSERSTEIN DISTANCE APPLICATION

Although Wasserstein is attracting more and more research interest in the scope of mass spectrometry, finding its working applications remains challenging. One of the successful results is the Wasserstein algorithm [57, 100]. It is an algorithm for linear regression of spectra, also known as deconvolution. The problem of deconvolution consists of estimating the proportions of identified reference spectra in the experimental spectrum of a mixture. Suppose that we have an experimental spectrum  $\mu$  of the mixture to be deconvolved and  $n$  reference spectra  $\nu_1, \dots, \nu_n$  of substances identified or expected to be present in the analyzed mixture. We want to find a convex combination of spectra that models the experimental spectrum. Define non-negative proportions  $\mathbf{p} = (p_0, p_1, \dots, p_n)^T$  of these substances in a mixture so that  $p_0 + p_1 + \dots + p_n = 1$ .  $p_0$  describes a fraction of signal that is not explained by any reference spectrum (a noise) and  $p_1, \dots, p_n$  describe the proportions of spectra  $\nu_1, \dots, \nu_n$  in the experimental spectrum. Define a model spectrum  $\nu_m$ , which describes the experimental spectrum as a combination of reference spectra:

$$\nu_m = p_1 \cdot \nu_1 + \dots + p_n \cdot \nu_n.$$

We aim to find  $\nu_m$  and  $p_0, \dots, p_n$  that best explain the experimental spectrum  $\mu$ . To properly explain the amount of signal that is not explained by reference spectra we introduce an additional auxiliary spectrum  $\omega$  (named a vortex, an abyss, or trash). We express the best explanation of the experimental spectrum by model spectrum as a Wasserstein distance minimization problem:

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} d^W(\mu, p_0 \cdot \omega + \nu_m).$$

We interpret  $\omega$  so that cost of transporting the signal into this spectrum is constant and equal to  $\kappa$  (cf. Figure 2.2). Such formulated problems can be solved using LP techniques as described in the original article [57]. The algorithm implementation is available publicly at [github.com/mciach/masserstein](https://github.com/mciach/masserstein). This algorithm estimates ion proportions without the requirement for extensive spectra preprocessing and works correctly with both centroided and profile spectra. It was benchmarked on several various datasets, which confirmed its accuracy as detailed in the original works [57, 100].

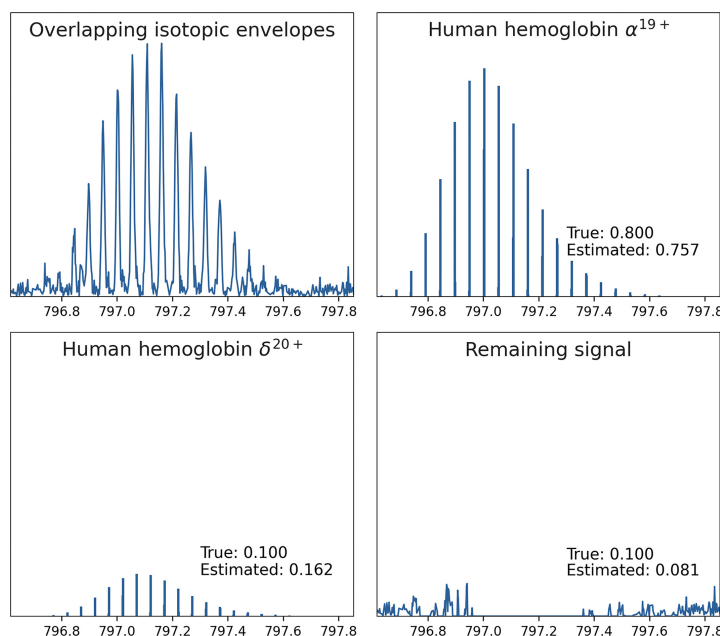


Figure 2.2: An example of a deconvolution of a simulated human hemoglobin ESI MS<sub>1</sub> spectrum for subunits  $\alpha^{19+}$  and  $\delta^{20+}$ . The proportions were estimated directly from the top-left spectrum, figure originates from the masserstein article [57].

## 2.3 GENERALIZED WASSERSTEIN DISTANCE DERIVATION

### 2.3.1 Entropic penalizing

Previously defined OT problem (2.1) is an LP problem and can be solved in polynomial time using the simplex algorithm [101]. We

propose, however, using the Sinkhorn-Knopp scaling approximation algorithm [102] for speeding up computation and better numerical stability. For a broader introduction to Wasserstein distance computation, consult the Peyre and Cuturi's book [56].

We start with reformulating the previous OT problem (2.1) by regularizing it with the entropic penalty:

$$\text{OT}_{\mathbf{M}}^{\varepsilon}(\mu, \nu) = \min_{\mathbf{T} \in \mathcal{U}(\mu, \nu)} \sum_{i,j} T_{ij} M_{ij} - \varepsilon h(\mathbf{T}) \quad (2.2)$$

where  $h(\mathbf{T})$  is entropic penalty term:

$$h(\mathbf{T}) = - \sum_{i,j} T_{ij} \log T_{ij}.$$

Analogously, if  $\mathbf{T}^{\varepsilon*}$  is the transport plan that minimizes  $\text{OT}_{\mathbf{M}}^{\varepsilon}(\mu, \nu)$ , then the cost of optimal transport is:

$$d^{\varepsilon}(\mu, \nu) = \sum_{i,j} T_{ij}^{\varepsilon*} M_{ij}.$$

Adding an entropic penalty makes the objective function strongly convex (cf. Figure 2.3). This results in a unique solution, a better theoretical convergence rate, and thus better numerical stability in comparison with the original LP problem (2.1). Interpreting problem (2.2) geometrically, the entropic regularization term moves the optimal solution of the LP problem from the vertices of the polytope toward its interior (cf. Figure 2.3). Indeed, as shown in [103] and [56] (Proposition 4.1), with  $\varepsilon$  converging to 0, the solution of regularized OT problem (2.2) converges to the solution of non-regularized problem (2.1) of maximum entropy (cf. Figure 2.4):

$$\mathbf{T}^{\varepsilon*} \xrightarrow{\varepsilon \rightarrow 0} \operatorname{argmin} \left\{ -h(\mathbf{T}^*) : \mathbf{T}^* \in \operatorname{argmin}_{\mathbf{T} \in \mathcal{U}(\mu, \nu)} \sum_{i,j} T_{ij} M_{ij} \right\}.$$

On the other hand, with diverging  $\varepsilon$ , the solution of regularized problem (2.2) converges to the coupling of the maximal entropy, i.e. joint probability distribution of random variables distributed by  $\mu$  and  $\nu$  as if they are independent:

$$\mathbf{T}^{\varepsilon*} \xrightarrow{\varepsilon \rightarrow \infty} \mu \nu^{\mathbf{T}}.$$

To compute the solution of problem (2.2), we use Sinkhorn-Knopp algorithm [102]. It is one of the first algorithms for solving the matrix scaling problem, i.e. for a quadratic matrix  $\mathbf{A}$  with positive entries we look for diagonal (scaling) matrices  $\mathbf{X}$  and  $\mathbf{Y}$  so that  $\mathbf{XAY}$  is doubly stochastic [104]. Recently, Cuturi proposed to apply it to solve optimal transport problems [105]. The algorithm scheme consists of iterative alternate scaling rows and columns of the input matrix as depicted in Algorithm 2.1.



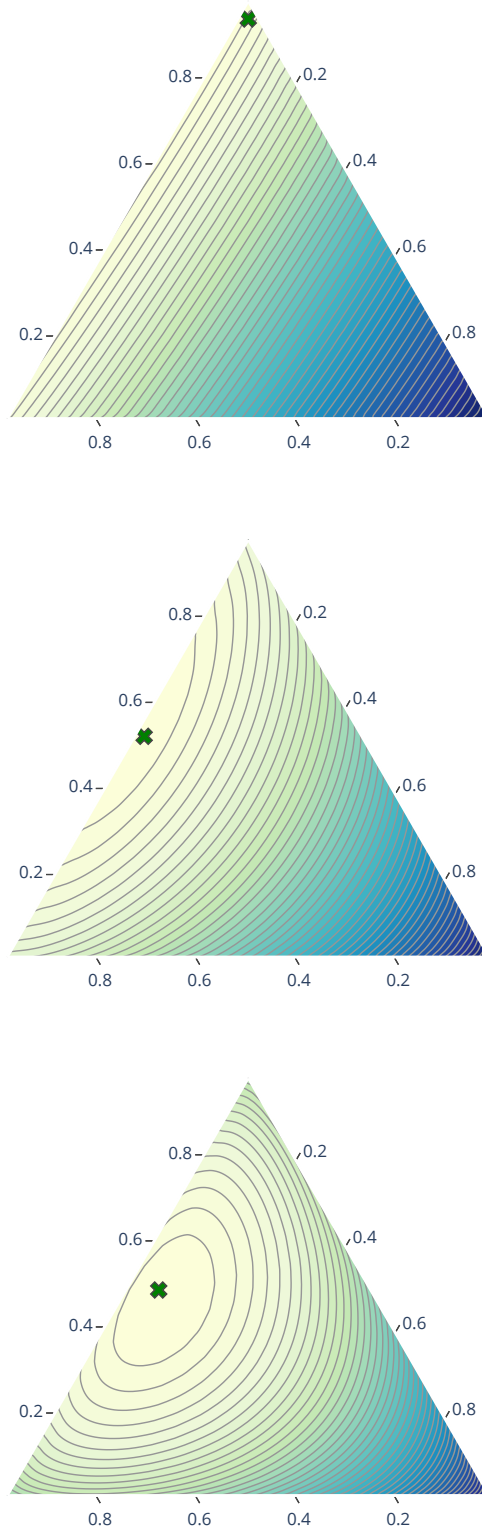


Figure 2.3: Impact of  $\varepsilon$  parameter on the objective function for the two variable minimization problem. For  $\varepsilon = 0$ , the problem reduces to a non-regularized LP problem with the optimum in the vertex of the polytope (first panel). With an increasing value of  $\varepsilon$  (second and third panel) optimum moves towards the interior of the polytope.

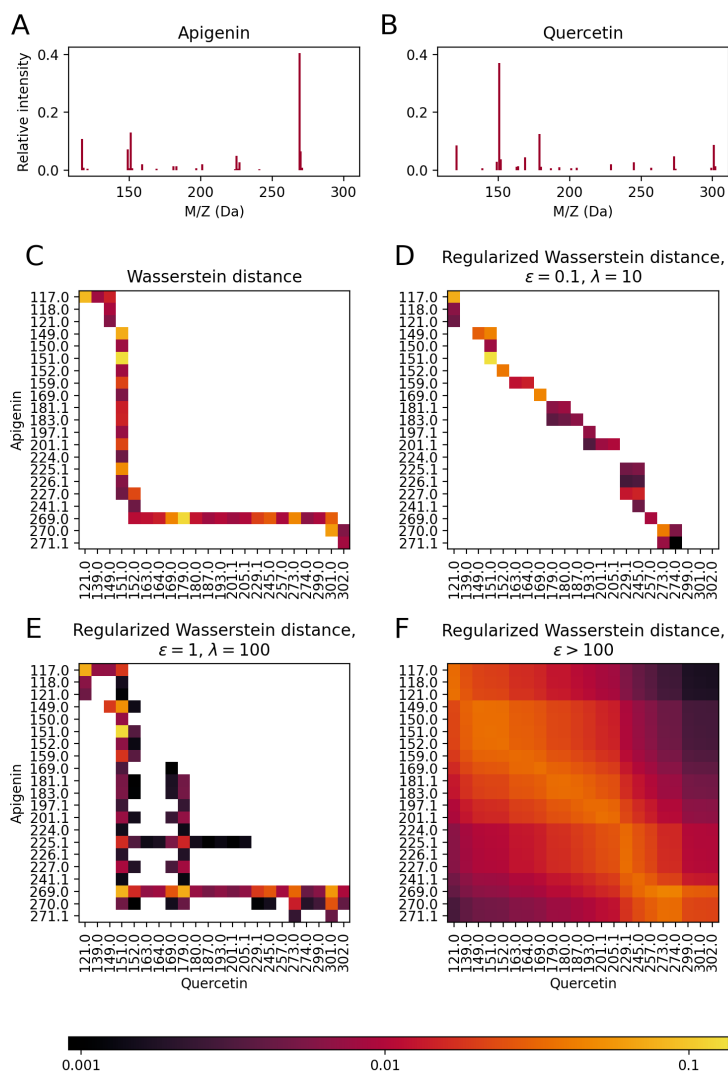


Figure 2.4: Optimal transport plans for various values of parameters  $\epsilon$  and  $\lambda$ . Suppose that we want to compute the distance between the spectra of apigenin (A) and quercetin (B). The non-regularized solution (C), even if finds the value of optimal transport, transports the signal between too-distant peaks. Properly chosen parameters of the regularized solution allow for finding a feasible transport plan (D). For too large  $\lambda$  (E), transport is similar to a non-regularized one. For large values of  $\epsilon$  (F), solution converges to  $\mu\nu^T$

**Algorithm 2.1:** The scheme of the Sinkhorn-Knopp algorithm.

**Data:**  $\mathbf{A}$   
**Result:** doubly-stochastic  $\mathbf{A}$   
**while** *not converged* **do**  
    | scale  $\mathbf{A}$  rows such that rows sum up to 1;  
    | scale  $\mathbf{A}$  columns such that columns sum up to 1;  
**end**  
**return**  $\mathbf{A}$

To apply the Sinkhorn-Knopp algorithm for calculating optimal transport, we start with Sinkhorn's observation [106] that the optimal solution  $\mathbf{T}^{\varepsilon*}$  is unique, and every element has the form:

$$T_{ij}^{\varepsilon*} = u_i v_j e^{-M_{ij}/\varepsilon}$$

for two scaling variables  $\mathbf{u} \in \mathbb{R}^n$  and  $\mathbf{v} \in \mathbb{R}^m$ . We define for  $\mathbf{x} \in \mathbb{R}^n$ , the  $\text{diag}(\mathbf{x})$  as  $n \times n$  matrix with diagonal containing vector  $\mathbf{x}$  and zero otherwise. We notice that we can rewrite transport plan  $\mathbf{T}$  as:

$$\mathbf{T} = \text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v})$$

for a matrix  $\mathbf{K} = e^{-M/\varepsilon}$ . Rows of the transport plan  $\mathbf{T}$  should sum up to  $\boldsymbol{\mu}$  and columns of  $\mathbf{T}$  should sum up to  $\mathbf{v}$ , so we can rewrite it as:

$$\text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v}) \cdot \mathbb{1}_m = \mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \boldsymbol{\mu} \quad (2.3)$$

and:

$$\text{diag}(\mathbf{v})\mathbf{K}^T\text{diag}(\mathbf{u}) \cdot \mathbb{1}_n = \mathbf{v} \odot (\mathbf{K}^T\mathbf{u}) = \mathbf{v}, \quad (2.4)$$

where the  $\odot$  operator is vector indexwise multiplication. The aim of scaling iterations is to alternatively satisfy equations (2.3) and (2.4), i.e for  $(l+1)$ -th iteration, scaling operations get the form:

$$\mathbf{u}^{(l+1)} \leftarrow \frac{\boldsymbol{\mu}}{\mathbf{K}\mathbf{v}^{(l)}} \quad \text{and} \quad \mathbf{v}^{(l+1)} \leftarrow \frac{\mathbf{v}}{\mathbf{K}^T\mathbf{u}^{(l+1)}}$$

where vector division is done indexwise. Finally, the Sinkhorn-Knopp algorithm for regularized OT problem (2.2) has the scheme as depicted in Algorithm 2.2. As shown by Altschuler et al., this algorithm is the  $\tau$ -approximation of the unregularized OT problem (2.1) with  $O(n^2 \log(n)\tau^{-3})$  time complexity [107]. Additionally, Franklin and Lorentz showed linear convergence of Sinkhorn-Knopp scaling iterations [108].

**Algorithm 2.2:** The scheme of the Sinkhorn-Knopp algorithm for computing regularized OT problems.

**Data:**  $M, \varepsilon, \mu, \nu$   
**Result:** optimal  $T^{\varepsilon*}$   
 $K \leftarrow e^{-M/\varepsilon};$   
 $\mathbf{v} \leftarrow \mathbb{1}_m;$   
**while** *not converged* **do**  
   $\mathbf{u} \leftarrow \frac{\mu}{K\mathbf{v}};$   
   $\mathbf{v} \leftarrow \frac{\nu}{K^T\mathbf{u}};$   
**end**  
**return**  $(\mathbf{u}_i \cdot K_{ij} \cdot \mathbf{v}_j)_{i,j}$

### 2.3.2 Dealing with noise

We observed that the Wasserstein distance does not deal well with noise, trying to match it with signal and vice-versa (cf. Figure 2.4). To cope with this problem, we implemented the generalization of previous problems for unbalanced measures as proposed by Chizat et al. [97]. This generalization allows for omit transporting too distant peaks with a penalty proportional to the amount of not transported signal. For this purpose, we rewrite problem (2.2) so that:

$$\text{OT}_M^{\varepsilon, F}(\mu, \nu) = \min_{T \in \mathcal{U}(\mu, \nu)} \sum_{i,j} T_{ij} M_{ij} - \varepsilon h(T) + F(T \cdot \mathbb{1}_m | \mu) + F(T^T \cdot \mathbb{1}_n | \nu). \quad (2.5)$$

$F$  is a divergence that allows us to approximate the optimal solution to  $\mu$  and  $\nu$  measures. There are several divergences  $F$  that can be applied here. For example, the most trivial is identity divergence  $F = \iota$  defined as:

$$\iota(\mathbf{x} | \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{y} \\ +\infty & \text{otherwise.} \end{cases}$$

With  $\iota$  divergence, the  $\text{OT}_M^{\varepsilon, F}$  problem (2.5) reduces to regularized OT problem (2.2). For our application, the total variation divergence achieved the best results:

$$F(\mathbf{x} | \mathbf{y}) = \lambda \text{TV}(\mathbf{x} | \mathbf{y}) = \lambda \|\mathbf{x} - \mathbf{y}\|_{\text{TV}}$$

which for our discrete setup is related to  $\ell_1$  norm as  $\|\mathbf{a}\|_{\text{TV}} = \sum_i |a_i|$ .  $\lambda$  is a parameter of this problem and can be interpreted as a penalty for not transporting a fragment of the signal. Analogously, if  $T^{\varepsilon, F*}$  is transport plan minimizing problem (2.5), the distance is equal to:

$$d^{\varepsilon, F}(\mu, \nu) = \sum_{i,j} T_{ij}^{\varepsilon, F*} M_{ij} + F(T^{\varepsilon, F*} \cdot \mathbb{1}_m | \mu) + F(T^{\varepsilon, F*T} \cdot \mathbb{1}_n | \nu). \quad (2.6)$$

For this problem, the scaling steps of the algorithm need to be extended by using the proximal operator:

$$\text{prox}_{F/\varepsilon}^{\text{KL}}(\mathbf{z}, \mathbf{p}) = \underset{\mathbf{s} \in \mathbb{R}^n}{\text{argmin}} \left\{ F(\mathbf{s}|\mathbf{p}) + \frac{1}{\varepsilon} \text{KL}(\mathbf{s}|\mathbf{z}) \right\},$$

where  $\mathbf{p}$  is an intensity vector  $\boldsymbol{\mu}$  or  $\boldsymbol{\nu}$  respectively and  $\text{KL}(\mathbf{x}|\mathbf{y})$  is Kullback-Leibler divergence:

$$\text{KL}(\mathbf{x}|\mathbf{y}) = \sum_i x_i \log \left( \frac{x_i}{y_i} \right).$$

Operator  $\text{prox}_{F/\varepsilon}^{\text{KL}}$  is the extension of the proximal operator on Euclidean space  $E$ , which is defined for any lower semicontinuous function  $g$  as:

$$\text{prox}_g(x) = \underset{y \in E}{\text{argmin}} \left\{ g(y) + \frac{1}{2} \|x - y\|^2 \right\}.$$

The  $\text{prox}_g$  operator, in turn, is an extension of the projection operator. Indeed, when  $g = \iota_C$  is the convex indicator of set  $C$  then the proximal operator is exactly a projection on set  $C$ . However, when we use the proximal operator on a space of measures, we should replace euclidean distance with another operator of similar properties. One of the natural choices is KL divergence which has similar geometric properties. Scaling steps with proximal operator have the form as below:

$$\mathbf{u} \leftarrow \frac{\text{prox}_{F/\varepsilon}^{\text{KL}}(\mathbf{K}\mathbf{v}, \boldsymbol{\mu})}{\mathbf{K}\mathbf{v}}$$

$$\mathbf{v} \leftarrow \frac{\text{prox}_{F/\varepsilon}^{\text{KL}}(\mathbf{K}^T\mathbf{u}, \boldsymbol{\nu})}{\mathbf{K}^T\mathbf{u}}$$

and finally, the algorithm for finding optimal transport has the form as in Algorithm 2.3. For its efficient computation, Chizat et al. derived the closed-form formula for the proximal operator with total variation divergence  $F = \lambda \text{TV}$ :

$$\text{prox}_{F/\varepsilon}^{\text{KL}}(\mathbf{s}, \mathbf{p}) = \min \left\{ \mathbf{s} \cdot e^{\frac{\lambda}{\varepsilon}}, \max\{\mathbf{s} \cdot e^{-\frac{\lambda}{\varepsilon}}, \mathbf{p}\} \right\}.$$

Python-based implementations of algorithms described in this Chapter are available in the POT package [109]. Moreover, recently, a PyTorch [110] implementation of these algorithms was released [111]. For handy application in mass spectrometry, we implemented these algorithms in the MassSinkhornmetry package available at [github.com/grzsko/MassSinkhornmetry](https://github.com/grzsko/MassSinkhornmetry). This library is purely written in C++ using Eigen library [112] for matrix computations and has available API for both C++ and Python languages.

**Algorithm 2.3:** The scheme of the Sinkhorn-Knopp algorithm for computing regularized OT problems with additional approximations

**Data:**  $\mathbf{M}, \lambda, \varepsilon, \boldsymbol{\mu}, \mathbf{v}$   
**Result:** optimal  $\mathbf{T}$   
 $\mathbf{K} \leftarrow e^{-\lambda \mathbf{M}};$   
 $\mathbf{v} \leftarrow \mathbb{1}_m;$   
**while** *not converged* **do**  
     $\mathbf{u} \leftarrow \frac{\text{prox}_{\mathbb{F}/\varepsilon}^{\text{KL}}(\mathbf{K}\mathbf{v}, \boldsymbol{\mu})}{\mathbf{K}\mathbf{v}};$   
     $\mathbf{v} \leftarrow \frac{\text{prox}_{\mathbb{F}/\varepsilon}^{\text{KL}}(\mathbf{K}^\top \mathbf{u}, \mathbf{v})}{\mathbf{K}^\top \mathbf{u}};$   
**end**  
**return**  $(\mathbf{u}_i \cdot \mathbf{K}_{ij} \cdot \mathbf{v}_j)_{i,j}$

For the application of aforementioned algorithms in LC-MS, we extend the spectrum to a multidimensional signal set, a feature (cf. Figure 1.1). A feature is a convex set of peaks that corresponds to a single analyte. It contains signals across multiple MS scans of different RT and may not contain the full signal of any scan. Definitions and algorithms for computing OT between features remain the same as for spectra. For calculating cost, we use  $\ell_1$  metric, i.e. a sum of distances in M/Z and linearly scaled RT as described in the next Chapter.

An example of one of the most effective applications of optimal transport in mass spectrometry is the feature dissimilarity measure implemented in Alignstein, the retention time alignment algorithm. It is the algorithm by feature matching, i.e. it finds the correspondence of already detected features. Its key strength is properly finding the correspondence swapped order features of (cf. Figure 3.1). It is possible because the algorithm represents features by all signals contained within their boundaries. To cope with this representation, we compare features using generalized regularized optimal transport cost (2.6) (further referred to as Generalized Wasserstein Distance, [GWD](#)) described in the previous Chapter. We take advantage of computing features' similarity not only by the distance between them but also by their spatial differences (cf. Figure 1.1) and efficient scalability with dimension. Approximating of transport plan allows for comparing noisy features by introducing an appropriate penalty for omitting noise. This provides a highly flexible measure for effectively finding feature similarities and detecting non-obvious specific chromatogram patterns.

Alignstein runs in three phases (cf. Figure 5.6 and Section 3.1 for details): after appropriate preprocessing, feature centroids are clustered to find candidates for consensus features, which are then verified by the feature matching phase. During the last phase, the algorithm computes the optimal feature matching, which represents the most similar feature pairs throughout all chromatograms. We solve this problem by reducing it to finding the maximum flow of minimum cost in an appropriate flow network (cf. Fig. 3.3). Consensus features are then created from optimal feature matching with regard to initial centroid clustering. The such formulation allows for aligning chromatograms without a requirement for a reference sample or a prior feature identification. It also easily scales with a number of input chromatograms. Finally, this algorithm is not limited to correcting [RT](#) perturbations in repeated experimental runs, it also accurately aligns the majority of detected corresponding biomarkers from samples of different experimental treatments.

### 3.1 ALGORITHM FORMULATION

As an input, Alignstein takes chromatograms to be aligned and its result is a list of consensus features, i.e. a set of corresponding features

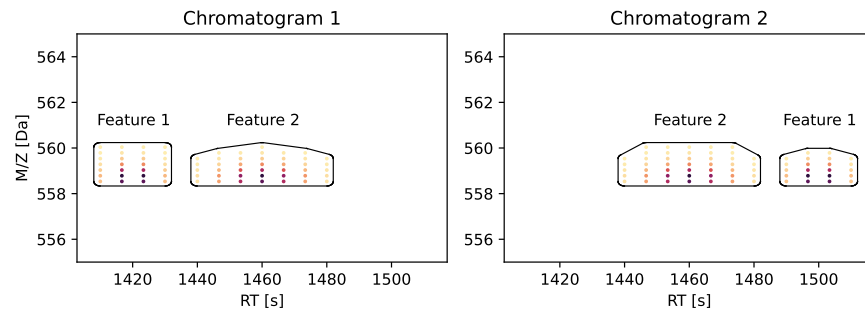


Figure 3.1: Example of swapped features representing four times charged peptides: HTALYSSDSVRNVRKKDTTG (Feature 1) and HTAIYSSDSVRNVRKKDTTG (Feature 2). Isotopic envelopes were generated using the IsoSpec tool [113] and smoothed over RT with a gaussian filter. Retention times were predicted using Pyteomics package [114]. The Euclidean distance between corresponding shifted features reduced to a point is 0.0 and 80.0, and between non-corresponding features is 40.0 and 40.0. Whereas GWD for corresponding features equals 0.3 and 80.3, and for non-corresponding features, to 46.3 and 46.3. For such an example, a simple feature matching algorithm, using GWD, would match the features correctly, and for the Euclidean distance, this solution would be ambiguous.

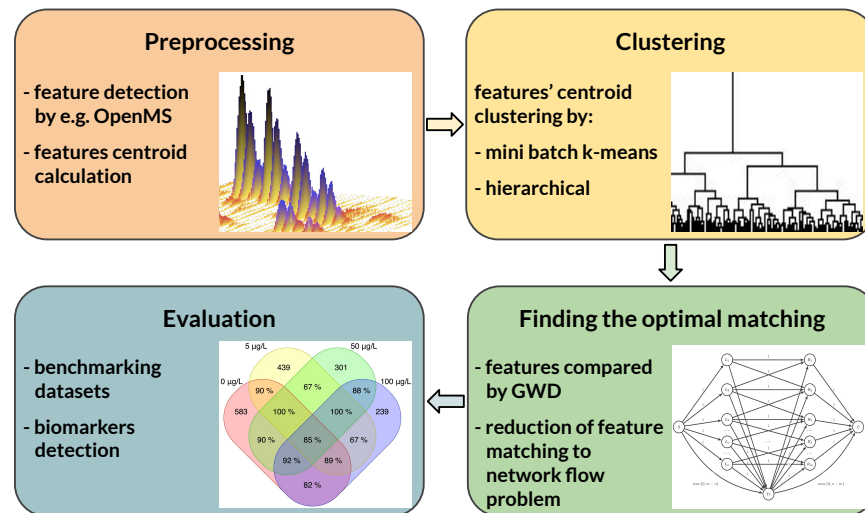


Figure 3.2: The outline of the Alignstein algorithm. It starts with feature preprocessing, for which then centroids are computed and clustered. As a next step, the problem of optimal feature matching is solved. The result is obtained with regard to prior clustering and can be further analyzed and verified.



**Algorithm 3.1:** Alignstein algorithm

```

Input: chromatograms  $ch_1, \dots, ch_n$ ,
Result: consensus features  $c_1, \dots, c_s$ 
// Preprocessing phase
features1, ..., featuresn ← preprocessFeatures( $ch_1, \dots, ch_n$ )
// Centroid clustering phase
forall featuresi do
| centroids ←
|   centroids ∪ {centroids of all features from featuresi}
end
clusters ← cluster centroids
// Matching phase
forall featuresi do
| matchingi ← match features from featuresi to
|   ∪j∈{1, ..., n} \ i featuresj
end
c1 ..., cm ←
  createConsensuFeatures(matching1, ..., matchingn)

```

from distinct chromatograms. The Alignstein algorithm pseudocode is shown as Algorithm 3.1.

**PREPROCESSING PHASE** In this phase, features are collected and prepared for further analysis as summarized in function `preprocessFeatures`. Features can be provided by the user. Otherwise, Alignstein detects them using the Feature Finder Centroided algorithm from OpenMS package [115] on the fly. Usually, software-detected features are represented only by their boundaries (e.g. `RT` and `M/Z` spans or convex hulls) and thus Alignstein collects all signal peaks contained within feature boundaries before the run. In the beginning, features are denoised by removing signals of the lowest intensity (about 1 % of signal) and normalized by its total ion current. For further processing, Alignstein scales `RT` so that the `RT` axis variation is roughly in the same order of magnitude as the `M/Z` axis variation. Scaling is done by dividing `RT` by a factor proportional to  $\frac{AL_{RT}}{AW_{M/Z}}$ , where  $AL_{RT}$  is the average feature length (along the `RT` axis) and  $AW_{M/Z}$  is average feature width (along the `M/Z` axis).

Function preprocessFeatures
<p><b>Input:</b> chromatograms <math>ch_1, \dots, ch_n</math>,  <b>Result:</b> feature sets <math>features_1, \dots, features_n</math></p> <pre> <b>forall</b> <math>ch_i</math> <b>do</b>     <math>features_{det_i} \leftarrow</math> parse or detect features in chromatogram     <math>ch_i</math>     <math>features_i \leftarrow \emptyset</math>     <b>forall</b> <math>feature_{det} \in features_{det_i}</math> <b>do</b>       <math>feature \leftarrow</math> collected signal from <math>ch_i</math> represented by       <math>feature_{det}</math>       <math>feature \leftarrow</math> normalize feature       push feature to <math>features_i</math>     <b>end</b> <b>end</b> <math>AL_{RT} \leftarrow</math> average feature length from <math>features_1, \dots, features_n</math> <math>AW_{M/Z} \leftarrow</math> average feature width from <math>features_1, \dots, features_n</math> <b>forall</b> <math>features_i</math> <b>do</b>     <b>forall</b> <math>feature \in features_i</math> <b>do</b>       scale RT of feature by factor proportional to <math>\frac{AL_{RT}}{AW_{M/Z}}</math>     <b>end</b> <b>end</b> </pre>

**CENTROID CLUSTERING PHASE** After preprocessing, Alignstein starts with the centroid clustering phase which consists of collecting centroids of all features and clustering them. Centroid clustering aims to create candidates for consensus features, which are further verified during the matching phase. Because the number of centroids from all chromatograms may be significantly large, centroid clustering is done in two steps: firstly, we split the whole space into several smaller pieces using the Mini-batch K-Means algorithm [116], then we do final precise clustering using hierarchical clustering [117].

**MATCHING PHASE** During the matching phase, Alignstein searches for feature similarities over chromatograms. It is done by matching features from every chromatogram with features of the rest of the chromatograms. Formally, for every chromatogram  $i$ , its set of features,  $features_i$  is matched to the union of features from the rest of the chromatograms:  $REST_i = \bigcup_{j \in \{1, \dots, n\} \setminus i} features_j$ . Matching can be expressed as the problem of finding the optimal matching between features from  $features_i$  to the rest of features  $REST_i$ , so that:

- every feature from  $features_i$  can be matched with at most one feature from  $REST_i$ ,
- every feature from  $REST_i$  can be matched by at most one feature from  $features_i$ ,

- for every cluster  $c_k$ , at most one feature  $f_j \in \text{REST}_i$  contained within  $c_k$  can be matched,
- for every cluster  $c_k$ , either one feature contained within it is matched or no feature is matched with a constant penalty,
- cost of matching two features  $f_1, f_2$  is proportional to **GWD** between  $f_1$  and  $f_2$ ,
- feature from  $\text{feature}_i$  can be not matched with a constant penalty,
- cost of matching is a sum of costs of matched features and penalties for not matching,
- result matching is a matching of minimal cost.

The constant penalty of not matching allows omitting to match of too different features (with **GWD** larger than a penalty threshold). Moreover, there is a restriction that for every cluster only one feature can be matched. This assures that consensus features contain at most one feature from every chromatogram. Further, every cluster consensus feature is created as described below.

We reduce the above minimization problem to finding the maximal flow of minimum cost in a network shown in Figure 3.3. Alignstein finds the optimal solution using the primal network simplex algorithm [118].

For efficiency reasons, we compute **GWD** only for pairs of features that are in the same region of interest, i.e. are close enough to be matched. It allows omitting computing **GWD** between obviously too distant features, e.g. features with an **M/Z** distance larger than 10 Da. Checking if features are in the same region of interest is done by computing feature centroids' Euclidean distance.

**CONSENSUS FEATURES CREATION** The consensus feature for cluster  $c_k$  is obtained as a union of features matched to any feature contained in  $c_k$  as shown in Function **createConsensusFeatures**.

**Function** createConsensusFeatures

```

Input: matchings  $\text{matching}_1, \dots, \text{matching}_n$ ,
//  $\text{matching}_i$  describes matching of features from  $i$ -th chromatogram
to clusters of features from the rest of chromatograms
Result: consensus features  $c_1, \dots, c_s$ 
 $s \leftarrow$  number of clusters
 $c_1, \dots, c_s \leftarrow \emptyset, \dots, \emptyset$ 
forall  $\text{matching}_i$  do
|   forall  $\langle \text{matched feature}, \text{cluster } c_k \rangle \in \text{matching}_i$  do
|   |    $c_k \leftarrow c_k \cup \{\text{feature}\}$ 
|   end
end

```

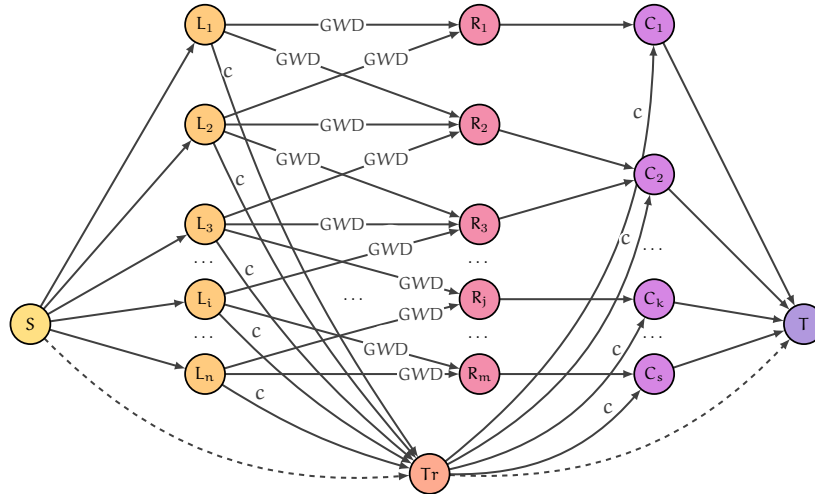


Figure 3.3: Flow network for finding the optimal feature matching between selected chromatogram, denoted by  $n$  features  $L_1, \dots, L_n$  and  $m$  features from the rest of chromatograms, denoted by nodes  $R_1, \dots, R_m$ . Clusters are denoted by  $s$  nodes  $C_1, \dots, C_s$ . Nonzero costs are described by edge labels. The cost between features  $L_i$  and features  $R_j$  is equal to  $\text{GWD}$  between them. Additional node  $\text{Tr}$  ('trash') gives the possibility to not match the feature with cost  $c$ . Every edge has a capacity equal to  $1$ , except the edge between  $S$  (source) and  $\text{Tr}$  and the edge between  $\text{Tr}$  and  $T$  (sink) with capacities equal to  $\max\{0, s - n\}$  and  $\max\{0, n - s\}$  respectively (at most one of them has nonzero capacity). Edges between  $R_1, \dots, R_m$  and  $C_1, \dots, C_s$  give the restriction that any feature can be matched with at most one cluster. As a result, we take all matchings  $(L_i, C_k)$ . We recognize the consensus feature by its cluster.

**THE SPECIAL CASE OF TWO CHROMATOGRAMS** In the special case, when only two chromatograms are aligned, the clustering phase is omitted and features are matched directly. Optimal matching is computed by minimizing the global cost of matching, i.e. the sum of  $\text{GWD}$ s between matched features and penalties for not matching. Every feature can be either matched with exactly one feature from another chromatogram or not matched with a constant penalty. Here, the constant penalty for not matching can be interpreted as a maximal distance up to which features are considered similar. Analogously as in general algorithm formulation, we reduce the feature matching problem to finding the maximum flow of minimal cost in the network described in Figure 3.4.

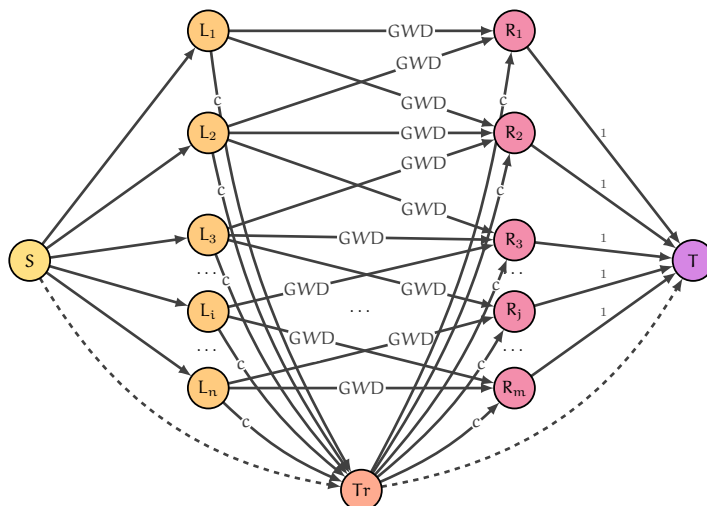


Figure 3.4: Flow network for finding the optimal feature matching between one chromatogram, denoted by  $n$  features  $L_1, \dots, L_n$  and  $m$  features from the other chromatogram, denoted by nodes  $R_1, \dots, R_m$ . Nonzero costs are described by edge labels. The cost between features  $L_i$  and features  $R_j$  is equal to  $GWD$  between them. Additional node  $Tr$  ('trash') gives the possibility to not match the feature with cost  $c$ . Every edge has a capacity equal to  $1$ , except the edge between  $S$  (source) and  $Tr$  and the edge between  $Tr$  and  $T$  (sink) with capacities equal to  $\max\{0, s - n\}$  and  $\max\{0, n - s\}$  respectively (at most one of them has nonzero capacity). As a result, we take all matchings  $(L_i, R_j)$ .

### 3.2 IMPLEMENTATION DETAILS

Alignstein is implemented as a Python 3 package and available at <https://github.com/grzsko/Alignstein>. It uses C++ implementation of  $GWD$  from the MassSinkhornmetry package. For centroid clustering, we used clustering algorithms implemented in the scikit-learn package [119]. For solving the minimum cost flow problem, we used the data structures and algorithms implemented in NetworkX [120] package.



## ALIGNSTEIN VERIFICATION

---

To assess Alignstein’s accuracy, we compared it with other [RT](#) alignment algorithms on publicly available benchmark datasets. Moreover, we checked Alignstein’s applicability as a biomarkers screening tool on the dataset of Marine mussel’s intestinal proteins. Finally, on a partially computationally generated dataset, we assessed how Alignstein resolves the correspondence of signals of swapped elution order.

### 4.1 ALGORITHM BENCHMARKING

We evaluated the accuracy of Alignstein by assessing its alignment quality on public benchmark datasets. We reproduced the evaluation protocol from the Critical Assessment of Alignment Procedures ([CAAP](#)) study [[121](#)]. It was the analysis and comparison of 7 alignment algorithms: OpenMS [[42](#)], msInspect [[122](#)], MZmine 1 [[123](#)], SpecArray [[124](#)], XAlign [[125](#)], and XCMS [[126](#)]. This study was an analysis of two proteomic datasets (P1 and P2) and two metabolomic (M1 and M2) datasets. However, due to the currently limited data availability, we omitted dataset M2. P1 set contained the analysis of *E. coli* protein extracts and consisted of 6 fractions at different salt bumps (0, 20, 40, 60, 80, and 100 mM ammonium chloride), every fraction in 2 different runs. Analogously, P2 contained the analysis of protein extract from *M. smegmatis* in 5 fractions every in 3 replicated runs. Datasets (P1 and P2) were originally available in Open Proteomic Database <http://data.marcottelab.org/MSdata/OPD/>. M1 contained the analysis of leaf tissue extract from *A. thaliana* in 44 repeated runs.

The original study describes preparation and analysis protocols for both sample sets. For every dataset and salt bump fraction, the authors prepared a set of ground truth consensus features, which represent feature correspondence over chromatograms of significantly high confidence. To assess the accuracy of alignment, the authors of the [CAAP](#) study proposed the generalization of precision and recall as alignment precision and alignment recall. Alignment precision measures how the given ground truth consensus feature was split over tool consensus features, i.e. it reflects the number of false positives. Alignment recall measures how many features of a given ground-truth consensus feature are found by the algorithm, i.e. it reflects the number of false negatives. Both alignment precision and recall are calculated as an arithmetic mean over all ground-truth consensus features. Furthermore, the authors of SIMA [[45](#)] and Wandy et al. [[52](#)] proposed the F-score which is the harmonic mean of alignment precision and align-

ment recall ( $\frac{2 \cdot P \cdot R}{P + R}$ , where P is alignment precision and R is alignment recall) to express the balance of alignment precision and alignment recall.

We used input chromatograms as mzML and mzXML files and features as featureXML files, as well as alignment precision and alignment recall evaluation script provided by the authors. Originally, in the CAAP study, only a fraction of all detected features were aligned. The evaluation protocol lacks, however, a detailed description of initial feature filtering for further alignment. Thus, we decided to filter features to those existing in ground-truth. The M1 dataset lacked the spatial information of analyzed features. For this reason, we reproduced feature detection using XCMS3 with parameters detailed in the CAAP study (method = "centWave", peakwidth = c(20, 50), snthresh = 5, ppm = 12). We matched features from the study with newly generated features by checking if the previous feature representation falls within the bounding box of new features. Such matched features were further input of the evaluation script.

We analyzed sets P1 and P2 and compared Alignstein with the results of the OpenMS alignment algorithm [42] from the CAAP study. We chose OpenMS because it achieved significantly better results than the other tools and represented a state-of-the-art solution at the time of the original study. Moreover, we included in comparison the available results of algorithms published more recently: MZMine 2 [44], SIMA [45], MassUntagler [50], and Wandy et al. [52]. MassUntangler was compared on the P1 set because it was developed only for pairwise alignment.

Alignstein obtained highly competitive results in CAAP evaluation. For the P1 dataset, it perfectly matched almost all features, its precision and recall were on average 0.94, similarly to MZmine 2 and OpenMS (cf. Table 4.1). SIMA obtained slightly worse results and the rest of the tools obtained lower values than SIMA. Interestingly, all tools achieved average alignment precision and recall no higher than 0.94. It may suggest that improperly matched features either are too distant to be matched based on LC-MS information or ground truth is misspecified.

For the P2 set, we achieved the highest average alignment recall (on average 0.82), i.e. our approach had a minimal number of unmatched features (cf. Table 4.2). It had a lower precision on average equal to 0.73 and was second only to OpenMS. Overall, we obtained the best average F-score value, equal to 0.77. For the M1 dataset, Alignstein achieved competitive results: precision equal to 0.88, recall 0.91, F-score 0.89 (cf. Table 4.3). This confirms that Alignstein scales effectively with the number of input chromatograms.

In the original CAAP study, the OpenMS alignment algorithm outperformed other tools. The authors of this study evaluated the algorithm from OpenMS version 1.0, which is no longer bundled with the OpenMS package after reimplementations in 2012. We reproduced the



SUBSET		ALIGNSTEIN	OPENMS	MZMINE 2	WANDY ET AL.	SIMA	MASSUNTANGLER
0-20	P	0,94	0,86	0,86	0,75	0,86	0,87
	R	0,94	0,86	0,86	0,79	0,83	0,76
	F	0,94	0,86	0,86	0,77	0,84	0,81
20-40	P	0,90	0,93	0,93	0,95	0,97	0,86
	R	0,90	0,93	0,93	0,95	0,94	0,73
	F	0,90	0,93	0,93	0,95	0,95	0,79
40-60	P	0,92	0,93	0,94	0,89	0,94	0,87
	R	0,92	0,93	0,94	0,86	0,91	0,8
	F	0,92	0,93	0,94	0,87	0,92	0,83
60-80	P	0,94	0,96	0,97	0,86	0,94	0,8
	R	0,94	0,96	0,97	0,9	0,92	0,68
	F	0,94	0,96	0,97	0,88	0,93	0,74
80-100	P	0,98	0,97	0,97	0,92	0,98	0,93
	R	0,98	0,97	0,97	0,92	0,96	0,89
	F	0,98	0,97	0,97	0,92	0,97	0,91
100-120	P	0,94	0,96	0,96	0,9	0,96	0,89
	R	0,94	0,96	0,96	0,92	0,95	0,87
	F	0,94	0,96	0,96	0,91	0,95	0,88

Table 4.1: Detailed results for P1 set in CAAP comparison. P stands for alignment precision, R stands for alignment recall, and F stands for F-score.

SUBSET		ALIGNSTEIN	OPENMS	MZMINE 2	WANDY ET AL.	SIMA
0	P	0,72	0,77	0,49	0,49	0,55
	R	0,84	0,65	0,56	0,48	0,61
	F	0,77	0,70	0,52	0,48	0,58
20	P	0,67	0,92	0,78	0,79	0,75
	R	0,79	0,76	0,93	0,81	0,89
	F	0,72	0,83	0,85	0,80	0,81
40	P	0,82	0,76	0,77	0,78	0,81
	R	0,88	0,74	0,78	0,82	0,75
	F	0,85	0,75	0,77	0,80	0,78
80	P	0,78	0,80	0,61	0,68	0,74
	R	0,86	0,70	0,61	0,66	0,63
	F	0,82	0,75	0,61	0,67	0,68
100	P	0,70	0,90	0,75	0,85	0,77
	R	0,80	0,75	0,88	0,85	0,89
	F	0,74	0,82	0,81	0,85	0,83

Table 4.2: Detailed results for P2 set in CAAP comparison. P stands for alignment precision, R stands for alignment recall, and F stands for F-score.

evaluation of the CAAP study on the current version of OpenMS (2.7). Unfortunately, it achieved significantly worse results despite strenuous attempts to adjust the algorithm parameters to the data. Its alignment precision and recall are on average 60 percentage points lower than the results reported in the CAAP study.

#### 4.2 DETECTING REPEATING BIOMARKERS

For assessment of Alignstein's ability to detect specific biomarkers, we analyzed chromatograms created in the work of Barranger et al. [31]. The original study aimed to measure the effects of polluting the marine mussels (*Mytilus galloprovincialis*) environment with fullerene (C60) and benzo[a]pyrene (BaP). For this purpose, the authors did a proteomic analysis.

	ALIGNSTEIN	OPENMS	MZMINE 2	SIMA
P	0.88	0.69	0.74	0.75
R	0.91	0.87	0.91	0.92
F	0.89	0.77	0.82	0.83

Table 4.3: Comparison of alignment precision (P), alignment recall (R), and F-score (F) for M1 set in CAAP study.

#### 4.2.1 Data collection and curation

Mussels were collected in Trebar with Strand, Cornwall, UK, and were exposed in vivo to C60 and BaP at concentrations 0, 5, 50, and 100  $\mu\text{g/L}$  as described in the original study. For proteomic analysis, mussel intestinal proteins were collected. After digestion and purification, the peptides were analyzed by the liquid chromatography with tandem mass spectrometry (LC-MS/MS) system with a data-dependent acquisition mode as described in Sequiera et al. [127]. In summary, peptides were separated on Dionex Ultimate 3000 RSLC nanoflow system and analyzed in an Orbitrap Velos Pro FTMS (Thermo Finnigan) with positive ion mode ionization with Proxeon nanospray ESI source. In each run, the 10 most abundant ions were further analyzed with additional collision-induced dissociation fragmentation (30 % collision energy) in a linear ion trap spectrometer. For every BaP concentration from 0, 5, 50, to 100  $\mu\text{g/L}$  three replicates were obtained. Collected chromatograms for all BaP exposure levels were deposited in the ProteomeXchange Consortium PRIDE repository (PXD013805) [128, 129].

#### 4.2.2 Data analysis

We started the analysis with peptide identification. We searched for peptides in the Uniprot KnowledgeBase [130] database of taxa Mollusca, subcategory Bivalvia from the original work, and contaminants database from Global Proteome Machine, [www.thegpm.org](http://www.thegpm.org)). We used Comet tool [131, 132] for identification. The most important search parameters were: peptide mass tolerance of 10 ppm, trypsin as search enzyme, concatenated decoy search, and allowed missed enzyme cleavages no higher than 2.

We detected features in chromatograms using the OpenMS algorithm Feature Finder in the Centroided version. We annotated the detected LC-MS features with MS/MS Comet identifications. Peptide MS/MS identifications were represented in LC-MS by retention time in seconds and the ratio of the precursor neutral mass to the assumed charge. The feature was annotated with identification when LC-MS representation of identification was enclosed within feature boundaries. For further analysis, we considered annotated features.

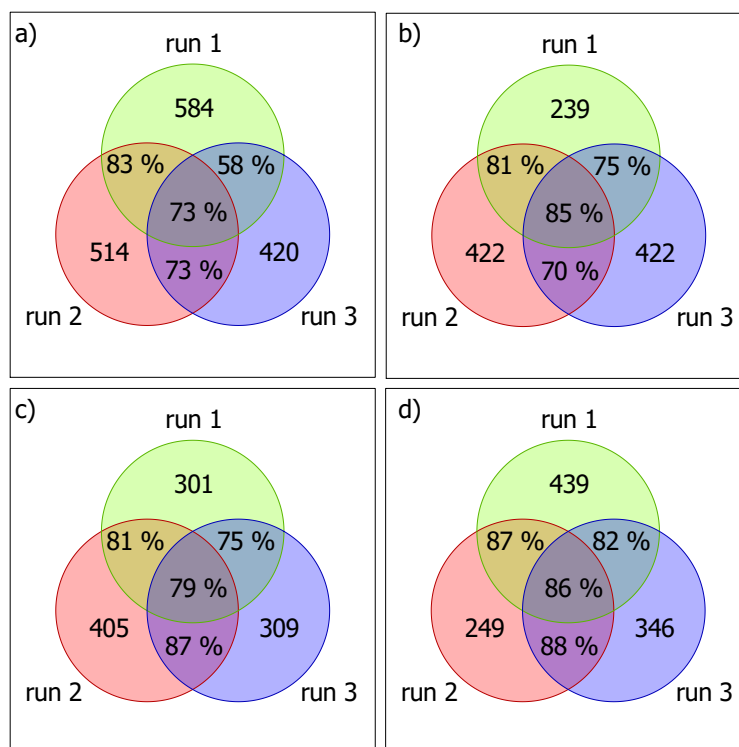


Figure 4.1: Identification recall calculated separately for identifications repeating in every chromatogram subset. a) For replicates of the sample with 0 µg/L BaP. b) For replicates of the sample with 5 µg/L BaP. c) For replicates of the sample with 50 µg/L BaP. d) For replicates of the sample with 100 µg/L BaP. Sets represent repeated runs of the same experiments, intersections contain identification recall and non-overlapping parts of the set contain the number of feature-annotated identifications.

We checked if Alignstein recognizes MS/MS information by spatial properties of LC-MS features. To quantify the accuracy of this recognition we proposed measuring the identification recall (IR) defined as follows. For every identification that repeats over chromatograms, we checked if its annotated features were properly matched by Alignstein. IR was calculated as a ratio of the number of correctly aligned annotated repeating identifications and the total number of annotated repeating identifications. For every BaP concentration, we computed IR for all aligned technical replicates of the sample. We achieved satisfactory results IR equal to 75 %, 78 %, 81 %, 86 % respectively for BaP concentrations 0, 5, 50, and 100 µg/L. As a baseline, we repeated this analysis for the OpenMS algorithm, which achieved similar results with IR equal to 81 %, 76 %, 85 %, and 83 %. Moreover, we calculated the IR separately for every subset of all aligned chromatograms. This demonstrated that our approach uniformly treats all chromatograms (cf. Figure 4.1).

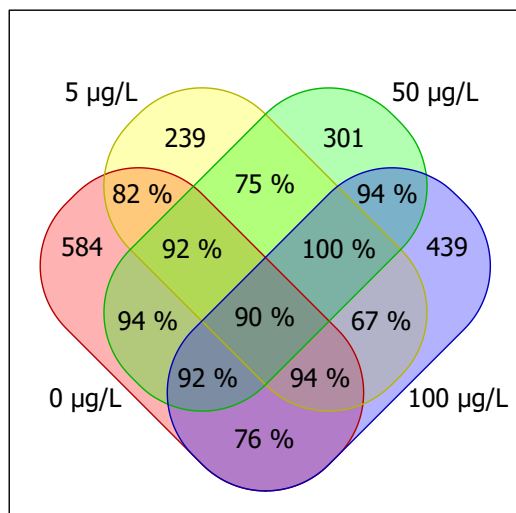


Figure 4.2: Identification recall calculated separately for identifications repeating every chromatogram subsets, for aligned chromatograms over all BaP concentrations. Sets represent chromatograms, the inconjunct part of the set contains the number of feature-annotated identifications, and conjunctions contain identification recall.

Moreover, we checked if Alignstein can also detect corresponding biomarkers for LC-MS measurements of samples under different experimental conditions. For this purpose, we repeated the analysis above by aligning chromatograms across all BaP concentrations. The overall IR was equal to 85 %. Contrary to the previous experiment, IR for OpenMS has fallen to 0.75 %. Alignstein's results were also uniform over all chromatogram subsets (cf. Figure 4.2) with IR values not lower than 67 %, reaching even 100 % for some subsets of repeated identifications. This proves that, despite the varying experimental conditions, our solution can correctly align the vast majority of corresponding features without accuracy loss. Finally, this experiment shows that it may be applied as a tool for biomarkers screening in LC-MS analysis.

#### 4.3 SWAPS EXPERIMENT

As mentioned in Section 1.1, the number of swapped features may reach even 3 % of all feature pairs. To approximate this number, we analyzed two replicates of 0 µg/L BaP concentration in the dataset described in Section 4.2.2. Computation was done for all pairs of annotated features with repeating identification in both chromatograms. We computed the fraction of these pairs that were swapped, i.e. a feature pair was considered as a swap when the feature RT means of the same identifications in two replicates were in a different order. Depending on chosen replicates, a fraction of swapped feature pairs varied from 3 % to 5 %.

We assessed that Alignstein properly matches swapped features. For this purpose, we collected over 580 identified features from the chromatograms obtained from Barranger et al.'s work [31] described in the previous section. We simulated RT drift by randomly moving features within range (-150 s, 150 s) in the RT dimension and within range (-0.3 Da, 0.3 Da) in the M/Z dimension. These two sets of features: one with original features and the second with drifted features represented chromatograms to be aligned. For such a formulation, we had full information on the fraction of feature pairs that were swapped, equal to 2 % (ca. 3400 feature pairs). We aligned these two sets and measured a number of properly matched features and a fraction of properly resolved swapped feature pairs. Our tool matched practically all drifted features (96 %) and the vast majority of swapped feature pairs (91 %). We compared our results with two open-source feature-matching algorithms: OpenMS, and LWBMatch. OpenMS had high feature matching precision, it matched the majority of drifted features (80 %). However, its accuracy drastically decreased when analyzing only swapped feature pairs (61 %). LWBMatch had a significantly lower matching precision, it matched 24 % of drifted features and only 3 % of swapped feature pairs.

#### 4.4 COMMENTS ON ALIGNSTEIN'S RESULTS

Alignstein is a novel, original algorithm for LC-MS alignment based on the GWD feature dissimilarity measure. This allows for incorporating not only distances between features, but also their spatial differences and thus more accurate feature alignment. The GWD emerges to be a key solution for correctly aligning signals with a swapped elution order, as demonstrated above.

In addition to correctly resolving feature swaps, Alignstein has more advantages over the majority of alignment algorithms. It requires no prior feature identification, so LC-MS data without additional tandem mass spectra suffice as input to the algorithm. Moreover, our approach makes no assumptions about the characteristics of the analyzed chromatograms, so it is not limited to one type of data (e.g. proteomic or metabolomic). Still, specific properties of the analyzed data (e.g. maximum drift size) can be passed as algorithm parameters. Finally, it treats uniformly all analyzed chromatograms, and thus it does not require a reference sample.

Alignstein requires only the prior feature detection as a data pre-processing step. Although approaches with this requirement are criticized [36, 48], we argue that the analysis with detected features is more accurate than the analysis of raw chromatograms. Properly executed feature detection effectively discriminates regions of high signal-to-noise ratio from chromatograms [133]. Moreover, multidimensional feature detection is crucial for collecting information about coelut-

ing ions (e.g. isotopic envelopes of compounds). Without this, any alignment algorithm might yield inaccurate results by aligning signals across isotopic envelopes.

Besides advantages, Alignstein has also limitations. It correctly matches the vast majority of features, but it happens to fail to match distant features. This mismatch can be explained by interpreting **GWD** as a sum of two costs: the cost of transporting the feature along the **RT** (to eliminate drift) and the cost of transformation (to incorporate feature-feature spatial differences). For a pair of distant, corresponding features, the cost of transport along the **RT** far exceeds the cost of transformation. For this reason, even highly dissimilar but much closer features may camouflage the correct feature correspondence. This can be particularly troublesome for complex datasets having a significant number of features, which are densely packed within chromatograms. This limitation can be only partially corrected by adjusting **GWD** parameters because the majority of corresponding feature pairs have **RT** differences of less than 10 seconds (cf. Figure 4.3) and thus the **GWD** parameters must be optimized for small feature distances. One of the possible solutions is to incorporate additional information for alignment, for example, **MS/MS** data. Thus, we plan to extend our algorithm to deal with **LC-MS/MS** datasets in a data-independent acquisition mode.

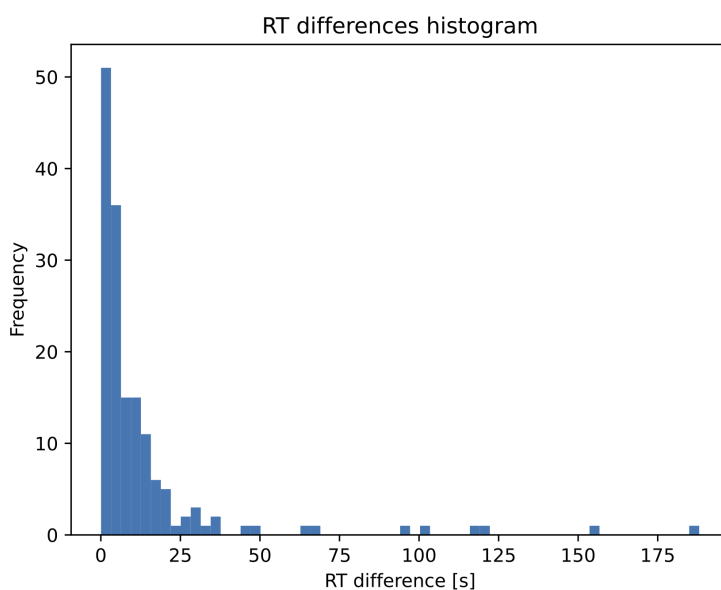


Figure 4.3: Histogram of average **RT** differences between feature pairs annotated with the same identification. The histogram is computed for chromatograms from Barranger et al. work [31], replicates of a sample with  $0 \mu\text{g/L}$  **BaP**. For better readability, outliers over 200 seconds are omitted. The majority of **RT** differences are not greater than 10 seconds.

The majority of alignment algorithms are not compared with any other tool [134]. This results in difficulties in comprehensively comparing Alignstein with the majority of algorithms. Thus, there is a constant need for dedicated LC-MS alignment assessment of currently being state-of-the-art alignment software that not only complements CAAP with other datasets but also selects the best currently available alignment algorithm. To the best of the authors' knowledge, CAAP is the only assessment of LC-MS alignment algorithms done on real datasets and thus it is widely used for validation. The limited availability of benchmark datasets may result in a growing tendency to analyze algorithms only on data from CAAP work and, therefore, to overfit to this dataset. The presented results verify that Alignstein is not affected by this problem. Not only it was validated on multiple datasets, but also almost all results are not significantly outstanding than other best-performing tools. One exception is an outstanding average recall for the P2 dataset, but it is consistent with the algorithm's design so that it maximizes the number of matches up to the user-defined parameter of the cost threshold.

In conclusion, Alignstein correctly aligns chromatograms as we have shown in the biomarkers detection experiment, by reproducing the CAAP evaluation study, as well as in swaps resolving computational comparison. Its highly competitive matching accuracy is the result of applying the generalization of the Wasserstein distance as a feature dissimilarity measure, which allows matching features without reducing feature spatial information or the dimension of data. Thus, Alignstein is capable of detecting non-obvious signal patterns and finding optimal alignment. Our solution provides a solid basis for further applications of optimal transport theory to the multidimensional problems of automated analysis in mass spectrometry. We hope that the optimal transport-based distances will become a new paradigm as a measure of spectra dissimilarity and will allow the construction of highly effective, robust, and accurate algorithms for mass spectrometry analysis.



Part III

PREDICTION OF SYNTHETIC ACCESSIBILITY



## APPROACHES TO SYNTHETIC ACCESSIBILITY SCORING

---

Retrosynthesis planning is a method of creating the synthesis scheme of chemical compounds, from low-priced, widely available precursors into target compounds. It requires, however, examining a substantial number of synthetic pathways. For this reason, the [CASP](#) approach successfully improves the retrosynthesis performance. However, the challenging problem of [CASP](#) is to properly prioritize potential synthetic pathways to those which are feasible and most promising. There is a need for an ‘oracle’ that can efficiently predict which potential intermediate molecule should be the object of interest. One of the potential solutions is predicting the synthetic accessibility of molecules. Here, we propose three approaches for the [ML](#)-based prediction of compounds’ synthetic accessibility. The first one uses stochastic gradient descent to model the distribution of a specially crafted set of descriptors and predict the likelihood of molecule structure. The second model is a supervised learning model. Its challenging element is creating a part dataset representing infeasible molecules, for which we use the bootstrap method. The last model is based on semisupervised learning designed for outliers detection. It does not require creating part of the dataset corresponding to non-existent molecules.

### 5.1 APPROACH BY MODELING MOTIFS CO-OCCURRENCES

We present a model based on the analysis of molecules’ structural patterns. The model predicts the likelihood of a molecule of a given structure. We represent a molecule using a Markov random field with nodes corresponding to the distribution of occurrences of specific structural patterns and edges corresponding to their co-occurrences. The model was trained using stochastic proximal gradient descent on a database consisting of 6 million molecules.

#### 5.1.1 *Dataset*

One of the challenges of chemoinformatics is finding the proper translation of human knowledge and intuition to computer-readable representation. To implement the expert knowledge into the model we designed a method of translating a molecule structural formula into a set of descriptors reflecting the existence of specific structural patterns in the molecule. We used a collection of over 250 carefully curated motif patterns, of which several examples are depicted in [Table 5.1](#).

Motif patterns (shorter referred to as motifs) are specific fragments of molecule structure, which correspond to specific molecular features e.g. functional groups. They are encoded in the SMARTS notation [135], a line notation for specifying substructural patterns. We used them to translate the structure of molecules into new-kind descriptors of constant size by annotating molecules with numbers of their motifs. These descriptors are further referred to as motif content descriptors.

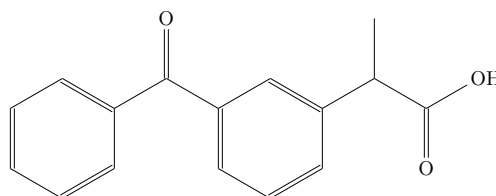
*Example.* For the sake of simplicity, suppose that our toy motif database consists of five motifs:

- carboxylic acid [#6][CX3](=O)[OH],
- ketone [#6][CX3](=O)[#6],
- phenyl c1ccccc1,
- benzyl c[CX4!H0],
- amine [CX4,c][NX3]([#1,CX4,c])[#1,CX4,c].

Suppose that we want to find motif content descriptors of ketoprofen, a popular anti-inflammatory drug. Its structure can be encoded in SMILES notation as

CC(C1=CC(=CC=C1)C(=O)C2=CC=CC=C2)C(=O)O

and its structural formula is depicted below



It contains one carboxylic acid group, one ketone group, two phenyl groups, one benzyl group, and no amine group, so its motif content descriptors are [1, 1, 2, 1, 0]. ◦

Calculating the motif content is an NP-complete problem because it is reducible to the subgraph isomorphism problem. To overcome this difficulty, we used rdkit [136] implementation of structural matching algorithms. They use well-tailored heuristics to decrease calculation time [137] and algorithms with assumed specific properties of molecules [138].

### 5.1.2 The model

As a training set, we used a database of over 600000 molecules encoded in SMILES notation [16]. By analysis of the distribution of motif content descriptors of molecules from the database, we claimed

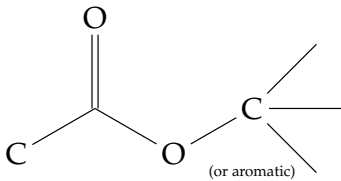
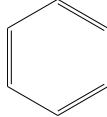
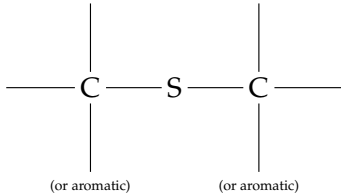
NAME	SMARTS	FORMULA
ester	<chem>[#6][CX3](=O)[O][CX4,c]</chem>	
phenyl	<chem>c1ccccc1</chem>	
thioether	<chem>[CX4,c][SX2][CX4,c]</chem>	

Table 5.1: Example motif patterns with their encoding in the SMARTS notation and a corresponding structural formula. The ester motif is described as the ester group bonded on one side with the carbon atom and the other side with the 4-valence or aromatic carbon atom. Phenyl is described as the 6-atom aromatic ring and thioether as a sulfur atom bonded with two 4-valence or aromatic carbon atoms.

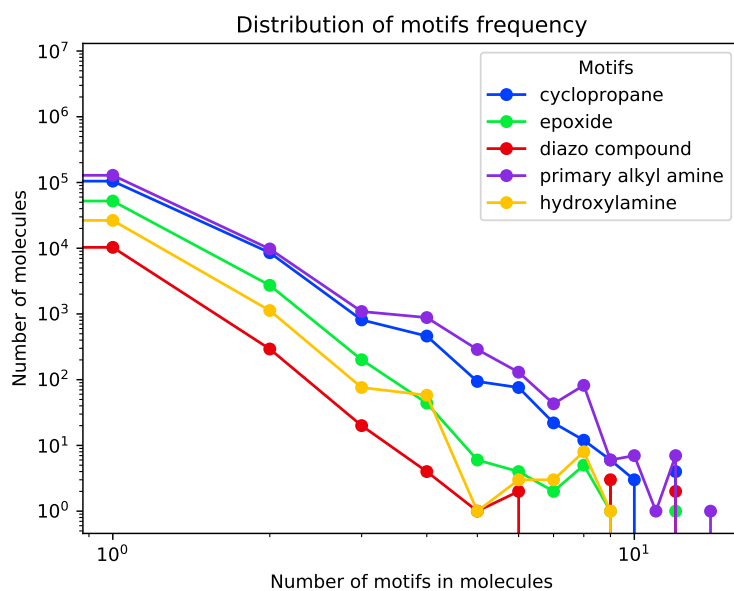


Figure 5.1: Distribution of several motifs over molecules database. For every motif, a histogram of a number of occurrences is depicted. On the log – log scale this distribution can be approximated as linear.

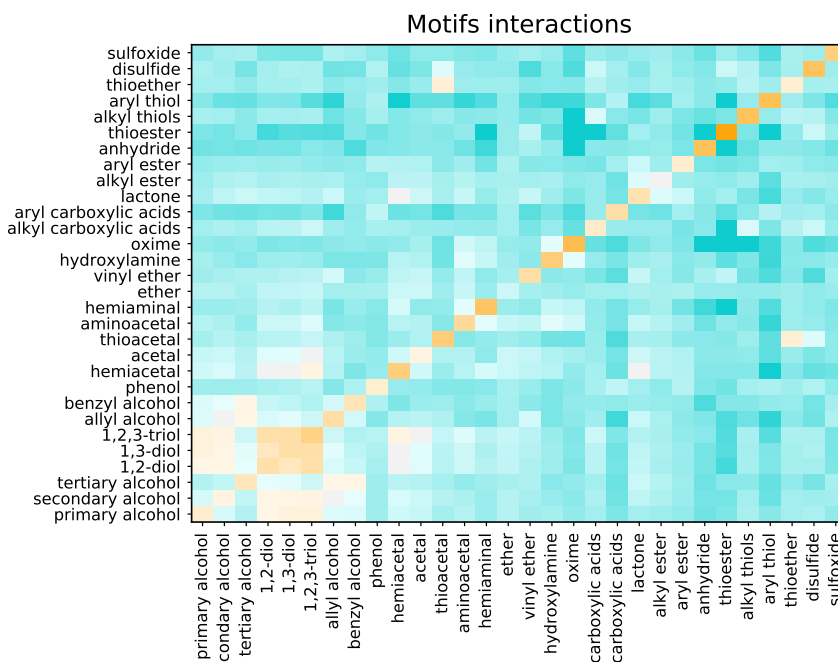


Figure 5.2: Interactions among the subset of motifs. Every cell colour corresponds to value of  $\log\left(\frac{a_{ij}}{a_i \cdot a_j}\right)$  where  $a_{ij}$  is average frequency of two motifs  $i$  and  $j$  together in molecules and  $a_i$ ,  $a_j$  are average frequency of motifs  $i$  and  $j$ . The highest values (orange) are noticeable for motifs of similar structure or function e.g. among alcohols.

that they had the zeta distribution (cf. Figure 5.1). However, motifs are not independently distributed (cf. Figure 5.2). For example, alcohols or sulfur functional groups are more likely to coexist in a single molecule. By resolving motif content distribution parameters and characteristics of their co-occurrences, we derived the joint distribution of motif content descriptors. Having a joint distribution, we were able to provide a synthetic accessibility score that describes a structural likelihood of an input molecule.

For deriving motif content distribution, denote  $M$  as a number of motifs,  $K$  as a number of observations (molecules in the database), and motif content of a single molecule as  $\mathbf{m} = (m_1, \dots, m_M)^T$ . Motif marginal distributions may be approximated with zeta distribution, i.e.:

$$\mathbb{P}(X_i = m_i) = \frac{(m_i + 1)^{-\alpha_i}}{\zeta(\alpha_i)}$$

for given coefficients vector  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_i, \dots, \alpha_M)^T$ , where  $\zeta$  is Riemann zeta function and  $X_i$  is a random variable with distribution of  $i$ -th motif.

To propose joint motif content distribution, we included fluctuating motif coincidences, referred to as motif interactions. We describe motif interactions by  $\boldsymbol{\theta} \in \mathbb{R}^{M,M}$ , where  $\theta_{ij}$  describes the interaction between

$i$ -th and  $j$ -th motifs. Then, joint motif content distribution is a product of marginal distributions with included interactions, i.e.:

$$\mathbb{P}_{\alpha\theta}(\mathbf{X} = \mathbf{m}) \propto \exp \left( \sum_i \alpha_i \log(m_i + 1) + \sum_{i < j} \theta_{ij} \mathbb{1}(m_i \neq 0) \mathbb{1}(m_j \neq 0) \right). \quad (5.1)$$

Such formulation is a Markov random field [139, 140], for which random variables of nodes correspond to distribution of  $m_i$  and edges correspond to interactions  $\theta_{ij}$ . Parameters  $\alpha$  can be derived using maximum likelihood estimator (MLE) for zeta distribution. We cannot, however, derive analytically an explicit version of the formula (5.1), and estimation of the coefficients  $\theta$  posed a challenge. We obtained  $\alpha$  and  $\theta$  by maximizing the probability  $\mathbb{P}_{\alpha\theta}(\mathbf{X} = \mathbf{m})$ . Because of the quadratic number of unknown parameters, we wanted to encompass only the most substantial ones and thus we regularized this problem by LASSO  $\ell_1$  penalty [141]:

$$(\hat{\alpha}, \hat{\theta}) = \underset{\alpha\theta}{\operatorname{argmin}} \{-\log \mathbb{P}_{\alpha\theta}(\mathbf{X} = \mathbf{m}) + \lambda \|\theta\|_1\}, \quad (5.2)$$

where  $\lambda$  is parameter of  $\ell_1$  penalty. We solved this optimization problem (5.2) using a proximal gradient method [142] as proposed by Atchadé, Fort, and Moulines [143] or by Miasojedow and Rejchel [144] for the Ising model [145]. Because we could not calculate explicitly a gradient of the function  $\log \mathbb{P}_{\alpha\theta}(\mathbf{X} = \mathbf{m})$  in formula (5.2), we estimated it instead using a stochastic version of the proximal gradient method. To this end, we used a Gibbs sampler [140], one of the Markov chain Monte Carlo methods for approximating joint distribution by creating a sequence of observations. It iteratively updates coordinates by sampling from full conditional distributions. Here, we used Metropolis-Hastings within Gibbs algorithm [146, 147], i.e. for each coordinate we did a step of Metropolis-Hastings algorithm instead of sampling from conditional distribution. The workflow of this algorithm is depicted in Figure 5.3.

To estimate a gradient, we introduced statistic  $\mathbf{t} \in \mathbb{R}^M$  averaging numbers of motifs over molecules, so that:

$$t_i = \frac{1}{K} \sum_{l=1}^K \log(m_i^l + 1)$$

where  $l$  iterates over molecules in the database. We also introduced the statistic  $\mathbf{s} \in \mathbb{R}^{M,M}$  averaging the number of motif-motif interactions over molecules, so that:

$$s_{ij} = \frac{1}{K} \sum_{l=1}^K \mathbb{1}(m_i \neq 0) \mathbb{1}(m_j \neq 0).$$

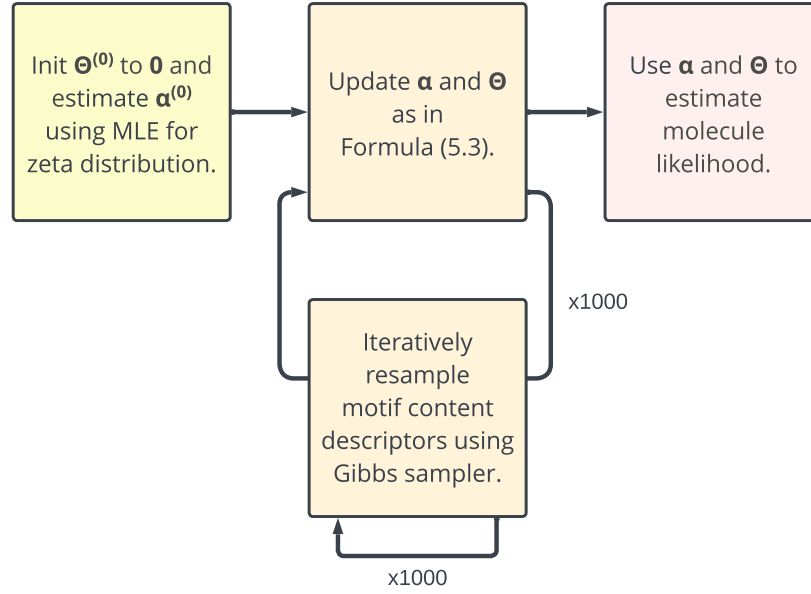


Figure 5.3: Workflow of stochastic gradient descent algorithm for detection of motif interactions.

We denoted  $\mathbf{T}_{\text{Gibbs}} = \langle \mathbf{t}^G, \mathbf{s}^G \rangle$  as  $\mathbf{t}$  and  $\mathbf{s}$  statistics over results of Gibbs sampler run and  $\mathbf{T}_{\text{obs}} = \langle \mathbf{t}^o, \mathbf{s}^o \rangle$  as  $\mathbf{t}$  and  $\mathbf{s}$  statistics over observations. The stochastic proximal gradient can be approximated using the Fisher score formula:

$$-\nabla_{\alpha, \theta} \log \mathbb{P}(\mathbf{X} = \mathbf{m}) = \mathbb{E}\mathbf{T} - \mathbf{T}_{\text{obs}} \approx \mathbf{T}_{\text{Gibbs}} - \mathbf{T}_{\text{obs}},$$

with index-wise tuple calculations.

#### Algorithm

We initialized  $\theta^{(0)}$  to zero and estimated  $\alpha^{(0)}$  using MLE for zeta distribution [148]. For every step  $s \in \{0, \dots, 1000\}$  of gradient descent, a Gibbs sampler was run  $g$  times ( $g \geq 1000$ ) and coefficients  $\alpha$ ,  $\theta$  were updated as below

$$\langle \alpha^{(s+1)}, \theta^{(s+1)} \rangle = \text{prox}_{\mathbf{I}} \left( \langle \alpha^{(s)}, \theta^{(s)} \rangle - \frac{1}{(s+1)^\xi} \mathbf{H} \right) \quad (5.3)$$

where  $\mathbf{H} = \mathbf{T}_{\text{Gibbs}} - \mathbf{T}_{\text{obs}}$ ,  $\xi$  is a parameter ( $0.5 < \xi \leq 1$ ).  $\text{prox}_{\mathbf{I}}$  is a proximal operator defined below:

$$\text{prox}_{\mathbf{I}} (\langle \alpha, \theta \rangle) = \langle \alpha, \theta' \rangle,$$

where  $\theta'$  is created from  $\theta$  with index-wise operations:

$$\theta'_{ij} = \text{sign}(\theta_{ij}) \cdot \max \left( 0, |\theta_{ij}| - \frac{\lambda}{(s+1)^\xi} \right).$$



One Metropolis within Gibbs sampler turn consisted of an iteration over motifs in a natural order and resampling motif content of some abstract molecule. This molecule was passed over steps and for the first step, its values are randomized. Content of  $i$ -th motif  $\tilde{m}_i$  was sampled as below:

1. sample its new value from the zeta distribution  $\tilde{m}_i \sim \text{zeta}(\alpha_i)$  (Metropolis step),
2. accept new value and actualize  $m_i$  with probability equal to  $\min(1, \exp(\text{Accept}(\tilde{m}_i, \mathbf{m})))$ , where:

$$\text{Accept}(\tilde{m}_i, \mathbf{m}) = \begin{cases} 0 & m_i \neq 0, \tilde{m}_i \neq 0 \\ \sum_j \theta_{ij} \mathbb{1}(m_j \neq 0) & m_i = 0, \tilde{m}_i \neq 0 \\ -\sum_j \theta_{ij} \mathbb{1}(m_j \neq 0) & m_i \neq 0, \tilde{m}_i = 0. \end{cases}$$

The complexity of the algorithm is  $O(M)$  for every step  $s$ . To speed up calculations we used the active set technique [149]. A training model phase usually takes about 20 hours of computation on a single processor.

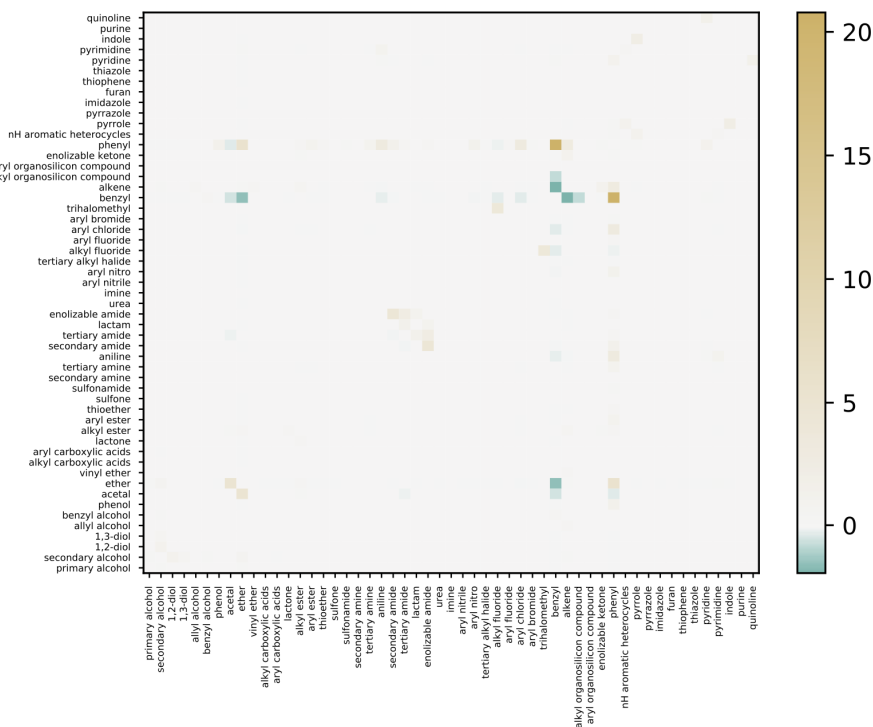


Figure 5.4: Heatmap of  $\theta$  coefficients describing motifs interactions for  $\lambda = 0.019$ . For better visibility only selected rows are displayed.

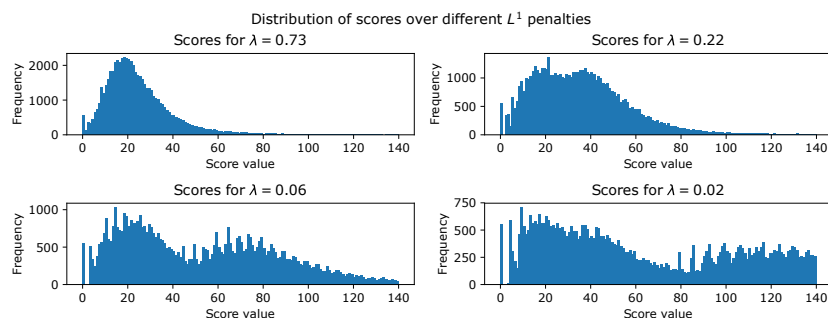


Figure 5.5: Distribution of scores over the test set for several  $\ell_1$  penalty values of  $\lambda$ . With decreasing  $\lambda$ , it is getting more binomial.

### 5.1.3 Comments on model accuracy

The model required hyperparameter tuning. While parameter  $\xi$  is rather straightforward, it is usually equal to 0.6 [150, 151], the parameter  $\lambda$  poses more challenges. We chose 20 possible values equally logarithmically distributed over  $[0.01 \cdot l, l]$ , where  $l$  is maximal value of  $\theta$  after first step, i.e.  $\max(\theta^{(1)})$ . For every  $\lambda$  value, we separately trained and tested the model. Training consisted of running 1000 steps  $s$  with resampling consisting of 1000 Gibbs sampler runs. The training set consisted of over 5983000 molecules and the test set consisted of over 60000 molecules. Figure 5.4 shows a heatmap of interactions  $\theta$  for  $\lambda$  equal to 0.019.

We noticed that the main limitation of this model is that it overfits to known and highly co-occurring interactions, for example, similar or overlapping structures of various kinds of alcohols. The design of the model was not robust to these kinds of interactions and as a result, they camouflaged non-trivial interactions appearing to only a tiny fraction of compounds. Moreover, as depicted in Figure 5.5, we notice that with decreasing value of  $\lambda$  distribution becomes more bimodal. Probably, it detected the subpopulation of molecules for which calculated model parameters are inappropriate.

Concluding, as shown by this model, molecular structural properties are non-homogenous and consist of several various subpopulations of different motif interaction schemes. The proposed model detects only a fraction of them and appears to be too general to address these issues. One of the possible solutions is changing the model design, as well as better molecular structural description, e.g. by incorporating motif spatial properties.

## 5.2 APPROACH BY SUPERVISED LEARNING

To address the limitations of the previous model, we implemented a new one, named Motif Feasibility Score (MF-Score). We incorporated

spatial properties of motif interactions in its design. To this end, we added additional two types of descriptors. Such defined descriptors were then the input of the ensemble of GBM models. The training set consisted of two fractions: representing feasible and infeasible molecules. The former one was the descriptors of molecules from the ZINC15 database [152]. The latter one was so-called decoys generated by randomizing fragments of descriptors from the feasible fraction with preserving observed motif interactions. Both the training and prediction protocols are summarized in Figure 5.6.

### 5.2.1 Descriptors and dataset

The descriptors of this model consist of three components:

- motif content descriptors,
- molecular mass,
- motif spatial descriptors.

Motif content descriptors express the existence of structural patterns in the molecule as described in Section 5.1.1. Molecular mass corresponds to molecule size and describes how motifs are packed within the molecule. Motif spatial descriptors reflect whether motif instances mutually interact and disturb the molecule stability. To create accurate motif spatial descriptors, we analyzed motif content over molecules of existing compounds. As a representative dataset of existing compounds, we used the ZINC15 database [152], from which we obtained a dataset of over 1.5 billion known organic molecules. For every molecule from the database, we calculated the motif content descriptors and interactions statistics. We defined the motif-motif interaction when almost all (over 97 %) nonzero existences of one motif imply not smaller existences of the other motif. We created 14 motif groups of highly interacting (co-occurring) motifs and 41 single non-interacting motifs. Motif spatial descriptors were represented as a matrix of maximal and minimal distances between all pairs of motif groups. Here, the distance between two motifs was defined as the length of the shortest path between motif instances in a graph corresponding to the molecule structure. Due to the high sparseness of this matrix, we reduced its dimensionality using a principal component analysis (PCA). We projected motif group interaction matrix onto 5 first principal components covering over 50 % of initial variance.

*Example.* Recall that ketoprofen has the motif content

$$[1, 1, 2, 1, 0]$$

in example toy motif database (cf. previous Example). Its molecular mass is 254.09 u. Motifs from the toy database belong to three motif groups:

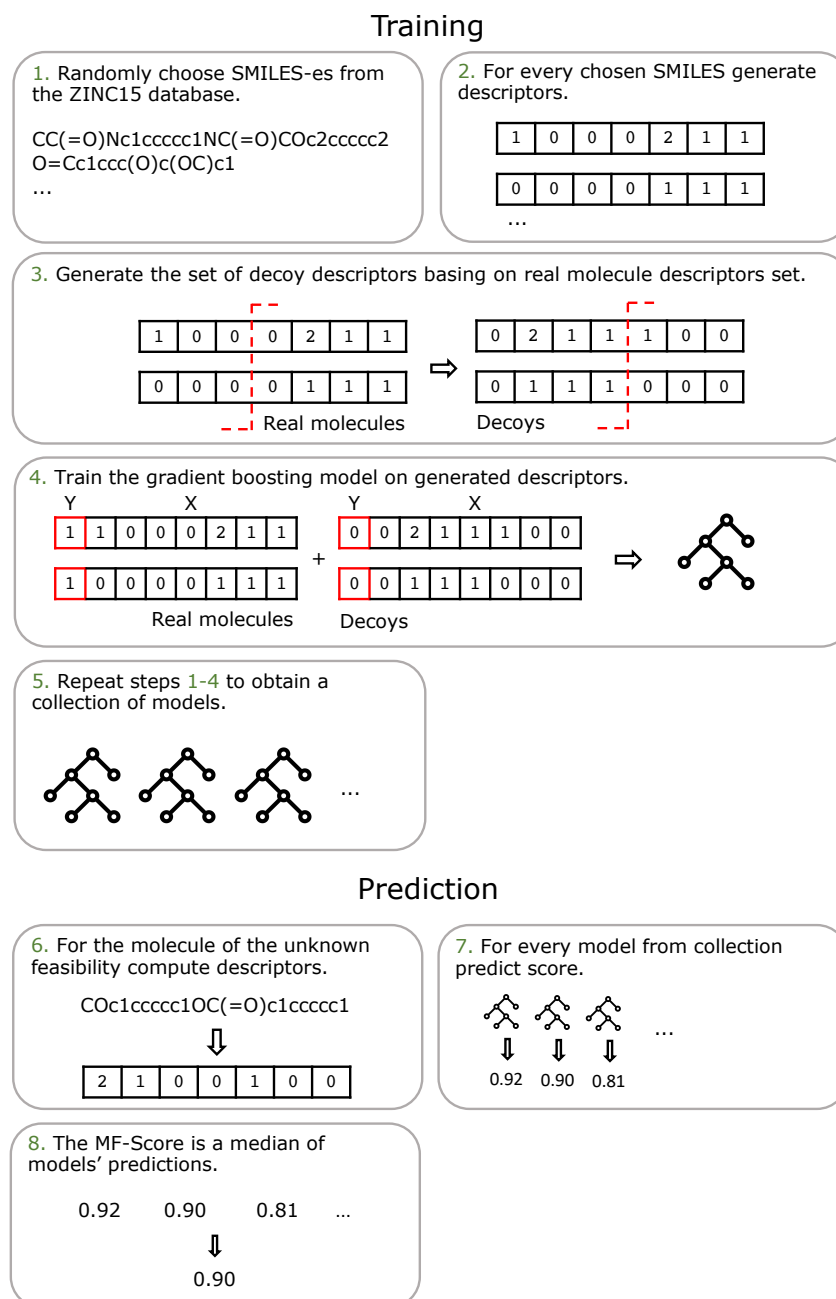
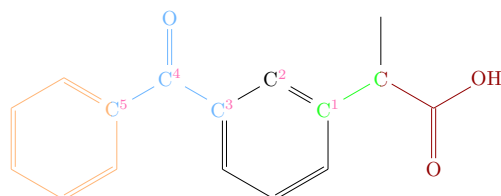


Figure 5.6: The workflow of the model training and prediction. Training phase: 1. The set of about 100000 SMILES-es of real molecules is randomly chosen from the ZINC15 database. 2. For every randomized molecule, descriptors are generated. 3. For the set of descriptors of real molecules, the set of decoys is generated using bootstrap methods. 4. The model is trained on real molecule descriptors as features with the result variable assigned to 1 (meaning feasible molecule) and the set of decoys as features with the result variable assigned to 0 (meaning infeasible molecule). 5. To obtain the collection of GBM models, procedure 1.-4. is repeated 40 times. Prediction phase: 6. Suppose, we have a molecule, which feasibility we want to predict. For this purpose, its descriptors are computed. 7. Every GBM model predicts molecule feasibility. 8. The MF-Score is the median of all model predictions.

- group A: carboxylic acid, ketone,
- group B: phenyl, benzyl,
- group C: amine.

It contains 5 motif instances — one carboxylic acid instance, one ketone instance, two phenyl instances, and one benzyl instance. Four of them are depicted below: Phenyl is annotated with orange, ketone with blue,



benzyl with green and carboxylic acid with burgundy. The maximal distance between instances of motifs from groups A (carboxylic acid, ketone) and B (phenyl, benzyl) is 5, i.e. the length of the shortest path between atoms of orange and green instances is 5 (enumerated with rose). The minimal distance between instances of motifs from groups A and B is 0 because blue and green instances are overlapping. Maximal and minimal distances between motif groups are listed in Table 5.2 which further after PCA reduction became motif spatial descriptors.

	group A	group B	group C
group A	3	5	NA
group B	0	4	NA
group C	NA	NA	NA

Table 5.2: Distances between instances of nonzero motif groups. The upper triangular part contains the maximal distance between motif instances of selected groups. The lower triangular part contains minimal distances between motifs of selected groups. Diagonal contains maximal distances between instances of the same group (minimal distances are always equal to 0). NA (not available) indicates no existing instance of a given group.

o

### 5.2.2 The model training

We used descriptors as predictors of molecule feasibility. The response variable was binary with 0 corresponding to non-existing (infeasible) molecules and 1 corresponding to the feasible molecule. Descriptors of the molecules randomized from the database represented part of the training dataset corresponding to feasible molecules. To represent infeasible ones we created a set of so-called decoys using the bootstrap

method [153]. Bootstrap is a technique of repeatedly sampling with replacement observations from the input dataset so that statistics of the input dataset are preserved. Here, we followed the assumption that randomly generated decoys may break some implicit properties of the feasible molecules and represent all possible structures that may not be feasible. Decoys were generated separately for every component of descriptors. Decoy motif content was generated for every group of motifs by copying the motif content of random feasible molecule. Decoy molecular mass was predicted by a ridge regression model trained on the motif content descriptors of feasible molecules. To create decoy spatial descriptors, we created a matrix of maximal and minimal distances between all motif groups. For every pair of motifs, the distance matrix was filled by copying distances from the randomly selected molecule with the same motif content of the given group pair. Then, we reduced the dimensionality of the distance matrix using the PCA components from the feasible part of the training set.

This dataset was the input of the GBM [154]. It is the method of iterative creating the ensemble of weaker models — here decision trees. Every new decision tree is created on the modified version of the initial data set — specifically, the dependent variable is updated with the value of current residuals. For our model, we used LightGBM [155] implementation of GBM. To obtain better stability of results and better memory usage, we created 40 GBM models trained on over 100000 molecules randomized from the database and the same number of decoys. Finally, the MF-Score was the median of all model predictions.

### 5.2.3 Model validation

We verified if the MF-Score correctly discriminates synthesizable molecules from non-synthesizable ones. We randomized 10000 molecules from the database for which we calculated descriptors. We also generated the same number of decoys. For both sets of descriptors, we predicted MF-Score. For the set of real molecules the average MF-Score value was ca. 0.7, for decoys ca. 0.3 (cf. Figure 5.7). To measure how MF-Score discriminates descriptors of real molecules from decoys, we computed the receiver operating characteristic (ROC) curve (cf. Figure 5.7). It allows for finding the best balance between the sensitivity and specificity of the classifier. We assessed the discrimination quality by calculating the area under the curve (AUC) which describes the probability that a randomly chosen real molecule would be scored better than a randomly chosen infeasible molecule. The AUC value equals 0.92 which means well discrimination between real molecules and decoys.

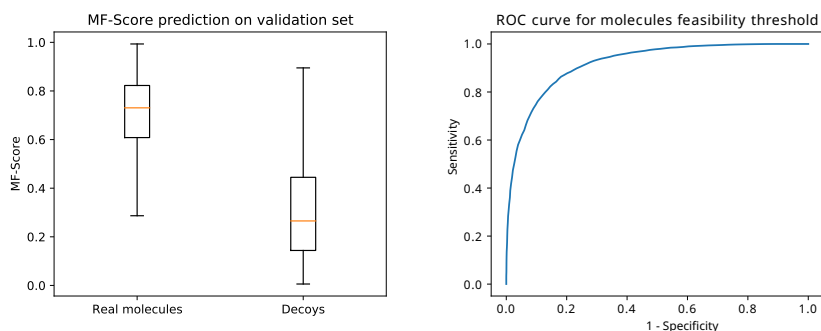


Figure 5.7: Left panel: Boxplots of MF-Score for real molecules and decoys. Right panel: ROC curve for different feasibility probability thresholds. The AUC is equal to 0.92.

#### 5.2.4 Comparison with existing solutions

We compared the results of MF-Score with the SAScore and the SCScore on 40 molecules obtained from original SAScore work [80]. A group of chemists also manually analyzed these molecules (a chemist score).

SAScore [80] is designed as a synthetic accessibility score of drug-like molecules for virtual screening exploration. It is calculated as a sum of fragment scores and complexity penalty. Fragment score is based on statistics of the frequency of extended-connectivity fingerprints of diameter 4 (ECFP<sub>4</sub>) [156] fragments from Pipeline Pilot [157] on almost one million molecules obtained from the PubChem database [158]. ECFP<sub>4</sub> is a method of creating a numeric representation of a chemical structure by traversing it, enumerating atoms, and hashing their representation. The fragment score aims to capture if fragments observed previously in the database are present in the analyzed molecule. The complexity penalty aims to capture if a molecule does not contain too many complex structures to be synthesized. It incorporates among others number of aromatic rings, stereocenters, macrocycles, or the size of the molecule. SAScore achieves values from 1 (easy to synthesize) to 10 (hard to synthesize). It is publicly available in RDKit package [136].

SCScore is a score for assessing the molecular complexity expressed as the expected number of reaction steps required to produce a target. This score was trained using neural networks [159] on the set of 12 million reactions obtained from the Reaxys database [160]. Molecules for this score are represented as 1024-bit Morgan fingerprints of radius 2 [20] which are similar to ECFP<sub>4</sub>. It achieves values from 1 (simple molecule) to 5 (complex molecule). This score was used as precursor prioritizer in ASKCOS Tree builder tool [72] and is publicly available in GitHub repository <https://github.com/connorcoley/scscore>.

We scaled the scores linearly to fit their values to the range [0, 1]. The results are depicted in Figure 5.8. We computed the Spearman

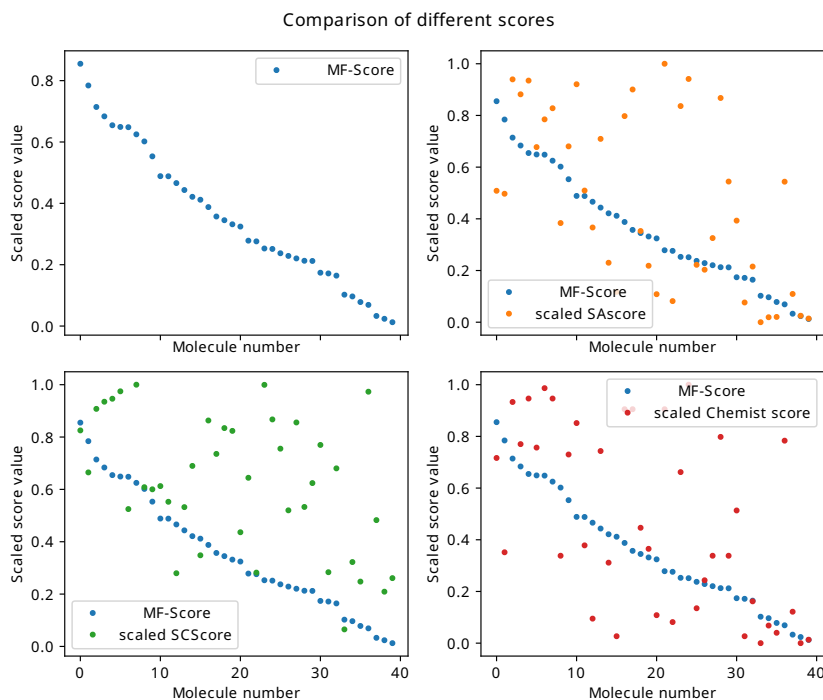


Figure 5.8: Comparison of different scores for the set of 40 molecules analyzed in Ertl et al. work [80]. Upper left: values of MF-Score for molecules ordered by the MF-Score value. Upper-right: comparison of MF-Score with scaled SAScore (correlation coefficient equal to 0.58). Bottom-left: comparison of MF-Score with scaled SCScore (correlation coefficient equal to 0.43). Bottom-right: comparison of MF-Score with scaled chemist score obtained from SAScore work [80].

rank correlation coefficient [161] of MF-Score versus SAScore and SCScore. All scores correlate, the Spearman correlation coefficient between MF-Score and SAScore equals 0.58 ( $p\text{-value} < 10^{-5}$ ) and the correlation between MF-Score and SCScore equals to 0.43 ( $p\text{-value} < 5 \cdot 10^{-3}$ ).

We also analyzed the distribution of scores of the different types of molecules. We obtained three groups of compounds from the ZINC15 database: drugs approved by the Food and Drug Administration (FDA accepted), a randomly chosen subset of secondary metabolites, which exist in nature (natural), and substances that can be easily purchased directly from the manufacturer (in-stock). Histograms of MF-Score for these substances are depicted in Figure 5.9. All substances are predicted as feasible in general (average MF-Score values between 0.6 and 0.65). SAScore distribution discriminates in-stock and FDA-accepted substances versus natural substances while the MF-Score does not. We explained this so that in-stock and FDA-accepted substances are mostly manufactured. This means that they are synthetically accessi-



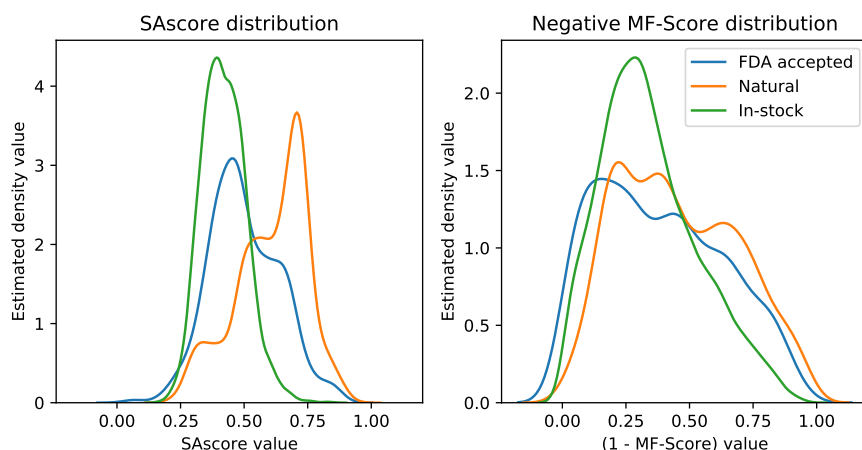


Figure 5.9: Comparison of score distributions for three substance types. Left panel: SAScore, right panel:  $1 - \text{MF-Score}$ . For every score kernel density estimator is depicted. All substances are obtained from ZINC15 database.

ble enough to be easily purchased without delivery delay. SAScore is trained with a focus on synthesizability while MF-Score focuses on more general feasibility. Secondary metabolites (natural products) are frequently difficult to synthesize but exist in nature (i.e. are feasible) and thus have different SAScore values.

### 5.2.5 GBM model predictive efficiency

We analyzed the decision structure of the single GBM model. We focused on preventing the model from overfitting. By feature importance analysis, we observed that the model primarily fits to molecular mass and spatial descriptors due to their larger variance (cf. Figure 5.10). We also observed that for motif content descriptors, the model mainly fits to motifs that contain nitrogen atoms (e.g. amide, urea), or aromatic rings (e.g. phenyl, benzyl).

### 5.2.6 Comments on model accuracy

As previously validated, the model correctly discriminates descriptors of feasible molecules from decoys. Comparison with existing tools did not reveal any inconsistencies with the expectation of results. Moreover, a massive analysis of different types of substances revealed the correct pattern of MF-Score predictions.

Analysis of single GBM model revealed, however, the risk of overfitting to obvious structural patterns. Here, feature importance analysis showed that the model may be overfitted to aromatic groups or those containing nitrogen. One of the possible reasons is the difficulty to gen-

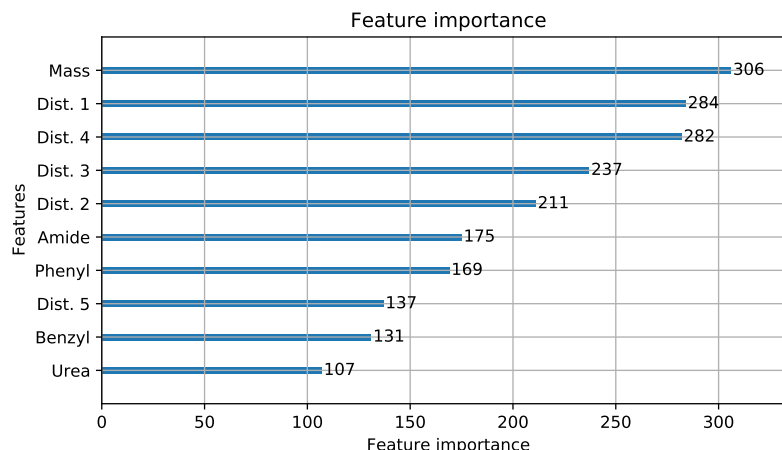


Figure 5.10: Descriptors feature importance for most important descriptors.

erate non-trivial decoys which would correctly describe non-existing molecules. This may be partially confirmed by the high accuracy of discrimination of real molecules from decoys. The additional reason may lie in the cognitive difficulty of creating a set the structural patterns that fully cover the variability of chemical compounds. Usually, their description focuses on traditional description molecules by their functional groups which in this case may be not enough.

Finally, we also considered [GBM](#) parameter tuning. We observed, however, its little impact on the decision structure. Moreover, it turned out that the most important factor for model correctness is the appropriate preparation of the training set, e.g. with a description of motif interactions that reflect real dependencies between motifs.

### 5.3 APPROACH BY SEMISUPERVISED LEARNING

#### 5.3.1 *The model and training*

To address the limitations of both previous models, we designed an OC-MF-Score based on the one-class classification approach. It allowed us to overcome the difficulty of creating appropriate decoys. Here, we used One-class Support Vector Machines ([OCSVM](#)) [[162](#)] model with radial basis function ([RBF](#)) kernel. Moreover, instead of creating descriptors of predefined structural patterns, we encoded the molecule structure using [ECFP4](#) [[156](#)], which recently have been proven to be effective for predicting various molecular properties in [ML](#) [[80](#), [83](#), [163–165](#)]. It has the advantage over manually prepared motifs because it encodes all possible structural patterns.

[ECFP4](#) is a method of creating a numeric representation of a molecule's chemical structure. To compute them, every non-hydrogen atom has assigned a numeric value (so-called *atomic invariant*), e.g. molecular mass or identifier. Atoms are then traversed in a bread-first search-like

manner starting from a randomly chosen atom up to a predefined size threshold (*radius*). Traversal subgraphs are encoded using atom invariants, then hashed, and deduplicated. Finally, all subgraphs are encoded in a bit array where the bit index corresponds to the hash value of the traversal subgraph.

OCSVM is the extension of Support Vector Machines (SVM), designed for outliers detection. It allows for detecting if the newly classified object is similar to those already observed or is an anomaly. Here, observation corresponds to ECFP<sub>4</sub> representation of the molecule. The basic idea of SVM is to find a hyperplane that separates two sets of points of given space so that margin between separated sets is maximized. Specifically, consider a set of observations  $\{(\mathbf{x}_i, y_i) : i = 1, 2, \dots, n\}$  where  $\mathbf{x}_i \in \mathbb{R}^r$  and  $y_i \in \{-1, +1\}$ . We want to separate points with positive  $y_i$  values from negative ones. SVM is capable of solving problems even with non-linear decisions boundary. It is done by non-linear transforming observations by a *feature map*  $\Phi: \mathbb{R}^r \rightarrow \mathcal{H}$  into highly dimensional *feature space*  $\mathcal{H}$  with inner product  $\langle \cdot, \cdot \rangle$  and norm  $\|\cdot\|$ , in which mapped sets became linearly separable. We separate them by a hyperplane in a feature space  $\mathcal{H}$ :

$$\beta_0 + \mathbf{x}^T \boldsymbol{\beta} = 0 \quad (5.4)$$

where  $\boldsymbol{\beta}$  is weight vector in feature space and  $\beta_0 \in \mathbb{R}$  is bias. The input dataset may, however, be noisy and contain erroneous points. To avoid overfitting to these points, a nonnegative *slack variable*  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)^T \geq \mathbf{0}_n$  is introduced. It allows for a fraction of points to lie within the margin (cf. Figure 5.11). The hyperplane (5.4) is obtained by solving the minimization problem for  $\beta_0$ ,  $\boldsymbol{\beta}$ , and  $\boldsymbol{\xi}$ :

$$\begin{aligned} \underset{\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^n \xi_i, \\ \text{subject to} \quad & \xi_i \geq 0, \\ & y_i (\beta_0 + \Phi(\mathbf{x}_i)^T \boldsymbol{\beta}) \geq 1 - \xi_i \quad i = 1, 2, \dots, n, \end{aligned} \quad (5.5)$$

where  $C > 0$  is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the point correction.

Using explicit observation mapping and solving the minimization problem (5.5) may be computationally expensive due to the high dimensionality of the feature space. Fortunately, so-called *kernel-trick* allows for avoiding computing inner products in space  $\mathcal{H}$  and exchanging them with a *kernel* function  $K: \mathbb{R}^r \times \mathbb{R}^r \rightarrow \mathbb{R}$ . The kernel of two vectors in  $\mathbb{R}^r$  gives the value of the inner product of their mappings into a feature space  $\mathcal{H}$ , i.e.  $K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ . It should be symmetric, reproducing, i.e. satisfying  $\langle K(\mathbf{x}, \cdot), K(\cdot, \mathbf{y}) \rangle = K(\mathbf{x}, \mathbf{y})$ , satisfy a Cauchy-Schwarz inequality, and be nonnegative-definite. The Mercer theorem assures that a kernel satisfying these properties is the

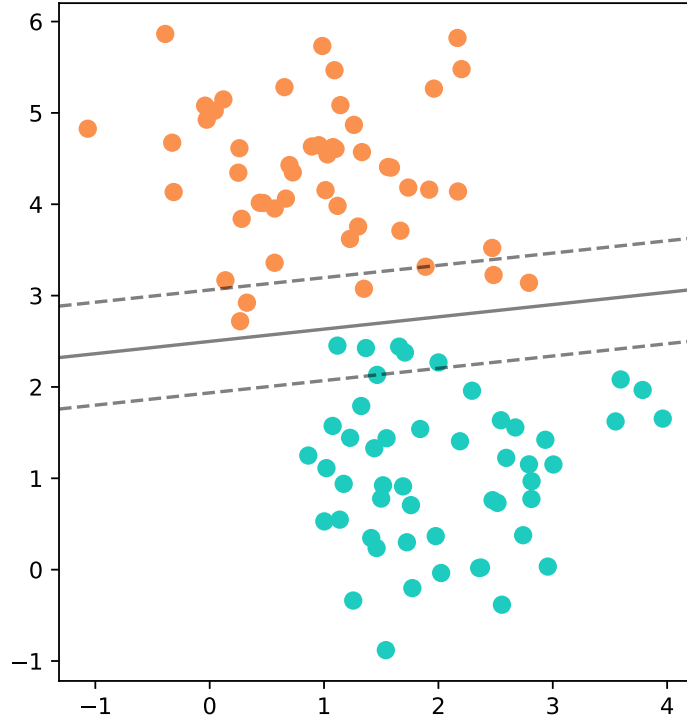


Figure 5.11: A hyperplane (solid line) separating blue data points from orange ones. The dashed line indicates the margin. Several points exceed the margin by a distance described by a slack variable. Points lying on the margin are support vectors.

inner product in a feature space  $\mathcal{H}$ . For our application, we used the [RBF](#) kernel:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \sum_{i=1}^r (x_i - y_i)^2\right)$$

where  $\gamma$  is a positive parameter of Gaussian function (cf. Figure 5.12).

[OCSVM](#) is a modification of [SVM](#) that separates observations of the one class into two sets: the origin and outliers. The parameter  $\nu$  may be interpreted as an upper bound on the fraction of observations that would be treated as outliers and a lower bound of the fraction of support vectors, i.e. vectors lying on the margin. For such formulation, previous minimization problem (5.5) gets the form

$$\begin{aligned} & \underset{\beta, \xi, \rho}{\text{minimize}} && \frac{1}{2} \|\beta\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho, \\ & \text{subject to} && \xi_i \geq 0, \\ & && (\Phi(\mathbf{x}_i)^T \beta) \geq \rho - \xi_i \quad i = 1, 2, \dots, n. \end{aligned} \tag{5.6}$$

Here,  $\rho$  moves all observations closer to the origin and thus the problem can be interpreted as finding the optimal trade-off between moving all points to the origin and a fraction of points to outliers.

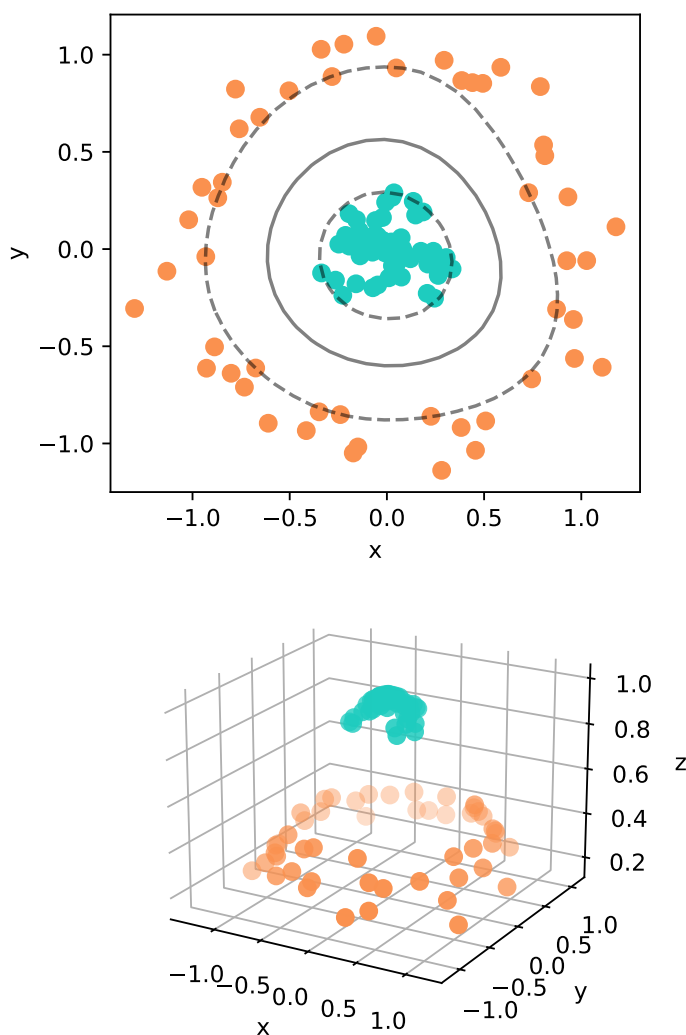


Figure 5.12: Simple example of separation two sets of points with [RBF](#) kernel (upper panel) and how they may become linearly-separable in three-dimensional feature space (bottom panel).

### 5.3.2 Model training and verification

The OC-MF-Score was trained as a [OCSVM](#) model on a representative subset of the ZINC15 database consisting of 100000 molecules (cf. Figure 5.13). All of them were encoded as [ECFP4](#) fingerprints of length 128. We set parameter  $\nu$  to 10 % and parameter  $\gamma$  to  $1/128$ .

We verified the results similarly to MF-Score by comparing with the SAscore, the SCScore, and the chemist score on 40 molecules obtained from original SAscore work (cf. Section 5.2.4). Results are depicted in Figure 5.14. OC-MF-Score highly correlates with SAscore and chemist score (Spearman correlation coefficient equal to 0.79 and 0.75 respectively) which suggests that the OC-MF-Score results are in line with expectations.

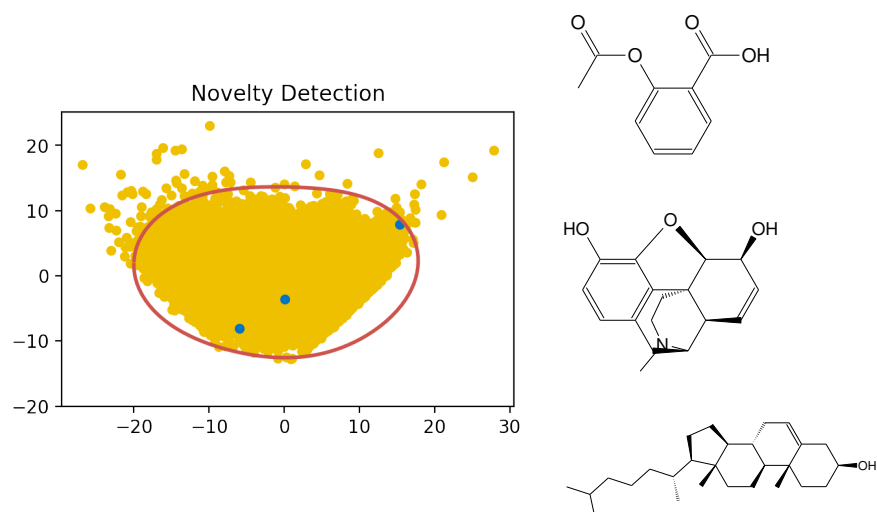


Figure 5.13: Visualization of OCSVM model on training dataset. For pictorial purposes, ECFP<sub>4</sub> is reduced using PCA. Data points are marked with yellow, the decision boundary is marked with red. Blue dots correspond to data points of aspirin (first formula), morphine (second formula), and cholesterol (third formula). Cholesterol, being a complex molecule, is close to the support vector.

Relying on a comparison with a single score may, however, result in overfitting to the verification set. Thus, the OC-MF-Score predictions may be partially unreliable. Moreover, benchmark datasets for synthetic accessibility scores are not available. We do not have any ground-truth data for synthetic accessibility scores accuracy assessment. To this end, for reliable verification of OC-MF-Score, we created a critical assessment of synthetic accessibility scores for application in retrosynthesis as described in Chapter 6.

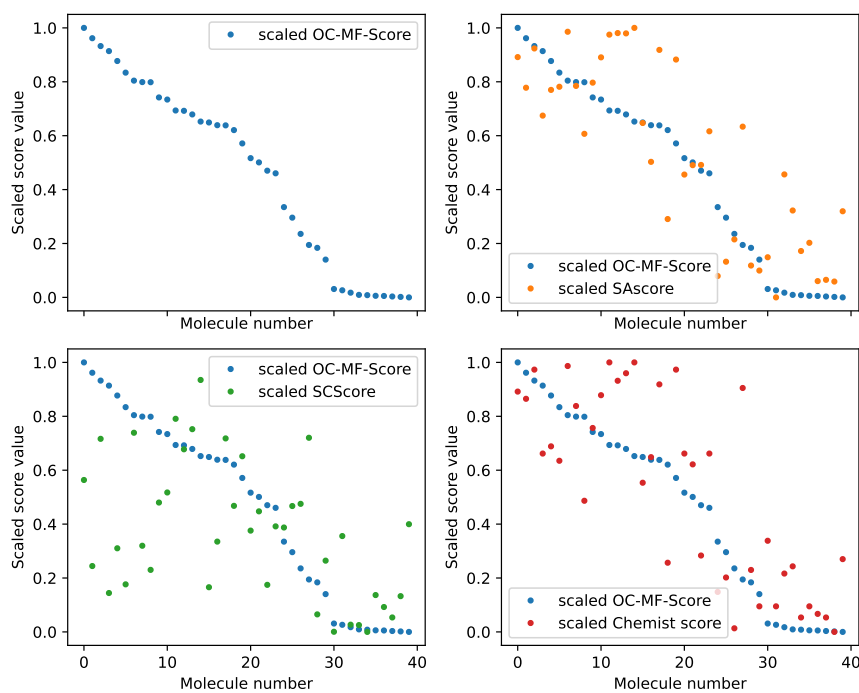


Figure 5.14: Comparison of different scores for the set of 40 molecules analyzed in Ertl et al. work [80]. Upper left: values of MF-Score for molecules ordered by the MF-Score value. Upper-right: comparison of OC-MF-Score with scaled SAscore (correlation coefficient equal to 0.79, p-value smaller than 0.001). Bottom-left: comparison of OC-MF-Score with scaled SCScore (correlation coefficient equal to 0.47, p-value equal to 0.002). Bottom-right: comparison of MF-Score with scaled chemist score (correlation coefficient equal to 0.75, p-value smaller than 0.001).





## ASSESSMENT OF SYNTHETIC ACCESSIBILITY SCORES IN COMPUTER-ASSISTED SYNTHESIS PLANNING

---

*Every empty tree is a full tree.*

— Michał Startek, Ph.D.

A wide range of ML approaches to synthetic accessibility scoring was recently designed, however, their applicability and correctness were studied to a limited extent. Moreover, there is a lack of critical assessment of synthetic accessibility scores with common test conditions. To this end, we assess if synthetic accessibility scores can reliably predict the outcomes of retrosynthesis planning. Using a specially prepared compounds database, we examine the outcomes of the retrosynthetic tool AiZynthFinder. We test whether synthetic accessibility scores: SAscore, SYBA, SCScore, and RAscore accurately predict the results of retrosynthesis planning and compare them with OC-MF-Score. Furthermore, we investigate if synthetic accessibility scores can speed up retrosynthesis planning by better prioritizing explored partial synthetic routes and thus reducing the size of the search space. For that purpose, we analyze the AiZynthFinder partial solutions search trees, their structure, and complexity parameters, such as the number of nodes, or treewidth.

This assessment is easily reproducible and is designed as a framework for evaluating and comparing novel synthetic accessibility scores. Its source code with usage instructions is publicly available at <https://github.com/grzsko/ASAP>.

### 6.1 ANALYZED SYNTHETIC ACCESSIBILITY SCORES

We analyzed four scores: SAscore, SCScore, SYBA, and RAscore, and compared them with OC-MF-Score. SAscore and SCScore were described in Section 5.2.4 and OC-MF-Score was described in Section 5.3.

The idea of the SYBA score is to train a model on comprehensive representations of both existing, easy-to-synthesize compounds as well as non-existing, hard-to-synthesize compounds. The former set was randomized from the ZINC<sub>15</sub> database and the latter set was created from an easy-to-synthesize one using Nonpher tool [166] by the iterative perturbing structure of the input molecules (adding/removing of atom or bond) up to a predefined complexity threshold. SYBA is a Bernoulli naïve Bayes classifier trained on both sets. Its implementation

is available as a Conda package or at <https://github.com/lich-uct/syba>.

RAScore is designed as a retrosynthetic accessibility score, i.e. score for fast prescreening molecules for the AiZynthFinder tool. It was trained on over 200000 molecules from ChEMBL [167]. For every molecule, a synthesis route was generated using AiZynthFinder to assess if the molecule is synthesizable. Two models were trained on these outcomes: neural network [159] and gradient boosting machines. RAScore implementation is publicly available at <https://github.com/reymond-group/RAScore>.

## 6.2 AIZYNTHFINDER, THE ANALYZED CASP TOOL

AiZynthFinder is an algorithm for computational synthesis planning. It utilizes the Monte Carlo tree search (MCTS) algorithm [168, 169], which is used for searching the tree of possible partial solutions to the analyzed problem. Here, solutions correspond to synthetic routes of the target molecule. Single MCTS round consists of 4 steps [170]: 1) selection of random leaf node, 2) expansion during which new nodes from leaf are created, 3) rollout, i.e. search simulation from new node till the complete solution or a partial solution exceeding a predefined depth, 4) backpropagation during which nodes are actualized after rollout. The node containing a partial solution is represented by i) its depth, ii) the set of in-stock molecules, and iii) the set of expandable molecules which need to be further transformed into simpler, buyable molecules. The depth of the node is defined as the maximal number of transformations that each of its molecules has to undergo to the target. A leaf node represents a complete solution if it does not need to be expanded, i.e. its list of expandable molecules is empty and its depth does not exceed a predefined threshold. Otherwise, a leaf node represents an infeasible partial solution with a depth exceeding a threshold i.e. it corresponds to the too-long synthetic route. The root node of the search tree contains a single expandable molecule representing the target compound. Nodes are connected with directed edges representing a reaction whose product is a single expandable molecule. Leaf selection is made by recursively traversing a search tree starting from the root by selecting children of maximum upper confidence bound (UCB) which expresses current node exploitation and how it is promising:

$$\text{UCB} = \frac{Q}{N_p} + U. \quad (6.1)$$

U describes how the node was already explored, i.e.:

$$U = 1.4 \cdot \sqrt{2 \cdot \frac{\ln N_{-1}}{N_p}},$$

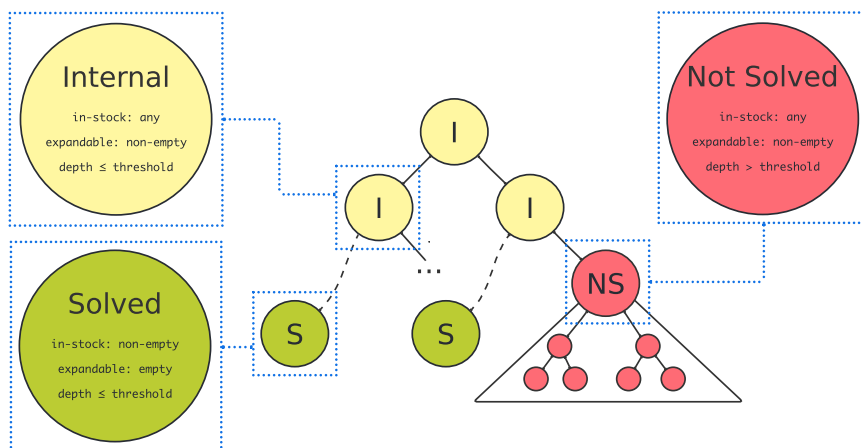


Figure 6.1: Classification of AiZynthFinder search tree nodes. Nodes are classified as: internal (I), solved (S), and not solved (NS). Internal nodes have a non-empty list of expandable molecules, but their depth is below a predefined depth. Solved nodes are leaves with all molecules in the in-stock list. A leaf marked as not solved means that it contains at least one expandable molecule and its depth exceeds a predefined threshold. Because we aim to discriminate promising nodes from non-promising ones as early as possible, we define a not solved node as all nodes that have no path to a solved leaf. In the majority of cases, we focus on roots of subtrees of not solved nodes.

where  $N_p$  is the number of times the child node has been visited, and  $N_{-1}$  is the number of times the parent node has been visited.  $Q$  describes how the node is promising, i.e. it is a sum of rewards from previous backpropagations. A single reward equals:

$$0.95 \cdot \frac{M_s}{M} + 0.05 \frac{1}{1 + \exp(m - 4)}, \quad (6.2)$$

where  $M$  is the number of molecules in the node,  $M_s$  is the number of solved molecules and  $m$  is the maximum number of transformations that every molecule have to undergo to become the root. A reward assesses how molecules of a given node are already expanded and how many steps are used. Nodes are expanded using a neural network applying reaction templates on expandable molecules in the node. Reactions are chosen so that the UCB of the product is maximized.

## 6.3 EVALUATION OF SYNTHESIS PLANNING AND SCORES

### 6.3.1 Dataset

We prepared a database of 49 compounds. The majority of these compounds are drugs or plant metabolites, of which 44 have documented synthesis. Molecules in our database were collected to represent var-

ious synthesis complexity, starting from easily synthesizable ones such as acetylsalicylic acid, to compounds of known synthesis but the more complex structure, such as morphine, compounds of known low-yielding synthesis, such as isocorydine, and not known to be synthesizable. On the other hand, the molecules were collected to represent several examples of high demand for synthesizability, such as drugs, plant metabolites, human metabolites, etc. All compounds have their structure encoded in SMILES notation from PubChem incorporating stereo orientation. Molecules from this database were further input dataset of AiZynthFinder tool and synthetic accessibility scores for their analysis.

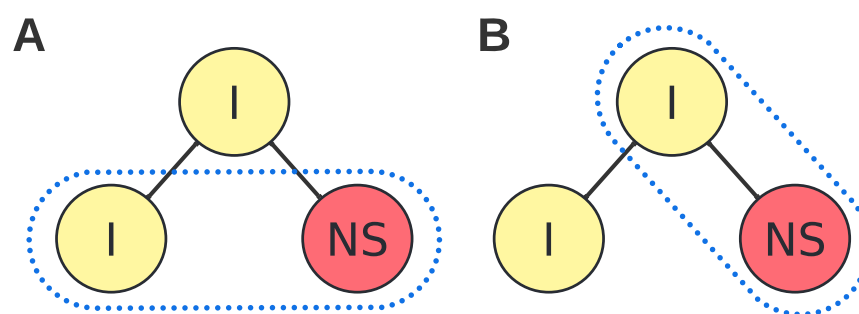


Figure 6.2: Configurations of analyzed nodes. Panel A: We checked if two nodes, internal and not solved which have the same internal parent can be discriminated by synthetic accessibility scores. Panel B: We checked also if synthetic accessibility scores can discriminate internal parents from their not solved children.

### 6.3.2 Analysis of the search trees

In the first analysis, we assessed if synthetic accessibility scores can model and predict outcomes of retrosynthesis planning. To express the complexity of retrosynthesis planning, we analyzed the search trees of AiZynthFinder runtime for molecules from our database. For these trees, we calculated statistics, such as the number of nodes, treewidth, and the number of leaf nodes that are not solved. We omitted to analyze tree depth because AiZynthFinder has strict limits for the depth of the search tree and the results would be uninformative.

Moreover, we checked if synthetic accessibility scores can act as nodes' prioritization heuristics. To this end, we classified all nodes into three groups: solved, not solved, and internal (cf. Figure 6.1). Solved nodes correspond to complete solutions, i. e. all their molecules are available in stock. Not solved nodes correspond to partial solutions of an infeasible synthetic route. We define not solved nodes as nodes for which there is no path leading to a solved node. The rest of the nodes are internal, i.e. nodes having a path to the solved leaf node. They correspond to these partial solutions which eventually lead to a

	AUC	ACCURACY
SAscore	0.90	0.81
RAscore	0.85	0.85
SCScore	0.67	0.69
SYBA	0.66	0.67
OC-MF-Score	0.53	0.51

Table 6.1: Comparison of analyzed synthetic accessibility scores in predicting the AiZynthFinder outcomes.

complete solution. For such nodes definition, if the root of a tree is not solved then the algorithm has not found any feasible synthetic route for a given target molecule. We express a score value of a node as one of the statistics (maximum, minimum, arithmetic mean) over all molecules in the node. For making calculations comparable, all scores were transformed so that they achieve values from the range [0,1] with 0 corresponding to an infeasible (non-synthesizable) molecule and 1 corresponding to a feasible (easily synthesizable) molecule. To check if synthetic accessibility scores properly prioritize nodes, we analyzed if synthetic accessibility scores discriminate internal nodes from not solved ones. Firstly, we considered these pairs connected with a single reaction step. We analyzed two configurations: i) siblings nodes internal and not solved with internal parent and ii) internal parent from not solved child (cf. Figure 6.2). Secondly, we checked if synthetic accessibility scores correctly discriminate internal nodes from not solved ones in general.

Finally, we checked if modified leaf selection, which incorporates nodes' synthetic accessibility scores, may speed up retrosynthesis planning. To this end, we modified UCB (Equation (6.1)) by substituting a fraction of a reward with one of the synthetic accessibility scores. Specifically, a reward (Equation (6.2)) was replaced with the value:

$$c \cdot \mathcal{SA} + (0.95 - c) \cdot \frac{M_s}{M} + 0.05 \frac{1}{1 + \exp(m - 4)}, \quad (6.3)$$

where  $c$  is a replaced fraction of reward ( $\frac{1}{4} \cdot 0.95$ ,  $\frac{2}{4} \cdot 0.95$ ,  $\frac{3}{4} \cdot 0.95$ ) and  $\mathcal{SA}$  is one of appropriately transformed synthetic accessibility scores.

## 6.4 RESULTS AND DISCUSSION

For all compounds from our database, we performed retrosynthesis planning using AiZynthFinder with default parameters. AiZynthFinder found a synthetic route for 22 compounds. For all found synthetic routes, 20 of them are known (precision 0.91), and for all known synthetic routes, 20 of them are found (sensitivity 0.45).

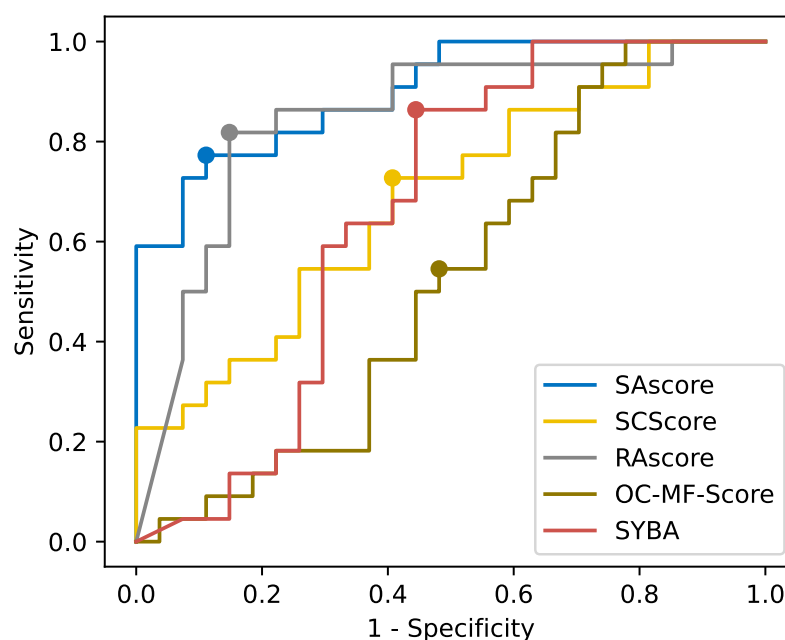


Figure 6.3: ROC curve for synthetic accessibility scores prediction of AiZynthFinder outcomes. Dots mark the best score threshold. AUC for curves are listed in Table 6.1.

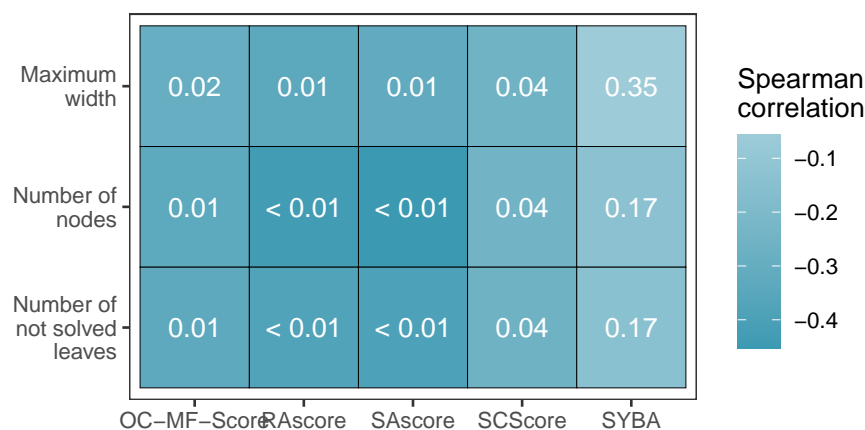


Figure 6.4: Heatmap of correlation between synthetic accessibility scores and complexity search tree parameters. The color indicates the value of the Spearman correlation, the white number indicates the p-value of the correlation test.

We assessed if synthetic accessibility scores correctly predicted the results of retrosynthetic planning. To find the optimal score thresholds that discriminate synthesizable target molecules from non-synthesizable ones, we analyzed ROC curve, cf. Figure 6.3. For every score and its optimal threshold, we computed the prediction accuracy of AiZynthFinder's outcomes. We also measured the quality of scores by calculating

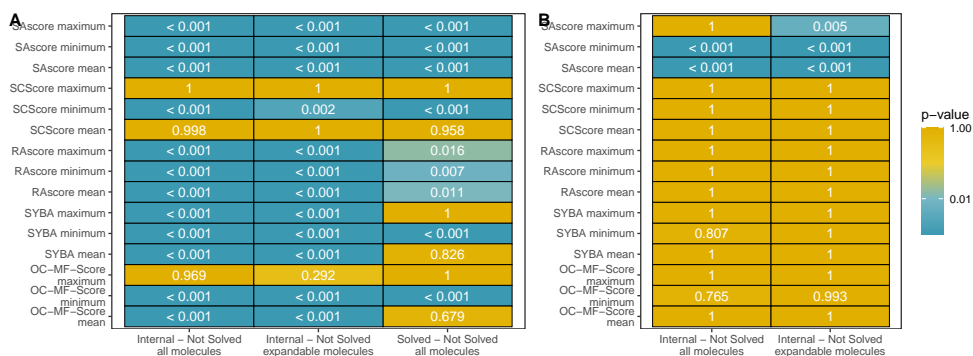


Figure 6.5: Heatmaps of t-test p-values for hypothesis whether synthetic accessibility scores discriminate node types. Panel A: For internal and not solved siblings node pairs and solved and not solved node pairs if their scaled score differences are greater than 0. Panel B: For internal parent and not solved child node pairs if their scaled score differences are greater than 0. Here, discrimination between solved and not solved is not applicable.

the AUC. Results are depicted in Table 6.1. For both AUC and accuracy, SAScore and RAScore achieve high results (both measures were over 0.81). On the contrary, for both SCScore and SYBA, the results are worse by about 20 percentage points. Unfortunately, OC-MF-Score achieves the worst results. RAScore’s good result is not surprising, because it was trained on the outcomes of the AiZynthFinder algorithm. This, combined with the low sensitivity of AiZynthFinder, allows us to claim that RAScore is a precise heuristic of AiZynthFinder outcomes, but not necessarily a synthetic accessibility score in general. The results of the SAScore may seem surprising. It is a slightly different score from the rest because it is not a standard ML model. It is designed as a combination of scores and penalties derived by experts from the presence of structural fragments in the PubChem database. From this, we infer that in retrosynthesis, human intuition and the power of the human mind still play an important role in planning a synthesis route especially in noticing the irregularities in the general synthesis rules. On the opposite, ML models are prone to imperfections, imbalance, bias, or gaps in training data. This lies in line with recent studies indicating ML limits in cheminformatics, for example for reaction yield prediction [171], for CADD [26], or for graph-based DL models for drug representation [172].

We checked also if synthetic accessibility scores can model the complexity of the retrosynthesis planning. We computed a Spearman rank correlation [161] between scores of target compounds and their search tree complexity parameters, such as treewidth, number of nodes, and number of not solved leaf nodes. Results are available in Figure 6.4. All of RAScore, SAScore, OC-MF-Score, and SCScore correlate negatively by at least one node aggregating statistic with all complexity parameters with significance below 0.04. On the contrary, SYBA does

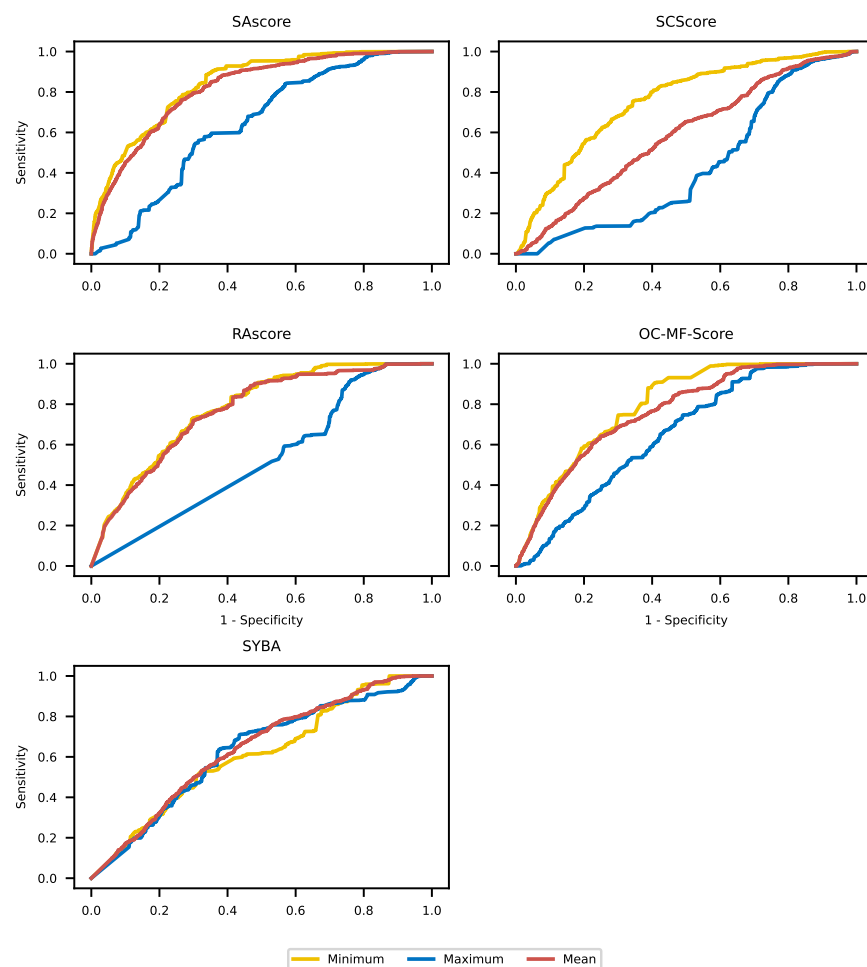


Figure 6.6: ROC curves for discrimination of internal and not solved nodes by appropriately scaled synthetic accessibility scores. AUCs are depicted in Figure 6.8.

not correlate with any of the complexity parameters. Analogously as earlier, RAScore and SAScore performed best, the strongest negative correlation was observed between these two scores and the number of nodes.

As a next step, we checked if scores can be a good heuristic for prioritizing nodes corresponding to partial solutions. Well-prioritized nodes would preferably select routes that are more promising for further search and boost the efficiency of retrosynthesis planning. To this end, we checked if synthetic accessibility scores can detect potentially infeasible partial synthesis routes. We assessed this by taking all pairs of internal and not solved siblings nodes and checking if the average score of internal nodes is greater than the score of not solved nodes (cf. Figure 6.2A). We used a one-sample t-test [173] for score differences of node pairs. The alternative hypothesis was that the mean of the pair differences distribution is greater than 0. We checked also if incorporating in-stock set molecules would not



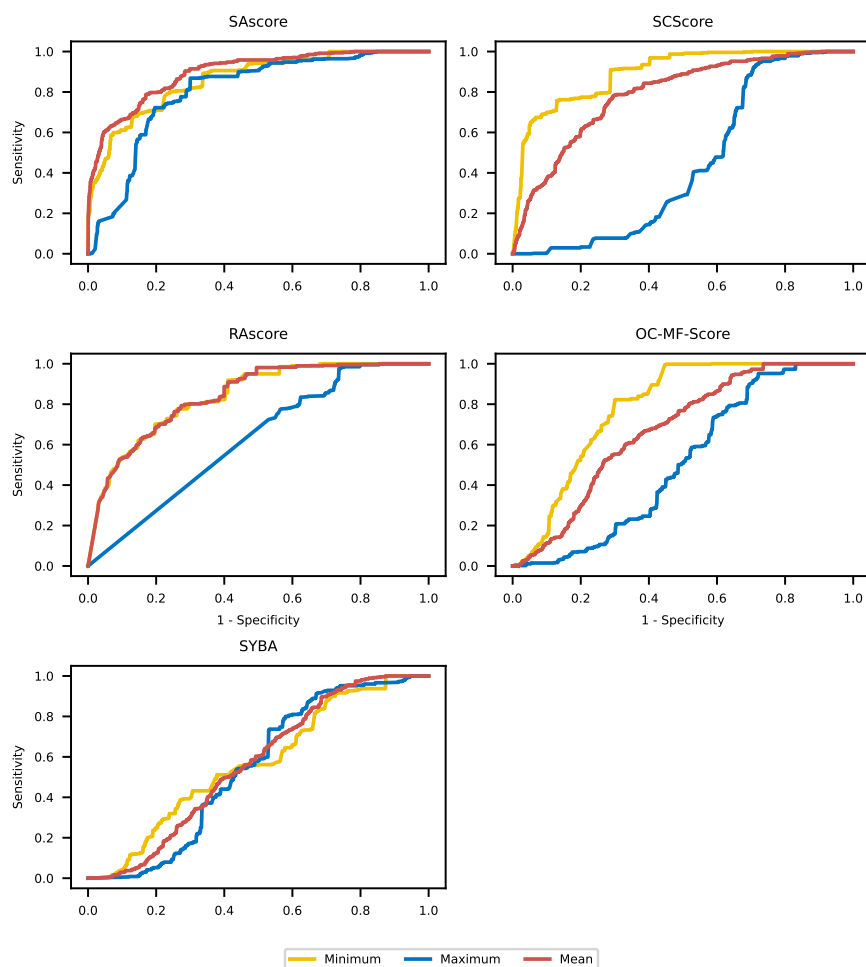


Figure 6.7: ROC curves for discrimination of solved and not solved nodes by appropriately scaled synthetic accessibility scores. AUCs are depicted in Figure 6.8.

bias the node statistics. Thus, we repeated the same test on node scores incorporating only expandable molecules. Results are depicted in Figure 6.5A. Practically, all scores with at least one aggregating statistic can correctly discriminate internal nodes from not solved and solved nodes from not solved. Omitting the set of in-stock molecules did not change the results.

We repeated the same analysis for pairs of the internal parent node and not solved child (cf. Figure 6.2B). Contrary to previous results, only SAScore can significantly discriminate the parent internal node from its not solved child (cf. Figure 6.5B).

Moreover, we checked in-depth if synthetic accessibility scores can correctly discriminate internal nodes from not solved ones and solved nodes from not solved ones. We collected all internal, solved, and not solved nodes. To find a threshold properly discriminating nodes, we analyzed ROC curves of synthetic accessibility scores, cf. Figure 6.6 and Figure 6.7. AUCs are depicted in Figure 6.8. Practically, all scores except

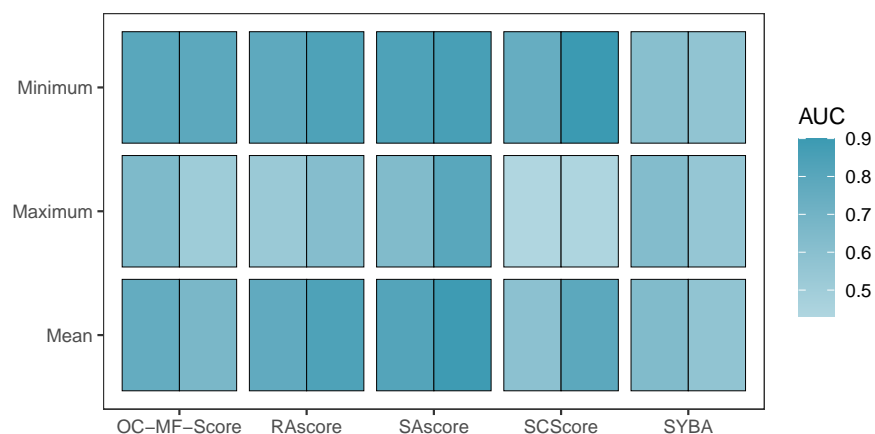


Figure 6.8: Heatmap of AUC of discrimination between internal and not solved nodes (left) and solved and not solved nodes (right).

SYBA correctly discriminate internal nodes from not solved and solved from not solved. Note that for each of the rest of the scores, only the mean and minimum aggregating functions are efficient. It is because minimum detects the presence of non-synthesizable outliers while maximum reports the best synthesizable molecules. Here, SAscore achieved the best results, and RAscore and OC-MF-Score were slightly worse. The rest of the scores were considerably worse.

Finally, we analyzed if directly replacing a fraction of the reward with an appropriately scaled synthetic accessibility score may boost the retrosynthesis planning as in Equation 6.3. If so then nodes during leaves selection would be better prioritized by UCB and thus computation time decreased. However, it turned out that this replacement did not improve significantly any parameter of search tree complexity. It may be caused by undermined reward fraction in UCB formula (6.1) or high fitting of search algorithm design to its internal scorings.

To conclude, we analyzed if synthetic accessibility scores can effectively boost the retrosynthesis process. Our analyses consisted of checking if synthetic accessibility scores correctly model retrosynthesis planning outcomes and effectively discriminate feasible partial synthetic routes from infeasible ones. We confirmed that synthetic accessibility scores can in the majority of cases well discriminate feasible molecules from infeasible ones and can be potential boosters of retrosynthesis planning tools.

Today, the big-data era requires retrosynthesis planning tools to be a fast and accurate replacement for laborious, human-mind-based manual work. We show, however, that designing retrosynthesis planning algorithms is still a challenging task and require constant improvement for faster runtime and more accurate results. For example, replacing a fraction of UCB failed to improve AiZynthFinder accuracy which suggests that synthetic accessibility scores need to be carefully crafted for the target tool.

Moreover, high, outlying SAscore results suggest that currently, pure ML techniques still do not replace completely a human mind in the retrosynthesis planning process. This implies that the accuracy of scores, although increasing, is still limited. This results in a constant need for improving the quality of training datasets, because ML models may overfit to specific properties of training datasets that appeared to be unbalanced or biased. Also, there should be constant pressure for better model design. We conclude that hybrid ML and human intuition-based synthetic accessibility scores with carefully crafted retrosynthesis planning algorithms can still efficiently boost the effectiveness of computer-assisted retrosynthesis planning. These tools may help for both finding synthetic routes of newly designed compounds as well as recognizing what is still unknown in chemistry.



## BIBLIOGRAPHY

---

- [1] S. Wold. "Spline Functions in Data Analysis." In: *Technometrics* 16.1 (1974), pp. 1–11.
- [2] Richard G. Brereton. "A Short History of Chemometrics: A Personal View." In: *Journal of Chemometrics* 28.10 (2014), pp. 749–760.
- [3] Bruce Kowalski, Steven Brown, and Bernard Vandeginste. "Editorial." In: *Journal of Chemometrics* 1.1 (1987), pp. 1–2.
- [4] Paul Geladi and Kim Esbensen. "The Start and Early History of Chemometrics: Selected Interviews. Part 1." In: *Journal of Chemometrics* 4.5 (1990), pp. 337–354.
- [5] Kim Esbensen and Paul Geladi. "The Start and Early History of Chemometrics: Selected Interviews. Part 2." In: *Journal of Chemometrics* 4.6 (1990), pp. 389–412.
- [6] Edmund R. Malinowski, Paul H. Weiner, and Alan R. Levinstone. "Factor Analysis of Solvent Shifts in Proton Magnetic Resonance." In: *The Journal of Physical Chemistry* 74.26 (Dec. 1970), pp. 4537–4542.
- [7] Alan M. Duffield, Alexander V. Robertson, Carl Djerassi, Bruce G. Buchanan, Georgia L. Sutherland, Edward A. Feigenbaum, and Joshua Lederberg. "Applications of Artificial Intelligence for Chemical Inference. II. Interpretation of Low-Resolution Mass Spectra of Ketones." In: *Journal of the American Chemical Society* 91.11 (May 1969), pp. 2977–2981.
- [8] E. J. Corey and W. Todd Wipke. "Computer-Assisted Design of Complex Organic Syntheses." In: *Science* (Oct. 1969).
- [9] E. J. Corey, Richard D. Cramer, and W. Jeffrey Howe. "Computer-Assisted Synthetic Analysis for Complex Molecules. Methods and Procedures for Machine Generation of Synthetic Intermediates." In: *Journal of the American Chemical Society* 94.2 (Jan. 1972), pp. 440–459.
- [10] N. A. B. Gray. "Dendral and Meta-Dendral — the Myth and the Reality." In: *Chemometrics and Intelligent Laboratory Systems* 5.1 (Nov. 1988), pp. 11–32.
- [11] "Chemoinformatics Theory." In: *Chemoinformatics: Theory, Practice, & Products*. Ed. by B. A. Bunin, B. Siesel, G. A. Morales, and J. Bajorath. Dordrecht: Springer Netherlands, 2007, pp. 1–49. ISBN: 978-1-4020-5001-5.

- [12] Thomas Engel. "Basic Overview of Chemoinformatics." In: *Journal of Chemical Information and Modeling* 46.6 (Nov. 2006), pp. 2267–2277.
- [13] R. D. Cramer, G. Redl, and C. E. Berkoff. "Substructural Analysis. A Novel Approach to the Problem of Drug Design." In: *Journal of Medicinal Chemistry* 17.5 (May 1974), pp. 533–535.
- [14] William J. Wiswesser. "How the WLN Began in 1949 and How It Might Be in 1999." In: *Journal of Chemical Information and Computer Sciences* 22.2 (May 1982), pp. 88–93.
- [15] Sheila Ash, Malcolm A. Cline, R. Webster Homer, Tad Hurst, and Gregory B. Smith. "SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation." In: *Journal of Chemical Information and Computer Sciences* 37.1 (Jan. 1997), pp. 71–79.
- [16] David Weininger. "SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules." In: *Journal of Chemical Information and Modeling* 28.1 (Feb. 1988), pp. 31–36.
- [17] R. L. DesJarlais, G. L. Seibel, I. D. Kuntz, P. S. Furth, J. C. Alvarez, P. R. Ortiz de Montellano, D. L. DeCamp, L. M. Babé, and C. S. Craik. "Structure-Based Design of Nonpeptide Inhibitors Specific for the Human Immunodeficiency Virus 1 Protease." In: *Proceedings of the National Academy of Sciences of the United States of America* 87.17 (Sept. 1990), pp. 6644–6648.
- [18] George W. Adamson and Judith A. Bush. "A Method for the Automatic Classification of Chemical Structures." In: *Information Storage and Retrieval* 9.10 (Oct. 1973), pp. 561–568.
- [19] Raymond E. Carhart, Dennis H. Smith, and R. Venkataraghavan. "Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications." In: *Journal of Chemical Information and Computer Sciences* 25.2 (May 1985), pp. 64–73.
- [20] H. L. Morgan. "The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service." In: *Journal of Chemical Documentation* 5.2 (May 1965), pp. 107–113.
- [21] Colin R. Groom and Frank H. Allen. "The Cambridge Structural Database in Retrospect and Prospect." In: *Angewandte Chemie International Edition* 53.3 (2014), pp. 662–671.
- [22] David W. Weisgerber. "Chemical Abstracts Service Chemical Registry System: History, Scope, and Impacts." In: *Journal of the American Society for Information Science* 48.4 (1997), pp. 349–360.
- [23] Peter Willett. "Chemoinformatics: A History." In: *WIREs Computational Molecular Science* 1.1 (2011), pp. 46–56.

- [24] Frank K. Brown. "Chapter 35 - Chemoinformatics: What Is It and How Does It Impact Drug Discovery." In: *Annual Reports in Medicinal Chemistry*. Ed. by James A. Bristol. Vol. 33. Academic Press, Jan. 1998, pp. 375–384.
- [25] Patrick Bleiziffer, Kay Schaller, and Sereina Riniker. "Machine Learning of Partial Charges Derived from High-Quality Quantum-Mechanical Calculations." In: *Journal of Chemical Information and Modeling* 58.3 (Mar. 2018), pp. 579–590.
- [26] José L. Medina-Franco. "Grand Challenges of Computer-Aided Drug Design: The Road Ahead." In: *Frontiers in Drug Discovery* 1 (2021).
- [27] Nathan Brown. "Chemoinformatics—an Introduction for Computer Scientists." In: *ACM Computing Surveys* 41.2 (Feb. 2009), 8:1–8:38.
- [28] David J. Wild. "Grand Challenges for Cheminformatics." In: *Journal of Cheminformatics* 1.1 (Mar. 2009), p. 1.
- [29] Kristin B. Runkle, Akriti Kharbanda, Ewa Stypulkowski, Xing-Jun Cao, Wei Wang, Benjamin A. Garcia, and Eric S. Witze. "Inhibition of DHHC20-Mediated EGFR Palmitoylation Creates a Dependence on EGFR Signaling." In: *Molecular Cell* 62.3 (May 2016), pp. 385–396.
- [30] Manveen K. Sethi, Morten Thaysen-Andersen, Hoguen Kim, Cheol Keun Park, Mark S. Baker, Nicolle H. Packer, Young-Ki Paik, William S. Hancock, and Susan Fanayan. "Quantitative Proteomic Analysis of Paired Colorectal Cancer and Non-Tumorigenic Tissues Reveals Signature Proteins and Perturbed Pathways Involved in CRC Progression and Metastasis." In: *Journal of Proteomics* 126 (2015), pp. 54–67.
- [31] Audrey Barranger et al. "Antagonistic Interactions between Benzo[a]Pyrene and Fullerene (C60) in Toxicological Response of Marine Mussels." In: *Nanomaterials* 9.7 (July 2019), p. 987.
- [32] Sara E. Tomechko, Guiming Liu, Mingfang Tao, Daniela Schlatzer, C. Thomas Powell, Sanjay Gupta, Mark R. Chance, and Firouz Daneshgari. "Tissue Specific Dysregulated Protein Subnetworks in Type 2 Diabetic Bladder Urothelium and Detrusor Muscle." In: *Molecular & Cellular Proteomics* 14.3 (Mar. 2015), pp. 635–645.
- [33] Bin Zhou, Jun Feng Xiao, Leepika Tuli, and Habtom W. Ressom. "LC-MS-based Metabolomics." In: 8.2 (2012), pp. 470–481.
- [34] Lloyd R. Snyder, Joseph J. Kirkland, and John W. Dolan. *Introduction to Modern Liquid Chromatography*. John Wiley & Sons, Inc., Nov. 2009.
- [35] K. Magnus Åberg, Erik Alm, and Ralf J. O. Torgrip. "The correspondence problem for metabonomics datasets." In: *Analytical and Bioanalytical Chemistry* 394.1 (2009), pp. 151–162.

- [36] R. Smith, D. Ventura, and J. T. Prince. "LC-MS Alignment in Theory and Practice: A Comprehensive Algorithmic Review." In: *Briefings in Bioinformatics* 16.1 (Nov. 2013), pp. 104–117.
- [37] Xianyin Lai, Lianshui Wang, and Frank A. Witzmann. "Issues and Applications in Label-Free Quantitative Mass Spectrometry." In: *International Journal of Proteomics* 2013 (2013), pp. 1–13.
- [38] Claudia Lindemann, Nikolas Thomanek, Franziska Hundt, Thilo Lerari, Helmut E Meyer, Dirk Wolters, and Katrin Marcus. "Strategies in relative and absolute quantitative mass spectrometry based proteomics." In: *Biological chemistry* 398.5-6 (2017), pp. 687–699.
- [39] Jürgen Cox, Marco Y. Hein, Christian A. Lubner, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. "Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ." In: *Molecular & Cellular Proteomics* 13.9 (Sept. 2014), pp. 2513–2526.
- [40] James A. Dowell, Logan J. Wright, Eric A. Armstrong, and John M. Denu. "Benchmarking Quantitative Performance in Label-Free Proteomics." In: *ACS Omega* 6.4 (2021), pp. 2494–2504.
- [41] Yunong Li and Liang Li. "Retention time shift analysis and correction in chemical isotope labeling liquid chromatography/mass spectrometry for metabolome analysis." In: *Rapid Communications in Mass Spectrometry* 34.S1 (2020), e8643.
- [42] Eva Lange, Clemens Gröpl, Ole Schulz-Trieglaff, Andreas Leinenbach, Christian Huber, and Knut Reinert. "A Geometric Approach for the Alignment of Liquid Chromatography—Mass Spectrometry Data." In: *Bioinformatics (Oxford, England)* 23.13 (July 2007), pp. i273–i281.
- [43] Arjen Lommen. "MetAlign: Interface-driven, Versatile Metabolomics Tool for Hyphenated Full-Scan Mass Spectrometry Data Pre-processing." In: *Analytical Chemistry* 81.8 (Mar. 2009), pp. 3079–3086.
- [44] Tomáš Pluskal, Sandra Castillo, Alejandro Villar-Briones, and Matej Orešič. "MZmine 2: Modular Framework for Processing, Visualizing, and Analyzing Mass Spectrometry-Based Molecular Profile Data." In: *BMC Bioinformatics* 11.1 (July 2010).
- [45] Björn Voss, Michael Hanselmann, Bernhard Y. Renard, Martin S. Lindner, Ullrich Köthe, Marc Kirchner, and Fred A. Hamprecht. "SIMA: Simultaneous Multiple Alignment of LC/MS Peak Lists." In: *Bioinformatics (Oxford, England)* 27.7 (Feb. 2011), pp. 987–993.



- [46] Zhongqi Zhang. "Retention Time Alignment of LC/MS Data by a Divide-and-Conquer Algorithm." In: *Journal of the American Society for Mass Spectrometry* 23.4 (Feb. 2012), pp. 764–772.
- [47] Hiroshi Tsugawa, Tomas Cajka, Tobias Kind, Yan Ma, Brendan Higgins, Kazutaka Ikeda, Mitsuhiro Kanazawa, Jean VanderGheynst, Oliver Fiehn, and Masanori Arita. "MS-DIAL: Data-Independent MS/MS Deconvolution for Comprehensive Metabolome Analysis." In: *Nature Methods* 12.6 (May 2015), pp. 523–526.
- [48] Shubham Gupta, Sara Ahadi, Wenyu Zhou, and Hannes Röst. "DIAAlignR Provides Precise Retention Time Alignment across Distant Runs in DIA and Targeted Proteomics." In: *Molecular & Cellular Proteomics* 18.4 (Apr. 2019), pp. 806–817.
- [49] Chiung-Ting Wu, Yizhi Wang, Yinxue Wang, Timothy Ebbels, Ibrahim Karaman, Gonçalo Graça, Rui Pinto, David M Herrington, Yue Wang, and Guoqiang Yu. "Targeted Realignment of LC-MS Profiles by Neighbor-Wise Compound-Specific Graphical Time Warping with Misalignment Detection." In: *Bioinformatics (Oxford, England)* 36.9 (Jan. 2020). Ed. by Jonathan Wren, pp. 2862–2871.
- [50] R. Ballardini, M. Benevento, G. Arrigoni, L. Pattini, and A. Roda. "MassUntangler: A Novel Alignment Tool for Label-Free Liquid Chromatography–Mass Spectrometry Proteomic Data." In: *Journal of Chromatography A* 1218.49 (Dec. 2011), pp. 8859–8868.
- [51] Jijie Wang and Henry Lam. "Graph-Based Peak Alignment Algorithms for Multiple Liquid Chromatography-Mass Spectrometry Datasets." In: *Bioinformatics (Oxford, England)* 29.19 (July 2013), pp. 2469–2476.
- [52] Joe Wandy, Rónán Daly, Rainer Breitling, and Simon Rogers. "Incorporating Peak Grouping Information for Alignment of Multiple Liquid Chromatography-Mass Spectrometry Datasets." In: *Bioinformatics (Oxford, England)* 31.12 (Feb. 2015), pp. 1999–2006.
- [53] Matthew The and Lukas Käll. "Focus on the Spectra That Matter by Clustering of Quantification Data in Shotgun Proteomics." In: *Nature Communications* 11.1 (June 2020), p. 3234.
- [54] Arun S. Moorthy and Anthony J. Kearsley. "Pattern Similarity Measures Applied to Mass Spectra." In: *SEMA SIMAI Springer Series*. Springer International Publishing, Dec. 2020, pp. 43–53.
- [55] Seongho Kim and Xiang Zhang. "Comparative Analysis of Mass Spectral Similarity Measures on Peak Alignment for Comprehensive Two-Dimensional Gas Chromatography Mass

- Spectrometry." In: *Computational and Mathematical Methods in Medicine* 2013 (2013), pp. 1–12.
- [56] Gabriel Peyré and Marco Cuturi. "Computational Optimal Transport: With Applications to Data Science." In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.
- [57] Michał Aleksander Ciach, Błażej Miasojedow, Grzegorz Skoraczyński, Szymon Majewski, Michał Startek, Dirk Valkenburg, and Anna Gambin. "Masserstein: Linear Regression of Mass Spectra by Optimal Transport." In: *Rapid Communications in Mass Spectrometry* (Sept. 2020), e8956.
- [58] Nathan A. Seifert, Kirill Prozument, and Michael J. Davis. "Computational Optimal Transport for Molecular Spectra: The Fully Discrete Case." In: *The Journal of Chemical Physics* 155.18 (Nov. 2021), p. 184101.
- [59] Nathan A. Seifert, Kirill Prozument, and Michael J. Davis. "Computational Optimal Transport for Molecular Spectra: The Semi-Discrete Case." In: *The Journal of Chemical Physics* 156.13 (Apr. 2022), p. 134117.
- [60] Olga Permiakova, Romain Guibert, Alexandra Kraut, Thomas Fortin, Anne-Marie Hesse, and Thomas Burger. "CHICKN: extraction of peptide chromatographic elution profiles from large scale mass spectrometry data by means of Wasserstein compressive hierarchical cluster analysis." In: *BMC bioinformatics* 22.1 (2021), pp. 1–30.
- [61] E. J. Corey. "General Methods for the Construction of Complex Molecules." In: *Pure and Applied Chemistry* 14.1 (Jan. 1967), pp. 19–38.
- [62] S. Hanessian, Jonathan Franco, and Benoit Larouche. "The Psychobiological Basis of Heuristic Synthesis Planning - Man, Machine and the Chiron Approach." In: *Pure and Applied Chemistry* 62.10 (Jan. 1990), pp. 1887–1910.
- [63] Wolf-Dietrich Ihlenfeldt and Johann Gasteiger. "Computer-Assisted Planning of Organic Syntheses: The Second Generation of Programs." In: *Angewandte Chemie International Edition in English* 34.23-24 (1996), pp. 2613–2633.
- [64] Ivar Ugi et al. "Computer-Assisted Solution of Chemical Problems—The Historical Development and the Present State of the Art of a New Discipline of Chemistry." In: *Angewandte Chemie International Edition in English* 32.2 (1993), pp. 201–227.
- [65] Sara Szymkuć, Ewa P. Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A. Grzybowski. "Computer-Assisted Synthetic Planning: The End of the Beginning." In: *Angewandte Chemie International Edition* 55.20 (2016), pp. 5904–5937.

- [66] Tomasz Klucznik et al. "Efficient Syntheses of Diverse, Medically Relevant Targets Planned by Computer and Executed in the Laboratory." In: *Chem* 4.3 (Mar. 2018), pp. 522–532.
- [67] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H. Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. "Predicting Retrosynthetic Pathways Using Transformer-Based Models and a Hyper-Graph Exploration Strategy." In: *Chemical Science* 11.12 (Mar. 2020), pp. 3316–3325.
- [68] Ian A. Watson, Jibo Wang, and Christos A. Nicolaou. "A Retrosynthetic Analysis Algorithm Implementation." In: *Journal of Cheminformatics* 11.1 (Jan. 2019), p. 1.
- [69] Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. "AiZynthFinder: A Fast, Robust and Flexible Open-Source Software for Retrosynthetic Planning." In: *Journal of Cheminformatics* 12.1 (Nov. 2020), p. 70.
- [70] Amol Thakkar, Thierry Kogej, Jean-Louis Reymond, Ola Engkvist, and Esben Jannik Bjerrum. "Datasets and Their Influence on the Development of Computer Assisted Synthesis Planning Tools in the Pharmaceutical Domain." In: *Chemical Science* 11.1 (Dec. 2019), pp. 154–168.
- [71] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. "Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI." In: *Nature* 555.7698 (Mar. 2018), pp. 604–610.
- [72] Xiaoxue Wang, Yujie Qian, Hanyu Gao, Connor W. Coley, Yiming Mo, Regina Barzilay, and Klavs F. Jensen. "Towards Efficient Discovery of Green Synthetic Pathways with Monte Carlo Tree Search and Reinforcement Learning." In: *Chemical Science* 11.40 (Oct. 2020), pp. 10959–10972.
- [73] Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. "Automatic Retrosynthetic Route Planning Using Template-Free Models." In: *Chemical Science* 11.12 (Mar. 2020), pp. 3355–3364.
- [74] Zhuang Wang, Wenhan Zhang, and Bo Liu. "Computational Analysis of Synthetic Planning: Past and Future." In: *Chinese Journal of Chemistry* 39.11 (2021), pp. 3127–3143.
- [75] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks." In: *Advances in neural information processing systems* 27 (2014).
- [76] Steven H. Bertz. "The First General Index of Molecular Complexity." In: *Journal of the American Chemical Society* 103.12 (June 1981), pp. 3599–3601.
- [77] Steven H. Bertz. "On the Complexity of Graphs and Molecules." In: *Bulletin of Mathematical Biology* 45.5 (Sept. 1983), pp. 849–855.

- [78] René Barone and Michel Chanon. "A New and Simple Approach to Chemical Complexity. Application to the Synthesis of Natural Products." In: *Journal of Chemical Information and Computer Sciences* 41.2 (Mar. 2001), pp. 269–272.
- [79] Krisztina Boda, Thomas Seidel, and Johann Gasteiger. "Structure and Reaction Based Evaluation of Synthetic Accessibility." In: *Journal of Computer-Aided Molecular Design* 21.6 (June 2007), pp. 311–325.
- [80] Peter Ertl and Ansgar Schuffenhauer. "Estimation of Synthetic Accessibility Score of Drug-like Molecules Based on Molecular Complexity and Fragment Contributions." In: *Journal of Cheminformatics* 1.1 (Dec. 2009), p. 8.
- [81] Milan Voršilák, Michal Kolář, Ivan Čmelo, and Daniel Svozil. "SYBA: Bayesian Estimation of Synthetic Accessibility of Organic Compounds." In: *Journal of Cheminformatics* 12.1 (May 2020), p. 35.
- [82] Jiahui Yu, Jike Wang, Hong Zhao, Junbo Gao, Yu Kang, Dongsheng Cao, Zhe Wang, and Tingjun Hou. "Organic Compound Synthetic Accessibility Prediction Based on the Graph Attention Mechanism." In: *Journal of Chemical Information and Modeling* 62.12 (June 2022), pp. 2973–2986.
- [83] Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. "SCScore: Synthetic Complexity Learned from a Reaction Corpus." In: *Journal of Chemical Information and Modeling* 58.2 (Feb. 2018), pp. 252–261.
- [84] Amol Thakkar, Veronika Chadimová, Esben Jannik Bjerrum, Ola Engkvist, and Jean-Louis Reymond. "Retrosynthetic Accessibility Score (RAScore) – Rapid Machine Learned Synthesizability Classification from AI Driven Retrosynthetic Planning." In: *Chemical Science* 12.9 (Mar. 2021), pp. 3339–3349.
- [85] Baiqing Li and Hongming Chen. "Prediction of Compound Synthesis Accessibility Based on Reaction Knowledge Graph." In: *Molecules* 27.3 (Jan. 2022), p. 1039.
- [86] Samuel Genheden and Esben Bjerrum. "PaRoutes: Towards a Framework for Benchmarking Retrosynthesis Route Predictions." In: *Digital Discovery* 1.4 (Aug. 2022), pp. 527–539.
- [87] Pascal Bonnet. "Is Chemical Synthetic Accessibility Computationally Predictable for Drug and Lead-like Molecules? A Comparative Assessment between Medicinal and Computational Chemists." In: *European Journal of Medicinal Chemistry* 54 (Aug. 2012), pp. 679–689.
- [88] Yukino Baba, Tetsu Isomura, and Hisashi Kashima. "Wisdom of Crowds for Synthetic Accessibility Evaluation." In: *Journal of Molecular Graphics and Modelling* 80 (Mar. 2018), pp. 217–223.

- [89] Grzegorz Skoraczyński, Anna Gambin, and Błażej Miasojedow. "Alignstein: Optimal Transport for Improved LC-MS Retention Time Alignment." In: *GigaScience* 11 (Nov. 2022), giac101.
- [90] Grzegorz Skoraczyński, Mateusz Kitlas, Błażej Miasojedow, and Anna Gambin. "Critical Assessment of Synthetic Accessibility Scores in Computer-Assisted Synthesis Planning." In: *Chemrxiv* (Nov. 2022).
- [91] Katty X. Wan, Ilan Vidavsky, and Michael L. Gross. "Comparing Similar Spectra: From Similarity Index to Spectral Contrast Angle." In: *Journal of the American Society for Mass Spectrometry* 13.1 (Jan. 2002), pp. 85–88.
- [92] Stephen E. Stein and Donald R. Scott. "Optimization and Testing of Mass Spectral Library Search Algorithms for Compound Identification." In: *Journal of the American Society for Mass Spectrometry* 5.9 (Sept. 1994), pp. 859–866.
- [93] Ari M Frank, Matthew E Monroe, Anuj R Shah, Jeremy J Carver, Nuno Bandeira, Ronald J Moore, Gordon A Anderson, Richard D Smith, and Pavel A Pevzner. "Spectral Archives: Extending Spectral Libraries to Analyze Both Identified and Unidentified Spectra." In: *Nature methods* 8.7 (2011), pp. 587–591.
- [94] Ryan Peckner, Samuel A. Myers, Alvaro Sebastian Vaca Jacome, Jarrett D. Egertson, Jennifer G. Abelin, Michael J. MacCoss, Steven A. Carr, and Jacob D. Jaffe. "Specter: Linear Deconvolution for Targeted Analysis of Data-Independent Acquisition Mass Spectrometry Proteomics." In: *Nature Methods* 15.5 (May 2018), pp. 371–378.
- [95] Florian Huber, Lars Ridder, Stefan Verhoeven, Jurriaan H. Spaaks, Faruk Diblen, Simon Rogers, and Justin J. J. van der Hooft. "Spec2Vec: Improved Mass Spectral Similarity Scoring through Learning of Structural Relationships." In: *PLOS Computational Biology* 17.2 (Feb. 2021). Ed. by Lars Juhl Jensen, e1008724.
- [96] L. V. Kantorovich. "Mathematical Methods of Organizing and Planning Production." In: *Management Science* 6.4 (July 1960), pp. 366–422.
- [97] Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. "Scaling Algorithms for Unbalanced Optimal Transport Problems." In: *Mathematics of Computation* 87.314 (Feb. 2018), pp. 2563–2609.
- [98] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. "The Earth Mover's Distance as a Metric for Image Retrieval." In: *International Journal of Computer Vision* 40 (2000), p. 2000.

- [99] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*. First. Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser Cham.
- [100] Szymon Majewski, Michal Aleksander Ciach, Michal Startek, Wanda Niemyska, Blazej Miasojedow, and Anna Gambin. “The Wasserstein Distance as a Dissimilarity Measure for Mass Spectra with Application to Spectral Deconvolution.” In: *18th International Workshop on Algorithms in Bioinformatics (WABI 2018)*. Ed. by Laxmi Parida and Esko Ukkonen. Vol. 113. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 25:1–25:21.
- [101] George B. Dantzig. *Origins of the Simplex Method*. Ed. by Stephen G. Nash. 1990.
- [102] Paul Knopp and Richard Sinkhorn. “Concerning Nonnegative Matrices and Doubly Stochastic Matrices.” In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.
- [103] R. Cominetti and J. San Martín. “Asymptotic Analysis of the Exponential Penalty Trajectory in Linear Programming.” In: *Mathematical Programming* 67.1 (Oct. 1994), pp. 169–187.
- [104] Zeyuan Allen-Zhu, Yuanzhi Li, R. Oliveira, and A. Wigderson. “Much Faster Algorithms for Matrix Scaling.” In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017).
- [105] Marco Cuturi. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport.” In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013.
- [106] Richard Sinkhorn. “A Relationship between Arbitrary Positive Matrices and Doubly Stochastic Matrices.” In: *The Annals of Mathematical Statistics* 35.2 (June 1964), pp. 876–879.
- [107] Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. “Near-Linear Time Approximation Algorithms for Optimal Transport via Sinkhorn Iteration.” In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.
- [108] Joel Franklin and Jens Lorenz. “On the Scaling of Multidimensional Matrices.” In: *Linear Algebra and its Applications*. Special Issue Dedicated to Alan J. Hoffman 114–115 (Mar. 1989), pp. 717–735.
- [109] Rémi Flamary et al. “POT: Python Optimal Transport.” In: *Journal of Machine Learning Research* 22.78 (2021), pp. 1–8.

- [110] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.
- [111] Thibault Sejourne, Francois-Xavier Vialard, and Gabriel Peyré. "The Unbalanced Gromov Wasserstein Distance: Conic Formulation and Relaxation." In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 8766–8779.
- [112] Gaël Guennebaud, Benoît Jacob, et al. *Eigen V3*. 2010.
- [113] Mateusz K. Lacki, Dirk Valkenborg, and Michal P. Startek. "IsoSpec2: Ultrafast Fine Structure Calculator." In: *Analytical Chemistry* 92.14 (June 2020), pp. 9472–9475.
- [114] Lev I. Levitsky, Joshua A. Klein, Mark V. Ivanov, and Mikhail V. Gorshkov. "Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework." In: *Journal of Proteome Research* 18.2 (Dec. 2018), pp. 709–714.
- [115] Hannes L Röst et al. "OpenMS: A Flexible Open-Source Software Platform for Mass Spectrometry Data Analysis." In: *Nature Methods* 13.9 (Aug. 2016), pp. 741–748.
- [116] D. Sculley. "Web-Scale k-Means Clustering." In: *Proceedings of the 19th International Conference on World Wide Web - WWW 10*. ACM Press, 2010.
- [117] Cecil C. Bridges. "Hierarchical Cluster Analysis." In: *Psychological Reports* 18.3 (June 1966), pp. 851–854.
- [118] Zoltán Király and Péter Kovács. "Efficient Implementations of Minimum-Cost Flow Algorithms." In: *Acta Univ. Sapientiae, Inform.* 4.1 (2012), pp. 67–118.
- [119] F. Pedregosa et al. "Scikit-Learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [120] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function Using NetworkX." In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gael Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.
- [121] Eva Lange, Ralf Tautenhahn, Steffen Neumann, and Clemens Gröpl. "Critical Assessment of Alignment Procedures for LC-MS Proteomics and Metabolomics Measurements." In: *BMC Bioinformatics* 9.1 (Sept. 2008).

- [122] Matthew Bellew, Marc Coram, Matthew Fitzgibbon, Mark Igra, Tim Randolph, Pei Wang, Damon May, Jimmy Eng, Ruihua Fang, ChenWei Lin, et al. "A Suite of Algorithms for the Comprehensive Analysis of Complex Protein Mixtures Using High-Resolution LC-MS." In: *Bioinformatics (Oxford, England)* 22.15 (2006), pp. 1902–1909.
- [123] Mikko Katajamaa and Matej Orešič. "Processing Methods for Differential Analysis of LC/MS Profile Data." In: *BMC bioinformatics* 6.1 (2005), pp. 1–12.
- [124] Xiao-jun Li, C Yi Eugene, Christopher J Kemp, Hui Zhang, and Ruedi Aebersold. "A Software Suite for the Generation and Comparison of Peptide Arrays from Sets of Data Collected by Liquid Chromatography-Mass Spectrometry\* S." In: *Molecular & Cellular Proteomics* 4.9 (2005), pp. 1328–1340.
- [125] Xiang Zhang, John M Asara, Jiri Adamec, Mourad Ouzzani, and Ahmed K Elmagarmid. "Data Pre-Processing in Liquid Chromatography–Mass Spectrometry-Based Proteomics." In: *Bioinformatics (Oxford, England)* 21.21 (2005), pp. 4054–4059.
- [126] Colin A. Smith, Elizabeth J. Want, Grace OMaille, Ruben Abagyan, and Gary Siuzdak. "XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification." In: *Analytical Chemistry* 78.3 (Jan. 2006), pp. 779–787.
- [127] Glen Lester Sequiera, Niketa Sareen, Vikram Sharma, Arun Surendran, Ejlal Abu-El-Rub, Amir Ravandi, and Sanjiv Dhingra. "High Throughput Screening Reveals No Significant Changes in Protein Synthesis, Processing, and Degradation Machinery during Passaging of Mesenchymal Stem Cells." In: *Canadian Journal of Physiology and Pharmacology* 97.6 (June 2019), pp. 536–543.
- [128] Juan A Vizcaíno et al. "ProteomeXchange Provides Globally Coordinated Proteomics Data Submission and Dissemination." In: *Nature Biotechnology* 32.3 (Mar. 2014), pp. 223–226.
- [129] Yasset Perez-Riverol et al. "The PRIDE Database and Related Tools and Resources in 2019: Improving Support for Quantification Data." In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D442–D450.
- [130] Michele Magrane and UniProt Consortium. "UniProt Knowledgebase: A Hub of Integrated Protein Data." In: *Database* 2011 (Jan. 2011), bar009.
- [131] Jimmy K. Eng, Tahmina A. Jahan, and Michael R. Hoopmann. "Comet: An Open-Source MS/MS Sequence Database Search Tool." In: *PROTEOMICS* 13.1 (Dec. 2012), pp. 22–24.



- [132] Jimmy K. Eng, Michael R. Hoopmann, Tahmina A. Jahan, Jarrett D. Egertson, William S. Noble, and Michael J. MacCoss. "A Deeper Look into Comet—Implementation and Features." In: *Journal of the American Society for Mass Spectrometry* 26.11 (June 2015), pp. 1865–1874.
- [133] Fatema Tuz Zohora, M. Ziaur Rahman, Ngoc Hieu Tran, Lei Xin, Baozhen Shan, and Ming Li. "DeepIso: A Deep Learning Model for Peptide Feature Detection from LC-MS Map." In: *Scientific Reports* 9.1 (Nov. 2019).
- [134] R. Smith, D. Ventura, and J. T. Prince. "Novel Algorithms and the Benefits of Comparative Validation." In: *Bioinformatics (Oxford, England)* 29.12 (Apr. 2013), pp. 1583–1585.
- [135] Inc. Daylight Chemical Information Systems. *SMARTS - A Language for Describing Molecular Patterns*.
- [136] *RDKit: Open-source Cheminformatics*. <https://rdkit.org/>.
- [137] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. "A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.10 (Oct. 2004), pp. 1367–1372.
- [138] Hans-Christian Ehrlich and Matthias Rarey. "Systematic Benchmark of Substructure Search in Molecular Graphs - From Ullmann to VF2." In: *Journal of Cheminformatics* 4.1 (July 2012), p. 13.
- [139] David Sherrington and Scott Kirkpatrick. "Solvable Model of a Spin-Glass." In: *Physical Review Letters* 35.26 (Dec. 1975), pp. 1792–1796.
- [140] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6.6* (Nov. 1984), pp. 721–741.
- [141] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288.
- [142] Neal Parikh. "Proximal Algorithms." In: *Foundations and Trends® in Optimization* 1.3 (2014), pp. 127–239.
- [143] Yves F. Atchadé, Gersende Fort, and Eric Moulines. "On Perturbed Proximal Gradient Algorithms." In: *Journal of Machine Learning Research* 18.10 (2017), pp. 1–33.
- [144] Blazej Miasojedow and Wojciech Rejchel. "Sparse Estimation in Ising Model via Penalized Monte Carlo Methods." In: *Journal of Machine Learning Research* 19.75 (2018), pp. 1–26.
- [145] E. Ising. "Beitrag Zur Theorie Des Ferromagnetismus." In: *Zeitschrift fur Physik* 31 (Feb. 1925), pp. 253–258.

- [146] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines." In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092.
- [147] W. K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109.
- [148] Norman L. Johnson, Adrienne W. Kemp, and Samuel Kotz. *Univariate Discrete Distributions*. John Wiley & Sons, Inc., Jan. 2005.
- [149] Nocedal Jorge and Wright S. *Numerical Optimization*. Springer New York, 2006.
- [150] Błażej Miasojedow, Eric Moulines, and Matti Vihola. "An Adaptive Parallel Tempering Algorithm." In: *Journal of Computational and Graphical Statistics* 22.3 (July 2013), pp. 649–664.
- [151] Matti Vihola. "Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate." In: *Statistics and Computing* 22.5 (Sept. 2012), pp. 997–1008.
- [152] Teague Sterling and John J. Irwin. "ZINC 15 – Ligand Discovery for Everyone." In: *Journal of Chemical Information and Modeling* 55.11 (Nov. 2015), pp. 2324–2337.
- [153] B. Efron. "Bootstrap Methods: Another Look at the Jackknife." In: *The Annals of Statistics* 7.1 (Jan. 1979), pp. 1–26.
- [154] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine." In: *The Annals of Statistics* 29.5 (Oct. 2001), pp. 1189–1232.
- [155] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 3149–3157.
- [156] David Rogers and Mathew Hahn. "Extended-Connectivity Fingerprints." In: *Journal of Chemical Information and Modeling* 50.5 (May 2010), pp. 742–754.
- [157] Moises Hassan, Robert D. Brown, Shikha Varma-O'Brien, and David Rogers. "Cheminformatics Analysis and Learning in a Data Pipelining Environment." In: *Molecular Diversity* 10.3 (Aug. 2006), pp. 283–299.
- [158] Sunghwan Kim et al. "PubChem in 2021: New Data Content and Improved Web Interfaces." In: *Nucleic Acids Research* 49.D1 (Jan. 2021), pp. D1388–D1395.

- [159] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity." In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133.
- [160] Alexander J. Lawson, Jürgen Swienty-Busch, Thibault Géoui, and David Evans. "The Making of Reaxys—Towards Unobstructed Access to Relevant Chemistry Information." In: *The Future of the History of Chemical Information*. Vol. 1164. ACS Symposium Series. Washington, DC: American Chemical Society, Jan. 2014. Chap. 8, pp. 127–148.
- [161] C. Spearman. "The Proof and Measurement of Association between Two Things." In: *The American Journal of Psychology* 15.1 (1904), pp. 72–101.
- [162] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. "Support Vector Method for Novelty Detection." In: *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, 1999.
- [163] Miyuki Sakai, Kazuki Nagayasu, Norihiro Shibui, Chihiro Andoh, Kaito Takayama, Hisashi Shirakawa, and Shuji Kaneko. "Prediction of Pharmacological Activities from Chemical Structures with Graph Convolutional Neural Networks." In: *Scientific Reports* 11.1 (Jan. 2021), p. 525.
- [164] Frederik Sandfort, Felix Strieth-Kalthoff, Marius Kühnemund, Christian Beecks, and Frank Glorius. "A Structure-Based Platform for Predicting Chemical Reactivity." In: *Chem* 6.6 (June 2020), pp. 1379–1390.
- [165] Samarjeet Prasad and Bernard R. Brooks. "A Deep Learning Approach for the Blind logP Prediction in SAMPL6 Challenge." In: *Journal of Computer-Aided Molecular Design* 34.5 (May 2020), pp. 535–542.
- [166] Milan Voršilák and Daniel Svozil. "Nonpher: Computational Method for Design of Hard-to-Synthesize Structures." In: *Journal of Cheminformatics* 9.1 (Mar. 2017), p. 20.
- [167] Anna Gaulton et al. "The ChEMBL Database in 2017." In: *Nucleic Acids Research* 45.D1 (Jan. 2017), pp. D945–D954.
- [168] Levente Kocsis and Csaba Szepesvári. "Bandit Based Monte-Carlo Planning." In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 282–293.
- [169] Rémi Coulom. "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search." In: *Computers and Games*. Ed. by H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, pp. 72–83.

- [170] Guillaume M. J-B. Chaslot, Mark H. M. Winands, H. Jaap Van Den Herik, Jos W. H. M. Uiterwijk, and Bruno Bouzy. "Progressive Strategies for Monte-Carlo Tree Search." In: *New Mathematics and Natural Computation* 04.03 (Nov. 2008), pp. 343–357.
- [171] G. Skoraczyński, P. Dittwald, B. Miasojedow, S. Szymkuć, E. P. Gajewska, B. A. Grzybowski, and A. Gambin. "Predicting the Outcomes of Organic Reactions via Machine Learning: Are Current Descriptors Sufficient?" In: *Scientific Reports* 7.1 (June 2017), p. 3582.
- [172] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. "Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models." In: *Journal of Cheminformatics* 13.1 (Feb. 2021), p. 12.
- [173] Student. "The Probable Error of a Mean." In: *Biometrika* 6.1 (1908), pp. 1–25.

#### COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both  $\text{\LaTeX}$  and  $\text{\LyX}$ :

<https://bitbucket.org/amiede/classicthesis/>

Happy users, including the author of this thesis, of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

*Final Version* as of January 4, 2023 (version: 1.0).