Department of Computer Science
Series of Publications A
Report A-2012-9

Computational methods for small molecules

Markus Heinonen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII (University Main Building, Unionkatu 34) on December 17th, 2012, at twelve o'clock noon.

University of Helsinki
Finland

Supervisor
Juho Rousu, Aalto University, Finland

Pre-examiners
Tapio Pahikkala, University of Turku, Finland
Steffen Neumann, Leibniz Institute of Plant Biochemistry, Germany

Opponent
Yves Moreau, KU Leuven, Belgium

Custos
Esko Ukkonen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi
URL: http://www.cs.Helsinki.fi/
Telephone: +358 9 1911, telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-8451-5 (paperback)
ISBN 978-952-10-8452-2 (PDF)
Computing Reviews (1998) Classification: G.2.1, G.2.2, I.6.5, I.5, J.3
Helsinki 2012
Helsinki University Print

Computational methods for small molecules

Markus Heinonen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
markus.heinonen@gmail.com
http://www.cs.helsinki.fi/u/mqheinon/

PhD Thesis, Series of Publications A, Report A-2012-9
Helsinki, December 2012, 130 + 68 pages
ISSN 1238-8645
ISBN 978-952-10-8451-5 (paperback)
ISBN 978-952-10-8452-2 (PDF)

Abstract

Metabolism is the system of chemical reactions sustaining life in the cells of living organisms. It is responsible for cellular processes that break down nutrients for energy and produce building blocks for necessary molecules. The study of metabolism is vital to many disciplines in medicine and pharmacy. Chemical reactions operate on small molecules called metabolites, which form the core of metabolism. In this thesis we propose efficient computational methods for small molecules in metabolic applications. In this thesis we discuss four distinctive studies covering two major themes: the atom-level description of biochemical reactions, and analysis of tandem mass spectrometric measurements of metabolites.

In the first part we study atom-level descriptions of organic reactions. We begin by proposing an optimal algorithm for determining the atom-to-atom correspondences between the reactant and product metabolites of organic reactions. In addition, we introduce a graph edit distance based cost as the mathematical formalism to determine optimality of atom mappings. We continue by proposing a compact single-graph representation of reactions using the atom mappings. We investigate the utility of the new representation in a reaction function classification task, where a descriptive category of the reaction's function is predicted. To facilitate the prediction, we introduce the first feasible path-based graph kernel, which describes the reactions as path sequences to high classification accuracy.

In the second part we turn our focus on analysing tandem mass spectrometric measurements of metabolites. In a tandem mass spectrometer, an input molecule structure is fragmented into substructures or fragments, whose masses are observed. We begin by studying the fragment identification problem. A combinatorial algorithm is presented to enumerate candidate substructures based on the given masses. We also demonstrate the usefulness of utilising approximated bond energies as a cost function to rank the candidate structures according to their chemical feasibility. We propose fragmentation tree models to describe the dependencies between fragments for higher identification accuracy.

We continue by studying a closely related problem where an unknown metabolite is elucidated based on its tandem mass spectrometric fragment signals. This metabolite identification task is an important problem in metabolomics, underpinning the subsequent modelling and analysis efforts. We propose an automatic machine learning framework to predict a set of structural properties of the unknown metabolite. The properties are turned into candidate structures by a novel statistical model. We introduce the first mass spectral kernels and explore three feature classes to facilitate the prediction. The kernels introduce support for high-accuracy mass spectrometric measurements for enhanced predictive accuracy.

Computing Reviews (1998) Categories and Subject Descriptors:
G.2.1 Combinatorics: Combinatorial algorithms
G.2.2 Graph theory: Graph algorithms
I.6.5 Model development
I.5 Pattern recognition
J.3 Life and medical sciences

General Terms:
Algorithms, machine learning, bioinformatics, computational biology, chemoinformatics

Additional Key Words and Phrases:
Metabolism, mass spectrometry, combinatorial algorithms, graph algorithms, kernel methods, graph kernels, computational complexity

Acknowledgements

I am grateful to Juho Rousu for supervising my thesis work and for the continuous support during my PhD studies in its every aspect.

I thank Esko Ukkonen for his guidance and supervision, especially during the years as a MSc student in the SYSFYS project. The "can-do" attitude of Esko gave a valuable orientation into science.

I thank the pre-examiners Tapio Pahikkala and Steffen Neumann for their effort and insightful comments that helped to improve this dissertation.

The contributions by the co-authors have been invaluable to the works presented in this thesis. I am especially indebted to Ari Rantanen for his guidance during the first years. The continuous cultivation of, sometimes far-flung, research ideas with Ari and Taneli Mielikäinen was remarkably fun and fostered the "outside the box" mentality that I have tried to keep in mind during my PhD studies.

I thank Juha Kokkonen, Jari Kiuru and Raimo Ketola for the invaluable experimental support during the fragment papers, without which the papers would not have been possible. The work by Sampsa Lappalainen was instrumental during the reaction mapping project. The path kernel paper would not have been possible without the helpful cooperation and the astonishing algorithms of Niko Välimäki and Veli Mäkinen. I am grateful to Huibin Shen for the implementation and experiments on the last paper, and to Nicola Zamboni for collaboration.

I thank Esa Pitkänen and Pasi Rastas whose feedback on various research ideas were instrumental for many ideas that ended up in the papers.

The excellent atmosphere at the Department of Computer Science has been due to the interesting discussions and interactions with Jussi Kollin, Esa Junttila, Mikko Arvas, Leena Salmela, Jarkko Toivonen, Janne Korhonen, Kimmo Palin, Pekka Parviainen and Niina Haiminen. The BioBeer tradition with Janne Ravantti, Päivi Onkamo, Teemu Kivioja, Leo Lahti, Ilkka Huopaniemi, Mikko Koivisto, Kerttu Majander, Jaana Oikkonen and Paula Jouhten has certainly broadened my outlook on science and beyond,

but has most importantly been really fun. I thank the members of the group Honguy Su, Katja Astikainen, Eva König, Arto Åkerlund, Pekko Parikka and Antti Tani for feedback and support.

I am grateful to the exceptional organisational and IT support by Combi, Algodan, FDK, HIIT, the department of Computer Science and Aalto that have allowed me an almost undisturbed focus on the research. I thank Ella Bingham for the uncomplicated support by the PhD School FICS. I am grateful to Combi and FICS for providing the funding for my PhD work.

I would like to thank my parents Suvi and Pekka, and my siblings Linda, Kenny and Tuomas for the support, and for inspiring and always encouraging me on my studies. Finally my heartfelt thanks go to Suvi, who supported me through the journey.

Contents

Original Publications of the Thesis

This thesis is based on the following peer-reviewed publications, which are referred to as Papers **I**–**V** in the text. Papers **I**–**V** are reproduced at the end of the thesis.

I Markus Heinonen, Sampsa Lappalainen, Taneli Mielikäinen and Juho Rousu
Computing atom mappings for biochemical reactions without subgraph isomorphism
Journal of Computational Biology 18(1):43–58, 2011.

II Markus Heinonen, Niko Välimäki, Veli Mäkinen and Juho Rousu
Efficient path kernels for reaction function prediction
In *Proceedings of 3rd International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS)*, pages 202–207, 2012.

III Markus Heinonen, Ari Rantanen, Taneli Mielikäinen, Esa Pitkänen, Juha Kokkonen and Juho Rousu
***Ab initio* prediction of molecular fragments from tandem mass spectrometry data**
In *Proceedings of the German Conference of Bioinformatics (GCB)*, (Tübingen, Germany), *Lecture Notes in Informatics (LNI)*, Vol P-83 pages 40–53, 2006.

IV Markus Heinonen, Ari Rantanen, Taneli Mielikäinen, Juha Kokkonen, Jari Kiuru, Raimo Ketola and Juho Rousu
FiD: a software for *ab initio* structural identification of product ions from tandem mass spectrometric data
Rapid Communications in Mass Spectrometry 22:3043–3052, 2008.

V Markus Heinonen, Huibin Shen, Nicola Zamboni and Juho Rousu
Metabolite identification and fingerprint prediction via machine learning

Part I

Chapter 1

Introduction

The aim of this thesis is to propose efficient computational methods to various biological and chemical problems involving small molecules.

This thesis consists of the present introductory Part I and five original publications reprinted at Part II at the end of the thesis. The purpose of this introductory part is to provide motivation, background, and a literature survey of the problems discussed in the original publications in a unified notation. We do not describe the proposed computational methods or experimental results in this introductory part, instead we refer the reader to the original publications in the Part II.

This thesis presents studies and results on four distinct bioinformatics and chemoinformatics applications presented in five original Papers. The research studies are:

- Computing atom mappings for biological reactions (Paper **I**)
- Computing a path kernel for reaction function prediction (Paper **II**)
- Computing fragment identifications of tandem mass spectrometric data (Papers **III** and **IV**)
- Computing metabolite identification through kernel methods from tandem mass spectrometric data (Paper **V**)

In each case a biological or chemical problem is formalised and computer scientific algorithms and methods are presented to solve the problem. The goal in each study has been to provide efficient state-of-the-art algorithms for domain problems.

We begin the introductory Part by collecting the common biological, chemical and mathematical background, as well as introducing machine learning and kernel methods in the Chapter 2.

The following three main Chapters 3 to 5 introduce the topics of the original Papers. Each main chapter introduces the respective biological or chemical problem, and gives a mathematical problem definition. We introduce necessary biological concepts and review the computer scientific literature on the problem domain with a summary of the state-of-the-art methods and algorithms.

In Chapter 3 we discuss the problem of reaction mapping. In a chemical reaction a set of reactant molecules are structurally modified and become product molecules. In reaction mapping the atom-to-atom correspondences between reactants and products are determined. The mapping problem is an instance of inexact graph matching, with a large – yet, in the chemical community unexploited – literature.

In Chapter 4 we move our focus to constructing representations of molecules and reactions by feature vectors that collect counts of their graph-theoretic parts. Widely used kernel methods utilise the representations for various prediction tasks. As an example we discuss a reaction function prediction task.

In Chapter 5 we discuss analysis of tandem mass spectrometric measurements of small molecules. A tandem mass spectrometer measures the masses of structural fragments of a molecule. Two main questions are explored: how to identify an unknown molecule based on its tandem mass spectrometric measurement, and how to identify the structural fragments, assuming that the molecule itself is known. We introduce both algorithmic and machine learning approaches for these problems.

We conclude with Chapter 6.

1.1 Original contributions

The contributions in this thesis are given in the original publications **I-V**. We summarise the main contributions below in the order of the original Papers.

- We introduce graph edit distance with support for arbitrary costs as a flexible cost function framework for reaction mapping problem. An A* algorithm for computing the optimal atom mappings of chemical reactions is introduced with several novel heuristics that exploit molecule-specific properties. We propose to enhance A* by introducing several atom-level chemical context descriptors, generated using a Message Passing framework. We compute optimal atom mappings for over 5,800 common organic reactions of the KEGG database. We

introduce the concept of using atom mappings for compact reaction graph representation.

- We introduce the first feasible path-based graph kernel, using a compressed path-index data structure to store the paths of a dataset of graphs efficiently. We introduce reaction graphs as a novel reaction representation for reaction classification. The path kernel is computed on over 17,000 reaction graphs to predict reaction function classification with MMCRF algorithm with state-of-the-art prediction accuracy. The path-based graph kernel is shown to achieve dramatically higher performance than the commonly used walk-based graph kernels.

- We propose an exhaustive subgraph search algorithm for enumeration of candidate fragments of tandem mass spectrometric product ion peaks. We introduce a bond energy based cost function to rank these candidates according to their chemical feasibility. A developed windows application implements the methods.

- We introduce the novel concept of fragmentation trees, and propose three fragmentation tree models of increasing complexity as a more accurate model for the fragmentation process. Mixed integer linear programming solutions are presented for these models to estimate the fragments of a tandem mass spectrum as a whole. Experiments show improved accuracy with both fragmentation trees and bond energies compared to state-of-the-art simulation-based fragment identification methods.

- An automatic metabolite identification framework based on pattern recognition for tandem mass spectrometry is introduced. The framework consists of two parts: we first predict binary properties of the unknown metabolite based on its mass spectral signals. We introduce a statistical model to turn these properties automatically into a ranked list of candidate structures from large molecular repositories. We introduce first non-trivial mass spectral kernels. We explore two families of kernels on three classes of mass spectral features extracted from the spectra of unknown compounds. The kernels introduce support for non-integral mass measurements.

The contribution of the author to all of the original publications was substantial.

In paper **I**, the author developed the algorithms and ran the experiments. The author and Sampsa Lappalainen implemented the algorithms.

The author developed and implemented the atom descriptors. The paper was co-written by the author and Juho Rousu.

In Paper **II**, the author co-developed the kernels with Juho Rousu. The implementation of the kernels and graphs was by the author, while Niko Välimäki wrote the path index data structure. The author ran the experiments. The paper was co-written by all authors equally.

In Paper **III**, the author implemented the methods and ran the experiments. The initial ideas were conceived by Ari Rantanen and Taneli Mielikäinen, while the author developed the enumeration algorithm and conceived the different MILP models.

In Paper **IV**, the author implemented the software, and ran the experiments with Ari Rantanen.

In Paper **V**, the author conceived the idea of representing spectra as densities and developed and derived the probability product kernels, along with the feature representation. The author developed the Poisson-Binomial model and supervised the implementation. The paper was co-written by the author and Juho Rousu.

Chapter 2

Background

In this chapter we introduce the essential chemical, biological and mathematical background necessary for the topics of the thesis. We review basic concepts of chemistry, metabolism, mass spectrometry, graph theory and kernel methods.

2.1 Chemistry

A compound is a chemical substance consisting of *atoms* connected by chemical *bonds* to form a structure. The atom is a basic unit of matter that consists of protons, neutrons and electrons. The number of protons of an atom determines its *chemical element*, e.g. carbon, oxygen or iron. The *elemental composition* or *elemental formula* of a compound denotes the counts and types of its atoms. For instance, a glucose has an elemental composition of $\mathtt{C}_6\mathtt{H}_{12}\mathtt{O}_6$.

The number of neutrons of an atom determines its *isotope*. For instance, a 12-carbon is the most common carbon isotope with 12 neutrons at approximately 98% abundance in organic matter. The next common isotope is the 13-carbon with an extra neutron with 1.007% abundance. An atom refers to a distribution of its isotopes, which usually have equivalent properties, but different masses.

The mass of atoms and compounds is measured in *atomic mass units* (u). The mass of 12-carbon is by convention exactly 12 u and the mass of 13-carbon is 13.004 u. The mass of a compound is then a distribution over the isotopic variants of its atoms. A *standard atomic weight* is the expected mass of atoms and compounds. However, in practise analysis is simplified by omitting the isotopic variants. In this thesis we use *atomic mass*, which is defined as the mass based on the most common isotope. A carbon has a

standard atomic weight of 12.011 u due to isotopes, while its atomic mass is exactly 12 u. As a hydrogen has a mass of 1.008 u and an oxygen has a mass of 15.994 u, the mass of glucose is then 180.063.

A *molecule* is an electrically neutral compound. In contrast, an electrically charged compound is called an *ion*, which has either a positive or negative charge due to an imbalance between positive protons and negative electrons.

2.2 Metabolism

Metabolism is the operation of chemical reactions sustaining life in the cells of living organisms [136]. The constantly ongoing chemical processes within cells allow the cells to break down nutrients for energy, and produce molecules as building blocks for the cell to function, grow and reproduce.

The main concepts of metabolism are metabolites and reactions. A *metabolite* is usually a small molecule of less than 1000 u participating in metabolic reactions. A metabolic *reaction* transforms a set of *reactant* metabolites into a set of *product* metabolites, often catalysed by an *enzyme*. The enzymes are proteins produced according to the genome of the organism through translation, which ultimately regulates the metabolism. Metabolic reactions form metabolic pathways by sharing metabolites in reactions as both reactants and products. For instance, a glucose metabolite is first transformed into a glucose-6-phosphate, which is then transformed into a fructose-6-phosphate. The subsequent reactions transform the compound further into pyruvate, which is the starting point of the TCA cycle, responsible for generation of energy.

The metabolism of common organisms are well-known [99]. Both metabolites and reactions are annotated in databases, such as KEGG [134] and BioCyc [45]. The methods of this thesis are independent of the data source. We use the KEGG database exclusively in this thesis.

2.3 Mass spectrometry (MS)

Mass spectrometry is a key analytical measurement technology for quantification and qualification of compounds [184]. It is along with Nuclear Magnetic Resonance (NMR) the main measurement technology available for chemists working with small molecules. Mass spectrometry is able to measure the chemical composition of a sample; for instance a cell culture. Cells can contain thousands of unique metabolites, each in various concentrations. Wide range of mass spectrometers exist that are suitable for

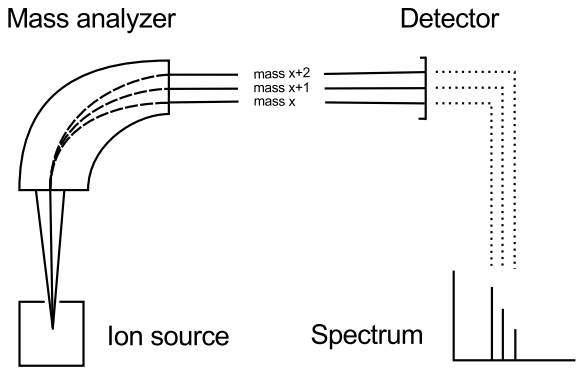


Figure 2.1: A schematics of a mass spectrometer. The (organic) sample is first ionised, and then individual ions are separated according to their mass-to-charge ratio in the mass analyser. The detector measures the ratios and produces a mass spectrum. Various technologies exist for each of the components of a mass spectrometer.

high-throughput and high-accuracy experiments.

A mass spectrometer measures the mass-to-charge ratio of compounds in a biological sample. The measurement results in a *mass spectrum*, with *peaks* indicating the mass-to-charge ratio of pools of molecular species. The height of the peak is denoted as *intensity*, which represents the size of the pool. Often a simplifying assumption of unit charges is made, either by using preprocessing to normalize charges, or by using a mass spectrometer which favours singly-charged species. Using this assumption, the mass spectrum gives the masses of the compounds directly. With suitable accuracy, the mass is sometimes enough to deduce the elemental formula of the analysed compound [140].

The key components in any mass spectrometer are (i) the *ion source*, which ionises the sample molecules (i.e. adds charge), (ii) the separation of the ions by electromagnetic fields in a *mass analyser*, and (iii) the measurement of the mass-to-charge ratio at the *detector*. The general idea of MS is to subject the trajectories of ions to electromagnetic fields. A compound resists these fields in relation to its mass and charge (See Figure 2.1).

Additionally, MS is often coupled with a gas chromatography (GC) or liquid chromatography (LC). Chromatography separates the molecules in the sample physically by forcing them through a resistant medium to reduce the complexity of the spectra. The compounds arrive through the medium at different times according to their mass and shape.

Mass spectrometers are only able to separate charged species in the mass analyser. Hence, the sample is ionised. However, ionisation places excess internal energy in the compounds, which renders them inherently unstable. The molecules seek to lose the internal energy and charge by structural rearrangements, leading sometimes to *in-source fragmentation*. The most common fragmentation event is bond cleavage, where a removed bond can lead to formation of two independent subcompounds called *fragments*. In the case of singly-charged species, only one of the fragments retains the charge. The other fragment is denoted the *neutral loss*, which is not visible in the spectrum. Fragments are sometimes denoted *product ions*.

The fragmentation is usually an undesired effect: some peaks correspond now to fragments instead of intact molecules, without any direct indication of the status of the peak. Fragmentation is prevented with two methods: by using "soft" ionisation, which does not place too much internal energy to the compounds to fragment, and by measuring the ions quickly such that the ions do not have enough time to undergo fragmentation.

Most widely used ionisation methods are soft. These include chemical ionisation (CI), electrospray ionisation (ESI) and laser-based method MALDI, which is especially suitable for macromolecules such as DNA or proteins. A common electron ionisation (EI) is a hard ionisation method, which induces in-source fragmentation to provide fragment peaks, and hence, more structural information.

Widely used mass analysers include Quadrupole (QqQ), Ion traps (IT), Time-of-Flight (TOF) instruments and Fourier Transform Ion Cyclotrons (FTICR). The Quadrupole uses radio frequencies to only pass to the detector ions of a specific mass in turn. Ion traps literally trap the ions and eject them based on masses. TOF instruments use electric fields to accelerate ions through a potential. The mass is measured on the flight time to detector. Finally, Ion cyclotron resonance methods trap the ions and produce a continuous signal of the ions, which is deconvoluted with Fourier Transform.

The mass analyser defines two important properties of MS: the mass range and mass accuracy. The mass range of most methods is up to 3000 u, which is suitable for metabolomics. The mass accuracy is the error in the mass measurement in parts-per-millions of the true mass. Ion traps and Quadrupoles have a low mass accuracy in the range of 100 ppm, while TOF and FTICR offer high accuracy up to 0.5 ppm. In a metabolite of mass 500 u, these translate into absolute errors ε up to 0.025 and 0.0005, respectively. The lowest accuracy is the nominal mass accuracy, where the masses are integral. Molecular masses of metabolites are naturally centred

around integral values and hence nominal mass data is usually rounded to closest integer without a significant loss of information.

Often the peaks are annotated by matching the peak masses against masses of known compounds. We discuss computational identification of compounds from the mass spectral signals in Chapter 5.

2.3.1 Tandem mass spectrometry (MS/MS)

Tandem mass spectrometry is a special type of MS where multiple MS devices are coupled together. Tandem mass spectrometry is denoted MS/MS or MS^2 with conventional MS denoted then as MS^1. Chaining of more than two mass spectrometers results in MS^n, which can be done in-space or in-time. In-space chaining connects several mass spectrometers physically (e.g. QqQ or QqTOF), while in-time chaining can perform several MS rounds in a single machine (e.g. FTICR). The main application of MS/MS is structural identification of unknown compounds, which is discussed in Chapter 5.

In MS/MS, we explicitly induce fragmentation. The first round of MS is used to separate a specific ion based on its mass-to-charge ratio, and the second high-energy MS is used to fragment that ion extensively (See Figure 2.2). The ion chosen for fragmentation from MS^1 is denoted *parent ion* or *precursor ion*. The resulting tandem mass spectrum contains peaks that correspond only to fragments of the precursor ion, and possibly a peak of non-fragmented precursor. Each peak indicates the mass of a fragment. The fragmentation process depends on the ion structure, applied energy and other operational parameters [184]. Thus, with standardised mode of measurement, the resulting tandem mass spectrum is a unique pattern that depends primarily on the structure of the precursor ion.

The fragmentation is induced by either using high-energy ionisation or collision-induced-dissociation (CID). In CID the ions are collided with inert gases, which causes fragmentation. The large amount of energy generated by CID often results in fragments that still contain large amounts of energy. These fragments can fragment again by another set of bond cleavages, producing *secondary fragments*. Continuing this process, the fragmentation can be modelled as a fragmentation tree of successive fragmentation reactions with intermediate ions, some of which are detected before further fragmentation.

The fragmentation is dominated by cleavages of chemical bonds, which "cut" the ions into substructures, an other common event is a change of bond order. Additionally, rearrangement reactions are also possible, where a new bond is formed. Chemically this occurs when two nearby atoms join

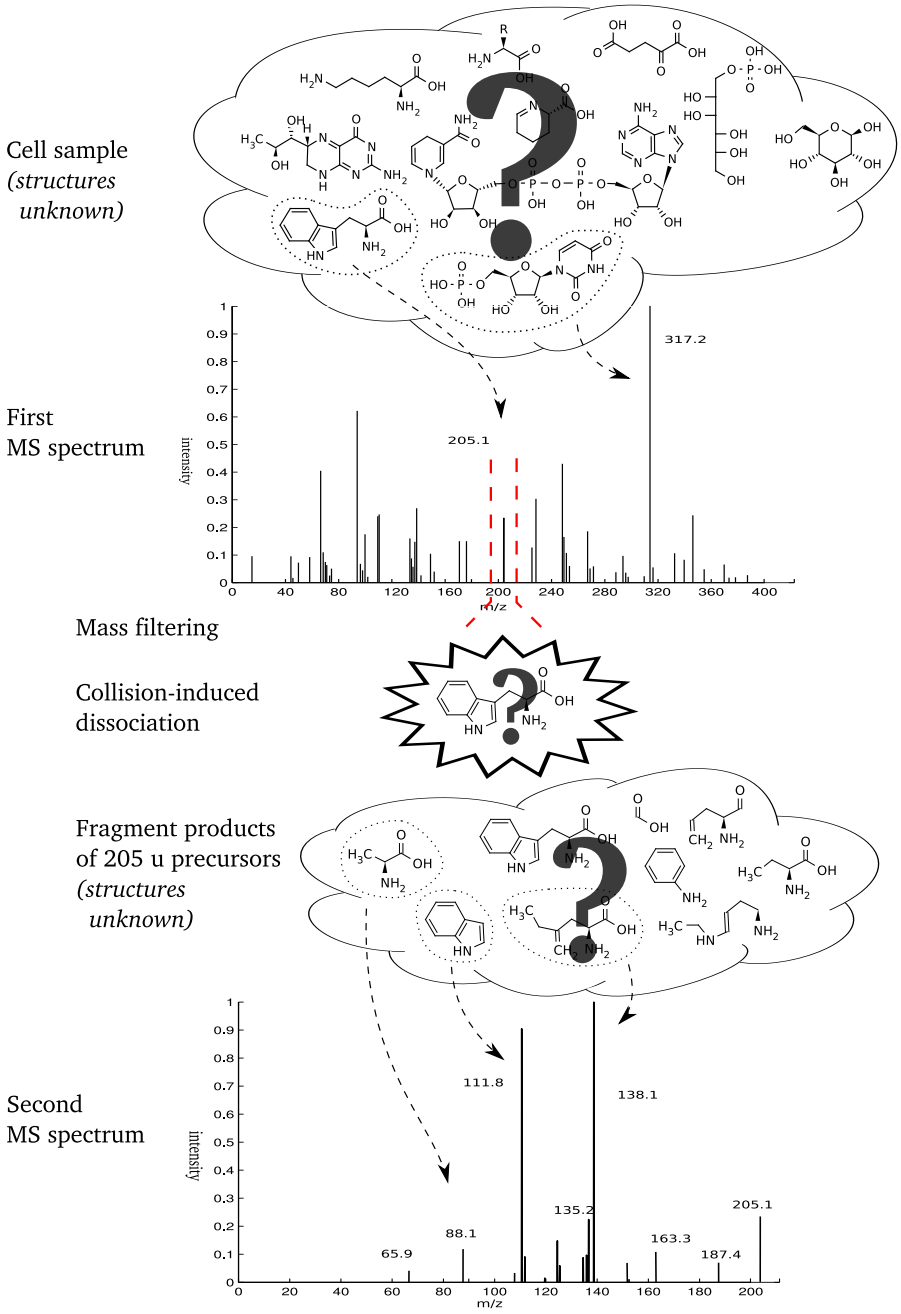


Figure 2.2: Schematic of tandem mass spectrometry. The first round of MS produces a 'wide' spectrum indicating the mass-to-charges of the compounds in the e.g. cell sample (top). In the second MS phase, a specific ion band is filtered for fragmentation, and the fragments are measured (bottom). Only the peak masses are measured by MS, the structures they correspond to need to be elucidated.

electrons for a new covalent bond. These reactions effectively generate new bonds in the fragmented structures.

The collision energy is often measured as electron volts eV, ranging from low-energy 10 eV collisions to high-energy 60 eV collisions. A low-energy CID tends to produce more heavier fragments with less fragmentation. High-energy CID produces smaller fragments [184]. For a more complete view of the possible fragments, spectra from different collision energies are often merged for subsequent analysis. In a RAMP spectra the collision energy is increased continuously during a single acquisition for a combined spectrum [194].

2.4 Graph theory

Graphs are powerful and versatile data structures, which can represent relational information about the modelled objects. In graph representation, objects are represented as vertices, and relations between objects with edges. In metabolism, both molecules and reactions are often represented as graphs. In molecules, the individual atoms are represented with vertices, and bonds between atoms with edges. Reactions can be interpreted as *graph transformations* and consist of a set of reactant and product molecules as graphs, and relationships between the vertices across both sides of the reaction.

Throughout this thesis we concern ourselves with small molecules in various computational tasks. We model the molecules consistently as two-dimensional labelled undirected graphs.

Graph concepts. A graph $G = (V, E)$ is a tuple where a vertex set $V = \{v_1, \ldots, v_n\}$ is connected by edges $E = \{e_1, \ldots, e_m\}$. A graph is directed if an edge $(v, u) \in E$ is distinct from an edge (u, v). Otherwise the graph is undirected. A *vertex-labelled graph* is associated with a labelling function $l : V \to \Sigma$, while an edge-labelled graph is associated with a function $l : E \to \Sigma$. A *edge-weighted* graph is associated with a weight function $w : E \to \mathbb{R}$, while a vertex-weighted graph has a function $w : V \to \mathbb{R}$.

The order of a graph $|G| = |V| = n$ is the number of its vertices. A graph is *connected* if there exists a path between every pair of vertices. Otherwise, we call the graph *disconnected*. A function $N(v)$ gives the neighbours of a vertex v. A degree $deg(v) = |N(v)|$ is the number of neighbours of a vertex.

A *subgraph* of a graph $G = (V, E)$ is $H = (V', E')$, where $V' \subset V$ and $E' \subset E$. We denote subgraph by $H \subset G$. A subgraph is called an *(vertex)-induced subgraph* if for all pairs of vertices in the subgraph,

all edges from *parent graph* G are present. In contrast, in an *edge-induced subgraph* only those vertices are included, which are adjacent to the edge set E'. Informally, an induced subgraph is defined as a subset of vertices with all accompanying edges between them, while an edge-induced subgraph is defined as a subset of edges with all endpoint vertices attached to the edge set.

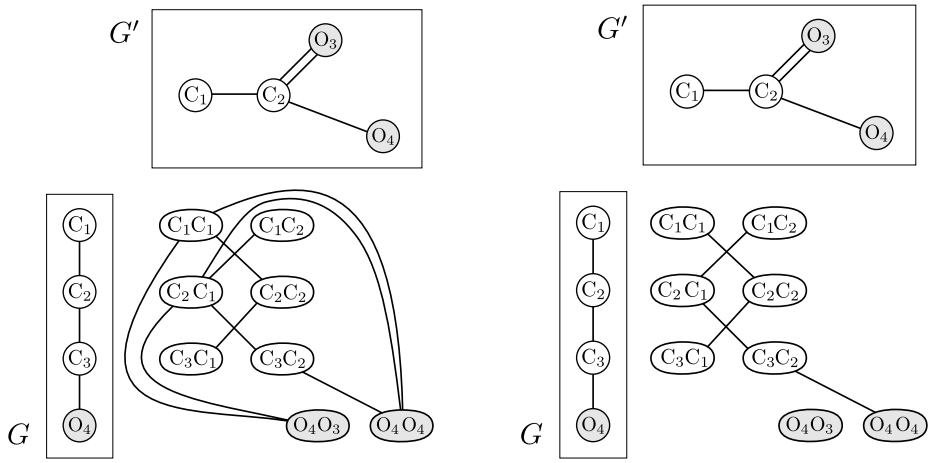
An *isomorphism* of graph G_1 and G_2 is a bijective mapping $f : V(G_1) \mapsto V(G_2)$ such that any two vertices $(v, u) \in V(G_1)^2$ are adjacent iff $(f(v), f(u)) \in E(G_2)$. For labelled graphs, we additionally require label matching $l(v) = l(f(v))$ for all $v \in V(G_1)$. Often only structural similarity is of interest, in which case the edge labels are not required to match through isomorphism. An isomorphism between two graphs is a structure preserving relation, which determines when two graphs are, in fact, the same graph up to a permutation of the vertices. An *automorphism* is an isomorphism between a graph and itself. An *subgraph isomorphism* is an isomorphism between a subgraph $H \subset G_1$ of graph G_1 and a graph G_2.

In a morphism problem we determine whether a certain morphism exists. For instance, in a graph isomorphism problem we determine whether two graphs are isomorphic. The graph isomorphism problem is NP-complete: no polynomial time-algorithms exist for the isomorphism problem [147]. However, no NP-completeness proof has been obtained either [147]. In practise efficient algorithms exist for both isomorphisms [54], and for special classes of graphs [170].

A *common subgraph* H between two graphs G_1 and G_2 is a subgraph of both. The common subgraph is sometimes denoted as a *common subgraph isomorphism* in the literature. A *maximal* common subgraph is a common subgraph which cannot be enlarged. A *maximum common subgraph* (MCS) is such a common subgraph that no larger common subgraph exists. A maximum common subgraph can be meaningfully defined for both regular and connected subgraphs, as well as for induced and edge-induced subgraphs. We refer these as CIMCS (connected induced MCS), CEMCS (connected edge-induced MCS), IMCS (induced MCS), and EMCS (edge-induced MCS). It is known that detection of a induced MCS (IMCS) is equivalent to determining the maximum clique in a *modular product graph* of G_1 and G_2 [164] (See Figure 2.3).

A modular product graph $G_\times$ of two graphs G and G' is defined as the label-matching subset of the cartesian product of the vertex sets $V(G)$ and $V(G')$

$$V_\times(G, G') = \{(v, v') \in V \times V' : l(v) = l(v')\},$$



(a) $G_\times(G, G')$ (b) $G_\otimes(G, G')$

Figure 2.3: (a) A modular product graph $G_\times$ of graphs G and G'. (b) A tensor product graph $G_\otimes$ of graphs G and G'. The oxygen atoms are highlighted with grey background.

and edges as

$$E_\times(G, G') = \{((v, v'), (u, u')) \in V_\times \times V_\times : l(v, v') = l(u, u'), \\ (v, u) \in E \wedge (v', u') \in E' \text{ or} \\ (v, u) \notin E \wedge (v', u') \notin E'\}.$$

In chemical literature it is sometimes called an association graph [106, 213]. A *clique* is a complete subgraph. In other words, a clique is a subgraph with an edge between all pairs of vertices of the subgraph. A maximum clique is the largest clique in a graph. A maximal clique is a clique that cannot be enlarged.

Graph sequences. Let a *walk* w in a graph G be a sequence of vertices $w = (v_1, \ldots, v_m)$, such that for all $i = 1, \ldots, m-1$ there exists and edge $(i, i+1) \in E$. The length of the walk is denoted as m. A walk w is *non-tottering* iff $v_{i-1} \neq v_{i+1}$ for all $i = 2, \ldots, m-1$ [175]. A non-tottering walk cannot backtrack to a node it just left.

A non-tottering walk can repeat a vertex given that there are at least two vertices in between. A *path* is a walk where no node repeats, i.e. $v_i \neq v_j$ for all $i, j \in 1, \ldots, m$. Contrary to walks, the number of paths is bounded as no path can be longer than $|V|$. A *shortest path* is a path

which coincides with the shortest sequence of steps from v_1 to v_m [29]. A *cycle* is a non-repeating walk, except for $v_1 = v_m$. A labelled sequence is $(l(v_1), \ldots, l(v_m))$.

There is a walk in the *tensor product graph* $G_\otimes$ of graphs G and G' that corresponds to a pair of label-matching walks in the original graphs G and G' [91] (See Figure 2.3). The tensor product graph[1] contains as vertices the label-matching subset of the cartesian product of vertices $V(G)$ and $V(G')$

$$V_\otimes(G, G') = \{(v, v') \in V \times V' : l(v) = l(v')\},$$

and edges as

$$E_\otimes(G, G') = \{((v, v'), (u, u')) \in V_\otimes \times V_\otimes : l(v, v') = l(u, u'), \\ (v, u) \in E \wedge (v', u') \in E'\}.$$

Molecules. A *chemical graph* is a labelled undirected graph $G = (V, E)$, where the vertices correspond to atoms and edges to chemical bonds between the atoms (See Figure 2.4). A labelling function $l : V \to \Sigma$ assigns an atom symbol (e.g. "Carbon", "Oxygen", etc.) to each vertex. We reuse a labelling function $l : E \to \Sigma$ also on the edges to determine the bond order ("single", "double" or "triple" bond). The degree of a vertex corresponds to the valence of its atom.

The *mass* of a molecule is the sum of atomic masses of its atom set

$$m(G) = \sum_{v \in V} m(v),$$

where $m(v)$ is the atomic mass of an atom type $l(v)$. Bonds have no mass.

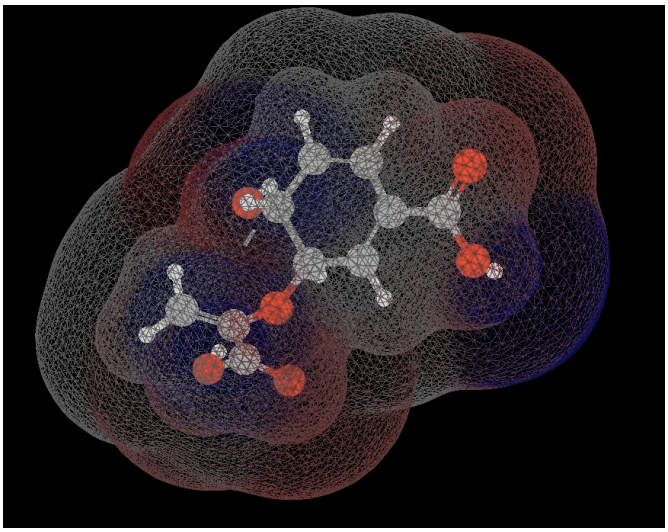
Reactions. A chemical reaction is a graph transformation, where a set of reactant molecules are transformed into a set of product molecules. Let a *reaction* $\rho = (R, P)$ be a tuple of reactant chemical graphs $R = (R_1, \ldots, R_r)$ and product chemical graphs $P = (P_1, \ldots, P_p)$ such that they correspond to a unidirectional reaction

$$R_1 + \cdots + R_r \longrightarrow P_1 + \cdots + P_p.$$

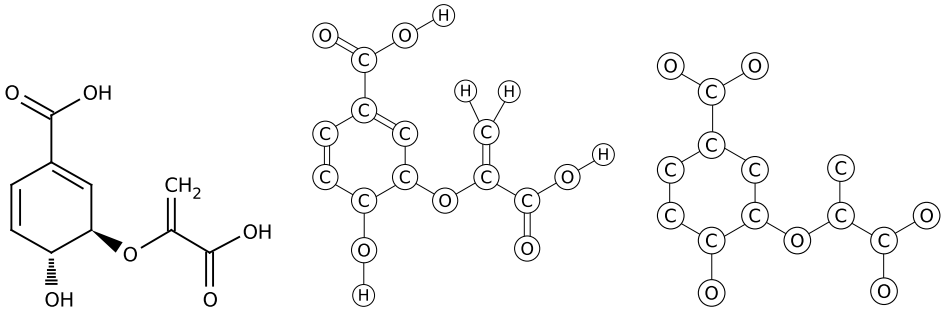
A bidirectional reaction is represented by two reactions with opposite directions.

A chemical reaction is *balanced* if for each label a, the total number of vertices with label a is equal in the reactants and products. That is, both sides have same atom sets and atoms are conserved in the reaction.

[1] Commonly denoted as direct product graph in the literature.



(a)



(b) (c) (d)

Figure 2.4: A chorismate ($C_{10}H_{10}O_6$) with four representations. (a) A three-dimensional representation with atom surfaces. (b) In a conventional two-dimensional chemical notation, carbon atoms are omitted on the backbone and bonds are labelled as "up" (solid triangle), "down" (dashed triangle), single bond or double bond. (c) A full graph representation. (d) A simplified graph representation with implicit Hydrogens and bond labels omitted to highlight the structure.

We denote the reactant (product) graph as a single, possibly disconnected graph $G_R = (V_R, E_R)$ (resp. $G_P = (V_P, E_P)$), where $V_R = \bigcup_i^r V(R_i)$ is the set of all vertices of reactant graphs, and $E_R = \bigcup_i^r E(R_i)$ is the corresponding edge set, respectively.

2.5 Molecular representations

The intuitive representation of molecules as two-dimensional labelled graphs is not the sole representation used in the chemical literature. In chemical databases there's a need for fast look-up of a query molecule. In a database of millions of compounds, performing graph isomorphism test against them would be too slow. The 1D string representations have been introduced for this problem with varying success. The string representation has become widely used alternative representation to graph representations. Line notations include proprietary standards SMILES [273, 274] and SMARTS [126], and the open standards InChI [245, 185, 83] and UCK [102].

SMILES, SMARTS and InChI standards represent the molecular structure as a string by graph serialisation in a canonised way. In SMILES atoms are represented with alphabetic letters, branching is indicated with parentheses and connectivity through placeholder numbers. The notation was extended with a backward-compatible SMARTS notation allowing wildcards. In recent years the international union of organic chemistry IUPAC has endorsed an open standard of InChI, which has been widely adopted.

UCK enumerates all paths up to a depth k in a chemical graph, lexicographically orders them and concatenates the paths [102]. It is not possible to expand UCK identifier back into graph form: it is only suitable for comparison of graphs.

The main aim of the line notation is that two structures are the same if the corresponding strings are equal, which coincides with the graph isomorphism problem. It is well known that SMILES and SMARTS do not actually hold this property in general case [192]. However, InChI claims that "If two InChI's are the same, then it is safe to assume that the compounds (structures) that they represent are the same" [245].

More complex representations include the 3D representation, which includes coordinates, and even more chemically accurate models including various chemical fields. We refer the reader to textbooks by Gasteiger and Engel [92], and Leach and Gillet [160].

2.6 Primer on machine learning

Machine learning is a discipline, which studies algorithms that learn functions from data. It is closely related to pattern analysis and pattern recognition, where regularities, relations and structures inherent in the data are studied. By learning to detect patterns in a data, the system has achieved modelling power to predict patterns in unseen data.

In machine learning we aim to learn from data a function $f : \mathcal{X} \to \mathcal{Y}$, which maps the input space $\mathcal{X}$ to the output space $\mathcal{Y}$. The input data describes the process we are interested in, while the output data represents a prediction *target*: an interesting property or pattern of the data which should be predicted based on input data and a *loss function* $\ell(f(x), y)$ on predictions $\hat{y} = f(x)$ guiding the learning process. The function is learnt based on a *sample* data. In *supervised* machine learning the dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ contains examples with known target values. In *unsupervised* machine learning the target values are missing or implicit. The purpose of machine learning is to learn a function that *generalises* into unseen, new data.

Examples of machine learning approaches include clustering [125], neural networks [224], decision trees [204], support vector machines [33, 238] and Bayesian networks [84].

Often the input space $\mathcal{X}$ is a real vector space $\mathbb{R}^d$ of d dimensions, where data points describe various variables of the process of interest. In *regression* the target variable y is continuous and usually real-valued. In *classification* the target variable is a discrete label or category, allowing categorisation of data points. In binary classification there are two target classes, while in *multiclass* classification there are more than two classes. A *multilabel* classification entails prediction of multiple labels for each instance x.

The function f can take many forms. A simple linear function

$$f(x) = \mathbf{w}^T x = \sum_{i=1}^{d} w_i x_i$$

takes each input variable into account independently. A linear function ignores non-linear relations between the input variables. When the data is *linearly separable*, there exists a linear function that can divide the data into two classes perfectly.

In linear models a weight vector $\mathbf{w}$ is learnt and it describes the function f uniquely. In real life datasets the relationships within the data are often a mix of linear and non-linear components, with noise included from various

sources. Learning a non-linear function to reflect the inherent non-linearity of the data is a challenging task. An alternative approach is to map the data into a non-linear space with addition of variables that are various multiples of the original variables, and subsequently learning a linear model with the new higher dimensional data points. The resulting linear model in the high dimension can be projected back into the original space, resulting in a non-linear model in the original space.

In kernel methods this paradigm is followed. A high-dimensional new representation of the data is constructed implicitly, and robust linear models are employed.

A detailed look into kernel methods is given in the book by Shawe-Taylor and Christianini [238].

2.6.1 Support vector machine

We begin by distinguishing between abstract objects $x \in \mathcal{X}$ and their *representation* $\phi(x) \in \mathcal{F}$, where the feature space $\mathcal{F}$ is a vector space. Usually the vector space is a real space $\mathbb{R}^d$ with d dimensions. This notation provides a unified way to handle both vectorial objects and, for instance, graph objects G and their feature representations $\phi(G)$.

Support vector machine (SVM) is a family of non-statistical binary supervised machine learning algorithms, which base the classification on *linear discriminants*. A linear discriminant $\mathbf{w} \in \mathcal{F}$ projects the data $\{x_1, \ldots, x_n\} \subset \mathcal{X}^n$ into one-dimensional space as $\mathbf{w}^T\phi(x_i)$, where a location parameter or bias b is added to define the boundary $\mathbf{w}^T\phi(x_i)+b = 0$ between positive and negative classes, respectively. In a linearly separable case, the values smaller than b belong to the negative class $y = -1$ and values larger than b to the positive class $y = +1$, hence

$$y = sign(\mathbf{w}^T\phi(x) + b).$$

The hyperplane

$$\{x : \mathbf{w}^T\phi(x) + b = 0\}$$

is a normal perpendicular to the weight vector $\mathbf{w}$ and defines the *decision boundary*. The feature space $\mathcal{F}$ is divided into positive and negative half-space through $\mathbf{w}$ and b. The class of a new unseen data point is predicted by computing which side of the decision boundary it resides through $\hat{y} = \mathbf{w}^T\phi(x) + b$, where x is a new data point and $\hat{y}$ is a predicted class label.

There are various ways to find good $\mathbf{w}$ and b parameters. In the *linear discriminant analysis* (LDA) approach the $\mathbf{w}$ is directed between the empirical sample means μ_+ and μ_- of the respective positive and negative

class data points $S_+ = \{x_i \in S : y_i = +1\}$ and $S_- = \{x_i \in S : y_i = -1\}$, while the location parameter is at the midpoint between the said means, resulting in

$$\mathbf{w} = \Sigma^{-1}(\mu_+ - \mu_-)$$
$$b = \mathbf{w}\frac{\mu_+ + \mu_-}{2},$$

where Σ_+ and Σ_- are the empirical sample covariances of the positive and negative classes S_+ and S_-, respectively. In LDA the empirical sample covariance matrix $\Sigma = \Sigma_+ = \Sigma_-$ is restricted to isotropy for both classes.

In *Fisher discriminant analysis* (FDA) both class-specific datasets have their own sample covariance matrices and the $\mathbf{w}$ is hence learnt as

$$\mathbf{w} = (\Sigma_- + \Sigma_+)^{-1}(\mu_+ - \mu_-),$$

while b is as in LDA. This small change incidentally minimises the misclassification rate as well, by directing the weight vector $\mathbf{w}$ to follow the shape of the respective datasets. However, rarely a unique $\mathbf{w}$ exists that minimises the misclassification rate. FDA always picks one based on the covariances of the data points.

In SVM the $\mathbf{w}$ is chosen such that the minimum distance between the normal of the hyperplane $\mathbf{w}$ and all data points is maximised. This leads to a discriminant which maximises the distance γ to the *closest* points to the decision boundary, effectively concentrating on getting the most uncertain data points as far from the decision boundary as possible. A benefit of this model is that points that are far away from the decision boundary can be ignored for computational improvements.

By labelling the data points with $y \in \{-1, +1\}$ it is easy to see that a data point is correctly classified if

$$y(\mathbf{w}^T\phi(x) + b) \geq 0,$$

as the negative classes should have negative projection values along the $\mathbf{w}$, and positive class should have positive projection values.

The linear SVM formulation is then

$$\begin{aligned} \max \quad & \gamma \\ s.t. \quad & y_i(\mathbf{w}^T\phi(x_i) + b) \geq \gamma \qquad \text{for all } i = 1, \ldots, n. \end{aligned}$$

Note that for a particular $\mathbf{w}$ satisfying the constraints it holds that

$$y((a\mathbf{w})^T\phi(x)) = ay(\mathbf{w}^T\phi(x)) \geq a\gamma,$$

where $a \in \mathbb{R}_+$ is a constant. I.e. we can scale the $\mathbf{w}$ arbitrarily to achieve arbitrarily large margin.

Instead of finding the largest margin, we find the smallest norm of $\mathbf{w}$ such that the margin is at least exactly 1. We arrive at the conventional (yet, equivalent) SVM formulation

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ s.t. \quad & y(\mathbf{w}^T\phi(x_i) + b) \geq 1 \qquad \forall i. \end{aligned}$$

Soft-margin SVM

The standard hard-margin SVM requires the data to be linearly separable. This is an unrealistic assumption: complex phenomena often cannot be classified linearly, especially in noisy measurements. Hence, we add a penalty or error term

$$\xi_i = (1 - y_i(\mathbf{w}^T\phi(x_i) + b))_+,$$

where $x_+ = max(x, 0)$. The penalty is zero for a margin of at least 1, and rises linearly for smaller margins. The soft-margin SVM is

$$\begin{aligned} \min_{\mathbf{w},\xi,b} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i(\mathbf{w}^T\phi(x_i) + b) \geq 1 - \xi_i \qquad \forall i \\ & \xi_i \geq 0 \qquad \forall i, \end{aligned}$$

where C is the box constraint term, which determines the trade-off between minimisation of error ($C\sum_{i=1}^{n}\xi_i$) and the desire to have a smooth discriminant ($\frac{1}{2}||\mathbf{w}||^2$).

This constrained quadratic optimisation problem can be solved by using *Lagrange multipliers* α_i for each constraint $y_i(\mathbf{w}^T\phi(x_i) + b) \geq 1 - \xi_i$ and β_i for each constraint $\xi_i \geq 0$. A Lagrangian of the optimisation problem is

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\Big[\xi_i - 1 + y_i(\mathbf{w}^T\phi(x_i) + b)\Big] - \sum_{i=1}^{n}\beta_i\xi_i.$$

The corresponding dual is found by partial differentiation with respect to $\mathbf{w}$, b and ξ and finding a unique saddle point of L by setting the partial

derivatives to 0:

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \xi, \alpha, \beta) = \mathbf{w} - \sum_{i=1}^{n} y_i \alpha_i \phi(x_i) = 0$$
$$\frac{\partial L}{\partial b}(\mathbf{w}, b, \xi, \alpha, \beta) = \sum_{i=1}^{n} y_i \alpha_i = 0$$
$$\frac{\partial L}{\partial \xi}(\mathbf{w}, b, \xi, \alpha, \beta) = C - \alpha - \beta = 0.$$

Resubstituting the relations into the Lagrangian we obtain the value of the Lagrangian when minimised with respect to $(\mathbf{w}, b, \xi)$ as

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \phi(x_i)^T \phi(x_j)$$

under the constraints on α and β. The β is not part of the optimisation function and is omitted. As a result we have derived the dual problem equivalent to the original formulation. The dual is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ s.t. \quad & 0 \leq \alpha_i \leq C \qquad \forall i \\ & \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned}$$

According to the Karush-Kuhn Tucker (KKT) conditions only examples with a margin at most 1 have non-zero Lagrangian multiplier α_i. We call the examples with non-zero Lagrangian multipliers α_i *support vectors* due to the fact that only those data points participate in determining or "supporting" the discriminant $\mathbf{w}$.

The dual variable β can be recovered as $\beta = C - \alpha$ after the optimum of the dual is found. The vectors $\mathbf{w}$ and variable b are recovered from any support vector by

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \phi(x_i)$$
$$b = y_i - \mathbf{w}^T x_i = y_i - \sum_{j=1}^{n} y_j \alpha_j \phi(x_j)^T \phi(x_i).$$

The weight vector $\mathbf{w}$ is hence a linear combination of the data points [238].

The derived dual function of the primal soft-margin SVM formulation is still a constrained quadratic optimisation function. The benefit of the dual representation is twofold. First, the dual function includes our data points x_i only through dot products. A *kernel trick* can be applied on the dual form to perform SVM in a feature space associated with a chosen kernel. Second, by using kernel trick to efficiently precompute the dot products, the dual form only contains n variables and is independent of the dimensionality d, while the primal form has $n + d + 1$ variables.

2.6.2 Kernel methods

In the previous section we presented a robust linear classification algorithm based on principle of margin maximisation, and derived a dual form where the input data is only processed through dot products. In this section we review the principles of kernel methods, which explore the ramifications of representation of data exclusively through dot products, to great improvements in computational performance.

In conventional machine learning the data $x \in \mathcal{X}$ is represented through a *feature mapping* $\phi(x) \in \mathcal{F}$, where the feature representation $\phi(x)$ can be a real valued vector ($\mathcal{F} = \mathbb{R}^d$), a string or a graph. A machine learning algorithm is then explicitly designed to process such data. Alternatively, we can map a complex data type into a vector representation and utilise standard algorithms.

In kernel methods a radically different approach is chosen. Instead of representing the data explicitly as $\phi(x)$, the data is seen only through pairwise similarities. In practise, the data is represented through a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and the data set S is represented by the $n \times n$ matrix of pairwise similarities $K(x, x')$. All kernel methods process such matrices directly and ignore the original data. The approach standardises the machine learning algorithm to a fixed input. However the problem of data representation is now simply moved to constructing a good kernel for various types of data.

Kernel definition. A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel* iff it is symmetric (that is, $K(x, x') = K(x', x)$ for any $x, x' \in \mathcal{X}$) and positive definite, i.e.

$$\sum_{i,j}^{n} c_i c_j K(x_i, x_j) \geq 0$$

for any $n > 0$, any choice of n objects $x_1, \ldots, x_n \in \mathcal{X}$ and any choice of values $c_i \in \mathbb{R}$.

We call a valid symmetric positive definite kernels *Mercer* kernels[2]. The simplest Mercer kernel is an inner product between real vectors, that is

$$K(x, x') = \mathbf{x}^T \mathbf{x}' = \sum_{j=1}^{d} x_j x'_j.$$

This kernel is symmetric ($x^T x' = x'^T x$) and the positive definiteness is simple to verify by

$$\sum_i^n \sum_j^n c_i c_j K(x_i, x_j) = \sum_i^n \sum_j^n c_i c_j x_i^T x_j = || \sum_i^n c_i x_i ||^2 \geq 0.$$

The inner product kernel is also called a linear kernel. The kernel can be easily generalised into any data type by feature representation

$$K(x, x') = \phi(x)^T \phi(x').$$

Any mapping $\phi : \mathcal{X} \to \mathbb{R}^d$ for $d \geq 0$ is a Mercer kernel.

Definition 1. *For any kernel K on space $\mathcal{X}$, there exists a Hilbert space*[3] *$\mathcal{F}$ and a mapping $\phi : \mathcal{X} \to \mathcal{F}$ such that*

$$K(x, x') = \langle \phi(x), \phi(x') \rangle, \quad \textit{for any } x, x' \in \mathcal{X}$$

where $\langle v, u \rangle$ represents a dot product in the Hilbert space between any two points $u, v \in \mathcal{F}$.

This results highlights the first benefit of the kernel representation. Any Mercer kernel function is equivalent to a dot product in some feature space $\mathcal{F}$. Thus a kernel can be computed by representing the data x as $\phi(x)$ and computing the dot product there. However, we don't actually need to do this explicitly. It is sufficient to show that a similarity function K is symmetric and positive definite, and compute the function K directly. The representation $\phi(x)$ does not need to be computed – or even known – at all. There exists useful similarity functions that indeed are Mercer

[2]There exists a class of invalid kernels. For instance a max function is not positive definite, but has proved to be useful in practise [177]. Invalid kernels are not guaranteed to converge to a global optimum in SVM.

[3]A Hilbert space is a vector space accompanied with a complete dot product and complete norm functions. A Hilbert space is by definition also a Banach space.

kernels, however the feature space corresponding to these kernels can be infinite-dimensional or difficult to determine.

An example of such a kernel is the Gaussian radial basis function (RBF) kernel

$$K_{RBF}(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

where σ is the variance or bandwidth parameter. This is a Mercer kernel [238] which can be written as a dot product according to Definition 1. The feature space is infinite-dimensional and includes all possible monomials of input features with no restriction on the degrees [238].

Another class of kernels is the polynomial family of kernels

$$K(x, x') = K(x, x')^p,$$

where p is a non-negative integer. The corresponding feature space is indexed by all monomials of degree p

$$\phi(x) = \phi(x)_1^{i_1}\phi(x)_2^{i_2}\dots\phi(x)_N^{i_N},$$

where $i_1, \dots, i_N \in \mathbb{N}^N$ satisfies

$$\sum_{j=1}^{N} i_j = p.$$

The matrix $K = K(x, x')_{x,x' \in S}$ is called the *Gram* matrix and it completely characterises the dataset S. Often the kernel matrix is normalised as

$$\hat{K}(x, z) = \frac{K(x, z)}{\sqrt{K(x, x)K(z, z)}}.$$

The normalisation ensures that the kernel values lie on the unit hypersphere.

Kernel trick. The kernel trick is as follows

Proposition 1. *Any algorithm for vectorial data that can be expressed only in terms of dot products between vectors can be performed implicitly in the feature space associated with any kernel, by replacing each dot product by a kernel evaluation.*

The trick is simple, however it has huge practical consequences. An array of machine learning algorithms have kernel variants, including Principal component analysis [132, 229], linear discriminant analysis, canonical

correlation analysis, perceptrons and several clustering methods [238]. By kernelizating an algorithm we can turn the algorithm into a non-linear by simply using a non-linear kernel with no added computational cost at all.

In SVM the kernel trick is achieved through the dual form, where naturally only dot products of the input data exist. By replacing the dot products with kernels, we implicitly operate on objects in a possibly high-dimensional feature space. Hence, SVM is turned into an effectively non-linear classifier that still finds a global optimal.

2.6.3 Max-margin conditional random fields

The Max-margin conditional random field (MMCRF) is multilabel output prediction framework based on kernel methods [220, 221]. It extends the one-class SVM into a multilabel prediction by defining a conditional random field on the label structure. The MMCRF is applied in Paper **II**, where reaction graphs are classified into a sparse hierarchy of function categories. We refer the reader to the original paper for a thorough discussion of MMCRF [220].

In the context of multilabel classification, we consider a training set $((x_1, \mathbf{y}_1), \ldots, (x_n, \mathbf{y}_n))$ where $(x_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}$. The input space $\mathcal{X}$ is a set of objects and the output or label space $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_k$ consists of sets $\mathcal{Y}_j = \{-1, +1\}$. The label vector $\mathbf{y} = (y_1, \ldots, y_k) \in \mathcal{Y}$ is called a *multilabel* and its components y_j are called *microlabels*. We distinguish a pair $(x_i, \mathbf{y})$ from a pair $(x_i, \mathbf{y}_i)$, where x_i is input data, $\mathbf{y}_i$ is the correct multilabel and $\mathbf{y}$ is arbitrary, as *pseudo-example* to emphasise that such a pair has not been seen in the training data.

We assume an associative network $G = (V, E)$ on the output labels, where a vertex $j \in V$ corresponds to a j'th microlabel and edges $e = (j, j') \in E$ correspond to microlabel dependencies. The structure of the network G is fixed prior through some edge set E.

Additionally, we define a *joint feature space* $\mathcal{F}_{xy}$ through

$$\varphi(x, \mathbf{y}) = \phi(x) \otimes \psi(\mathbf{y})$$

as a tensor product between the input features $\phi(x)$ and output features $\psi(\mathbf{y})$. The tensor product contains all pairs $\phi(x)_j \psi(\mathbf{y})_k$ of input and output features. The benefit of a tensor kernel is that no alignment of input and output features is necessary. Instead, the kernel method determines the alignment automatically by learning the weight vector $\mathbf{w} \in \mathcal{F}_{xy}$, which also belongs to the joint feature space.

The MMCRF uses an exponential model class to determine the probability of an arbitrary multilabel $\mathbf{y}$ being correct for a given input object

x:

$$p(\mathbf{y}|x) \propto \prod_{e \in E} \exp\left(\mathbf{w}_e^T \varphi_e(x, \mathbf{y}_e)\right) = \exp\left(\mathbf{w}^T \varphi(x, \mathbf{y})\right),$$

where $\mathbf{y}_e = (y_j, y_{j'})$ is a pair of microlabels attached by the edge e, $\varphi_e(x, \mathbf{y}_e) = \phi(x) \otimes \psi_e(\mathbf{y}_e)$ is a tensor product of the input features and the output features of a pair of microlabels, with corresponding weights denoted by $\mathbf{w}_e$. The $\psi_e(\mathbf{y}_e)$ is a block of four features corresponding to configurations $(0,0), (0,1), (1,0), (1,1)$ of the labels of edge e in multilabel $\mathbf{y}$. Exactly one of these features is set as 1 according to the multilabel. Hence, the feature vector $\varphi_e(x, \mathbf{y}_e)$ represents the input features with respect to a specific edge configuration.

The primal optimisation problem is

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \\ s.t. \quad & \mathbf{w}^T \varphi(x_i, \mathbf{y}_i) - \mathbf{w}^T \varphi(x_i, \mathbf{y}) \geq \ell(\mathbf{y}_i, \mathbf{y}) - \xi_i \qquad \forall i \ \forall \mathbf{y} \\ & \xi \geq 0, \end{aligned}$$

which has a corresponding dual. The loss function $\ell(\mathbf{y}_i, \mathbf{y})$ measures the distance between a correct multilabel $\mathbf{y}_i$ and an incorrect multilabel $\mathbf{y}$ for the data point x_i. A simple choice is the Hamming loss, which counts the number of incorrect microlabels. The algorithm searches for a projection $\mathbf{w}$, such that the distance along the projection between a single correct example $(x_i, \mathbf{y}_i)$ and all incorrect pairings $(x_i, \mathbf{y})$ is preferably at least the corresponding loss. The optimisation problem is non-trivial and utilizes the network G as a conditional random field. We refer the details to the original paper [220].

After the optimal $\mathbf{w}$ is found, we solve the arg-max problem to get the multilabel prediction

$$y = \arg\max_{y \in \mathcal{Y}} \ \exp\left(\mathbf{w}^T \varphi(x, \mathbf{y})\right),$$

where the candidate space $\mathcal{Y}$ is often restricted to, for instance, multilabels seen in the training data.

Chapter 3

Computational reaction mapping

In this chapter we discuss motivation, background and algorithms for computational reaction mapping. We first formalise the reaction mapping problem with applications in Section 3.1. The problem is an instance of the graph matching problem and is closely related to the graph edit distance concept, which are discussed in Section 3.2. We continue by surveying reaction mapping algorithms in Section 3.3.

In Paper **I** optimal reaction mappings are computed, and subsequently used as inputs as reaction graphs in Paper **II** on reaction classification. The reaction graphs are discussed in the Chapter 4.

3.1 Introduction

In a chemical reaction $\rho = (R, P)$ a set of *reactant* molecules R are transformed into a set of *product* molecules P, often catalysed by an *enzyme*. An example reaction `alcohol dehydrogenase` oxidises ethanol into an acetaldehyde with the help of a NAD^{+}[1]. The reaction is described as

$$\text{ethanol} + NAD^{+} \longrightarrow \text{acetaldehyde} + NADH + H^{+}.$$

The reactants are ethanol and NAD^{+} with a positive charge (See Figure 3.1). The reaction products are acetaldehyde, uncharged NADH and a proton H^{+}. The reaction is catalysed by various alcohol dehydrogenase enzymes (See Figure 3.2). The enzyme acts as a catalyst, greatly increasing the speed of the reaction [104]. Informally, the reactants are bound to the enzyme, which modifies the reactants into products, which are subsequently released from the binding.

[1] Nicotinamide adenine dinucleotide

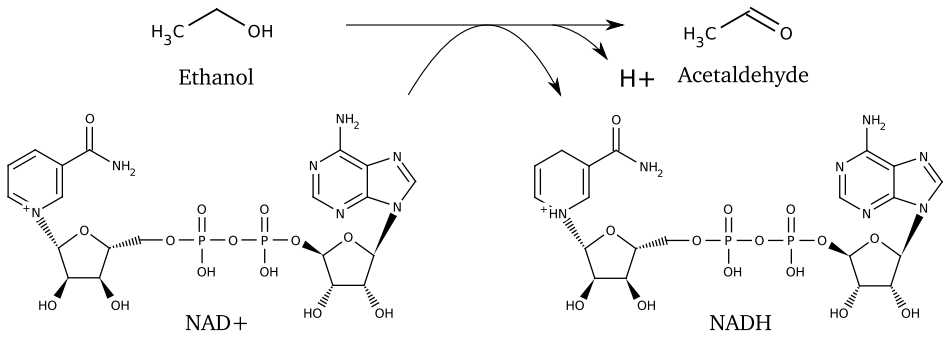


Figure 3.1: The alcohol dehydrogenase reaction.

During the reaction several chemical transformation steps are executed either successively or in parallel. In the human alcohol dehydrogenase the *reaction mechanism* contains 8 reaction steps [104], which can be represented as a sequence of bond breakage and bond formation events on the two reactants. These bond operations correspond physically to re-arranging the electron configurations of the reactant atoms [286]. The atom set remains unchanged. The result of the reaction process are the product molecules. Implicitly, the reaction transformation defines the correspondence between the atoms of the reactants and the products.

Reaction mapping is the reconstruction of these atom-to-atom correspondences, as occurring in nature. The mappings are hence dependent on the enzyme. Various experimental techniques can provide highly accurate reaction mechanisms and hence atom-to-atom correspondences. These include atom labellings, NMR and enzyme crystallography [104]. The experiments are complemented with manual curation, and accurate modelling through quantum-chemical or physico-chemical calculations. These approaches require substantial effort and time [104].

In metabolic modelling there are various applications for large-scale atom-level metabolic networks, which would be prohibitively costly and time-consuming using the aforementioned methods. Atom-level reconstruction of metabolic networks facilitates better understanding of the metabolic models [10], and can be used for consistency checking [11], global analysis of atom conservation ratios [118], reaction classification [287], and drug metabolisation prediction [26]. Another major line of applications lies in tracer experiments, where the atoms of cell's nutrients are isotopically or chemically labelled, enabling tracing of the "flow" of the atoms throughout the metabolic network [10, 187, 5]. Commonly used approach is the ^{13}C flux analysis, where isotopically labelled nutrient is fed to the cell and

Figure 3.2: Crystallographic schematic structure of the human ADH5 enzyme.

its pathways are traced by measuring the concentration of ^{13}C in different parts of the metabolic network [208, 209, 219, 26]. Atom-level mappings are necessary for this tracing.

While reaction mappings are collected to some extent, the databases are far from exhaustive. The ARM database of atom mappings contains atom mappings only for Carbon, Phosphorus and Nitrogen atoms [11]. The KEGG RPAIR database contains mappings only between metabolite pairs [151]. However, these are difficult to extend to mappings concerning the whole reaction without extensive manual work. In most other databases, atom mapping information is missing altogether.

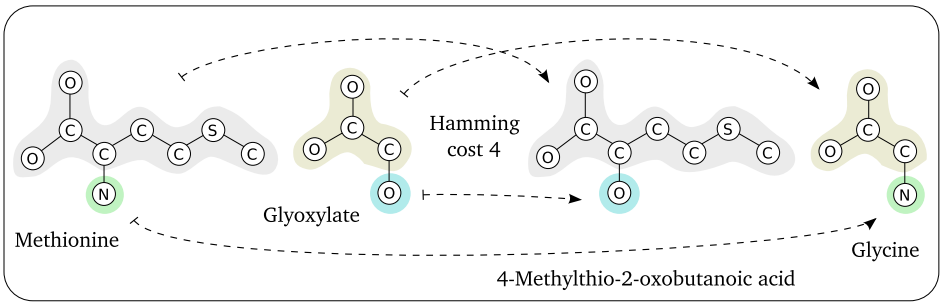
Computational reaction mapping methods rose to meet the demands of large-scale metabolic analysis and other applications. These methods search combinatorially for feasible reaction mappings using approximate optimality criteria.

In mathematical terms we define the *reaction mapping* as a bijective graph transformation $f : V_R \mapsto V_P$, where each vertex v in the reactant graph G_R is mapped to a vertex w in the product graph G_P, such that $l(v) = l(w)$. Reaction mapping is also denoted as *atom mapping*.

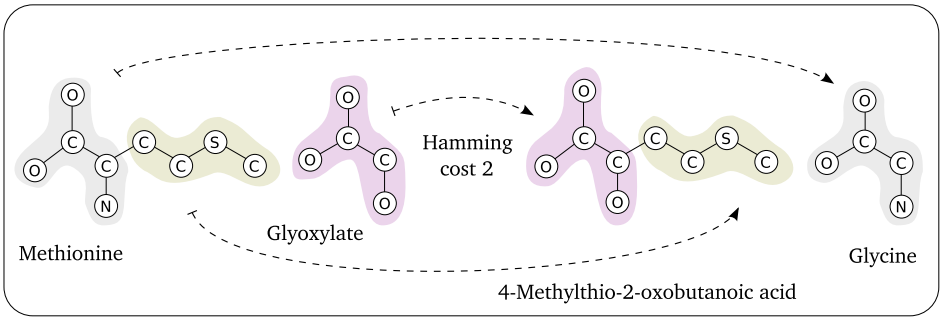
The *cost* c of a reaction is defined with respect to a reaction mapping f as a measure of edge operations (deletions, insertions and substitutions) implied by f:

Definition 2. *A cost $c(f)$ of a mapping f is the sum over all vertex pairs through the mapping*

$$c(f) = \sum_{(v,u) \in V_R \times V_R} c(v, u)$$



(a)



(b)

Figure 3.3: A reaction `R00652: L-Methionine + Glyoxylate <=> 4-Methylthio-2-oxobutanoic acid + Glycine` with two alternative atom mappings. The coloured regions indicate mapped atoms. (a) The Hamming cost is 4, as the atom mapping implies a swap of `N` and `O` atoms with respective cleavages and re-attachments. (b) The Hamming cost is 2 as only a single swap is necessary: the yellow substructure is separated from the grey substructure and attached to the purple substructure.

where the cost function is

$$c(v,u) = \begin{cases} w_{del}(v,u) & \text{if } (v,u) \in E_R \text{ and } (f(v),f(u)) \notin E_P \\ w_{ins}(v,u) & \text{if } (v,u) \notin E_R \text{ and } (f(v),f(u)) \in E_P \\ w_{sub}(v,u) & \text{if } (v,u) \in E_R \text{ and } (f(v),f(u)) \in E_P \\ 0 & \text{if } (v,u) \notin E_R \text{ and } (f(v),f(u)) \notin E_P \end{cases}$$

The weights w are non-negative.

Less formally, user-defined costs are paid for implied bond formation, bond removal and bond substitution events, respectively. Ideally, the cost $c(f)$ of a mapping f should correlate with the difficulty of executing a

certain reaction given a catalyst, or the probability of the reaction happening by chance. Accurate quantitative modelling of this kind is, however, computationally very demanding. At the extreme case it requires quantum chemical techniques or modelling of the energy landscapes of chemical reactions [87]. In metabolism applications, where models of thousands of reactions need to be produced, such models are not tractable. Instead, simple heuristic cost functions are often used.

In the simplest case both bond formation and bond removal are given a cost of 1, while bond substitution is costless[2]. In chemoinformatics this Hamming cost is used in the concept of *minimal chemical distance* [156, 16, 130, 131].

Problem 1 (Reaction mapping)**.** *The reaction mapping problem is "given a balanced chemical reaction, find a reaction mapping of minimum cost".*

In Figure 3.3 an example reaction is mapped through two alternative mappings with Hamming costs of 2 and 4, respectively. The lower mapping is optimal w.r.t. the Hamming cost. The upper mapping is produced by searching iteratively for maximum common subgraphs from both sides.

The optimal mapping is in general not unique. The reaction mapping problem with the Hamming cost function is known to be NP-complete [55]. In general it is easy to see that the reaction mapping problem is at least as difficult as the graph isomorphism problem, which emerges in the case of zero cost mapping [111].

An alternative problem formulation is the algebraic Dugundji-Ugi model, where the reaction is modelled as

$$R + X = P,$$

where R and P are the adjacency matrices of the reactant and product graphs, and X is a reaction matrix [68]. The reaction matrix defines the number of bond operations. In theory we can solve $X = P - R$, however, this requires a matching alignment of the adjacency matrices, i.e. the atom mapping. Several algorithms use the Dugundji-Ugi model as the basis for algorithmic development [162].

3.2 Graph matching

From a wider perspective, reaction mapping problem falls under the concept of *graph matching*. In graph matching mappings of type $f : V(G_1) \rightarrow$

[2]Bond substitution refers to a change in the order of the bond. These are usually less significant than structural changes, and are often ignored.

$V(G_2)$ from vertices of graph G_1 to vertices of graph G_2 are studied [42]. Graph matching is thus a process of finding a correspondence between the vertices and edges of two graphs, under for instance label matching constraints. Graph matchings are also *graph rewriting* procedures, where a vertex v ands its immediate edges from graph G_1 is replaced by a vertex $f(v)$, and its immediate edges from graph G_2 to turn the graph G_1 to graph G_2.

Graph matching is generally divided into exact and inexact matching. An exact matching requires strict correspondence between the two objects or at least among their parts. The former corresponds to a graph isomorphism problem, while the latter is a graph subgraph isomorphism problem. Weaker morphisms, such as *homomorphism*, allow many-to-one atom mappings. The maximum common subgraph problem is also an exact graph matching problem. Throughout these matchings an exact matching of at least subgraphs of the graphs is still assumed, and hence, a cost function of a matching is regarded unnecessary.

Inexact graph matching searches for an explicitly *error-tolerant* matching, where label matching or edge-preserving property are not strictly required. Instead, a cost function of Definition 2 is used to find a least-cost matching.

Inexact graph matching is closely related to both graph distance and graph transformations. In the latter model an optimal error-tolerant matching is seen as a least-cost sequence of graph edit operations to transform the graph G_1 into a graph G_2. The edit operations include insertions, deletions and substitutions of the edges. The cost is called the *graph edit cost* (GEC). The difference between graph edit cost and graph matching cost is subtle: while both values are in general equal, a least-cost matching can be realised with possibly multiple edit operation sequences.

A closely related concept of *graph edit distance* (GED) is obtained if the insertion and deletion operations carry identical cost. Hence, a GED is a metric, and also a well-defined graph distance function [52]. It can be shown that graph graph isomorphism, subgraph isomorphism and maximum common subgraph detection are all special instances of the graph edit distance computation under special cost functions [40, 41].

Formally, the reaction mapping problem is, depending on the cost symmetries, an instance of graph edit cost or graph edit distance problem [7, 258, 42, 88], which are special cases of inexact graph matching [52].

In chemoinformatics, the concepts of chemical distances and reactions have been discussed by Kvasnicka et al. [156].

3.3 Reaction mapping algorithms

Computational reaction mapping algorithms have been developed from two different points-of-view. The chemoinformatics community has mostly approached the problem from a viewpoint of iterative maximum common subgraph searches [248, 172, 164, 51, 252, 173, 182, 284, 270, 9, 106, 105, 189, 213, 162], where maximally large, identical "chunks" are searched from both sides of the reaction [5]. These chunks are then determined to undergo the reaction unchanged, resulting in an incrementally assembled mapping. These algorithms are often pragmatic, heuristic and often the optimisation criteria is not explicitly declared [55]. An algorithm is seen successful if the resulting atom mappings match expert knowledge. The MCS approach is computationally demanding and usually global optima are not found [106].

The idea of mapping atoms computationally in chemical reactions was first introduced by Vleduts [265]. In a pioneering work by Akutsu the drawback of ambiguous optimisation criteria of MCS was first documented and bounded combinatorial partitioning algorithms proposed instead [5]. In a pair of papers a weighted variant of MCS is introduced with an explicit optimisation criteria [8, 150].

Recently, Crabtree and Mehta presented efficient combinatorial algorithms to minimise the Hamming cost of reaction mappings [55]. The algorithms are built on the special combinatorial properties of the Hamming cost, and hence do not support other cost functions. In Paper **I** we introduce a graph edit distance as a flexible formalism to determine mapping cost. The method naturally supports arbitrary graph edit cost functions, and an efficient A*-based algorithm is developed to find optimal mappings.

Additionally, numerous graph matching algorithms are waiting to be applied to chemical graphs of reaction mapping problem [52, 88]. Inexact graph matching algorithms for labelled graphs are mostly search-based algorithms with various approaches to approximate algorithms [52]. Graph edit distance algorithms tend to focus on methods that learn the graph edit operation costs from a dataset [193]. Few algorithms are designed to support user-defined cost functions. These methods use e.g. binary linear programming [133] or the concept of supergraphs [79].

In following we present authoritative examples of aforementioned algorithmic classes. An exhaustive review of graph matching algorithms is left to Conte et al. [52], and a review on MCS on chemical graphs to Raymond and Willett [213].

3.3.1 Maximum common subgraph algorithms

The standard tool for computational reaction mapping has been maximum common subgraph algorithms where successive MCS's are found from the non-mapped regions of the reactant and product graphs until the whole reaction has been mapped. The MCS approach seems intuitive: it determines the successively largest regions of the reactant graph to undergo the reaction unmodified. However, Akutsu argues that there is no reason to believe that this leads to the correct mapping [5]. Crabtree and Mehta state that the MCS approach minimises the number of *reaction sites*, which is not equivalent to minimisation of bond operations [55].

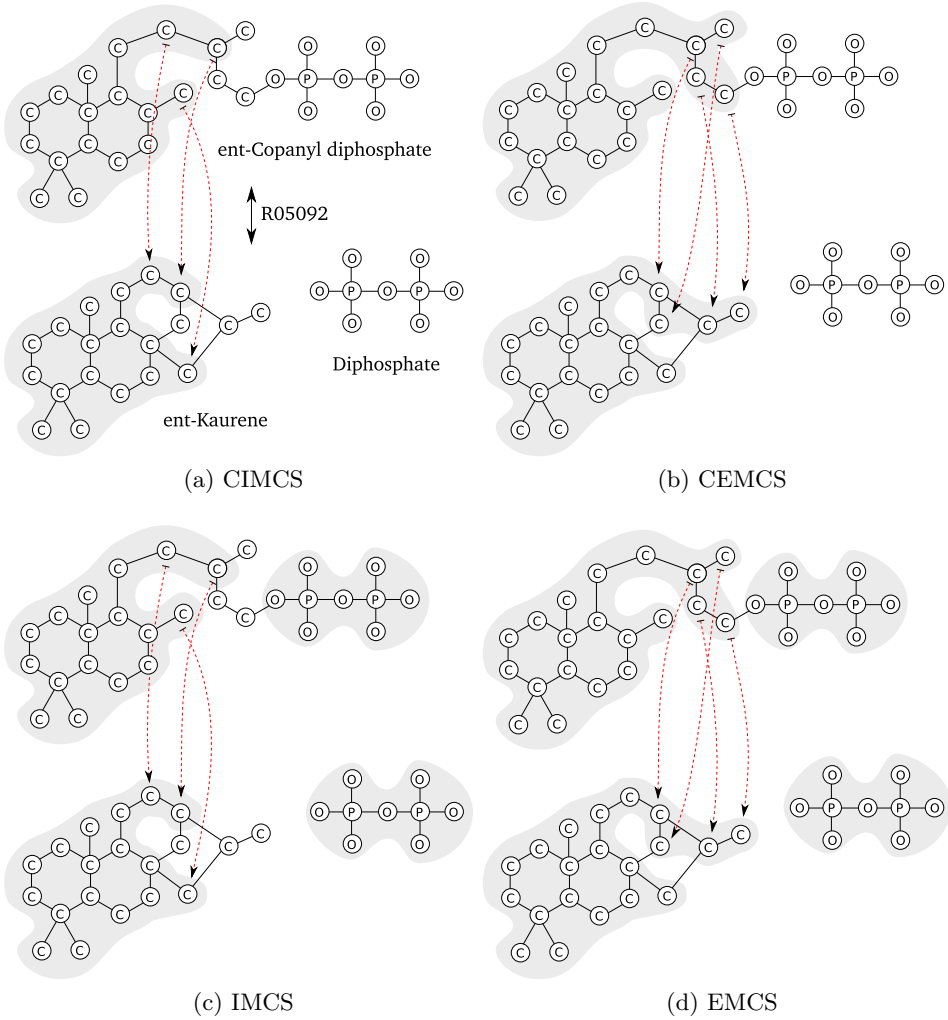
We begin the treatment on MCS algorithms by discussing the subgraph types relevant for the problem. The subgraph type (See Section 2.4) has a major effect on the resulting mapping. The connectedness of the subgraph determines how large regions the MCS covers. A connected MCS naturally defines regions of the reactants that undergo the reaction unmodified. Another distinction is whether the common subgraph is induced or edge-induced. In edge-induced subgraphs we are allowed to pick subsets of edges freely, while in induced subgraphs the edge set is determined by the vertices.

An example is indicated in Figure 3.4. Four cases are highlighted with an example reaction. The connected induced case (a) produces the smallest MCS. Searching for also disconnected subgraphs includes the $\mathtt{PO_4PO_3}$ parts. In (c) an edge-induced common subgraph is found. In (d) the found MCS happens to be a complete mapping of the graphs with a Hamming cost of 3. A single `C-O` bond is cleaved, while two `C-C` bonds are formed.

Another difference is the number of MCS's required to produce a full mapping. Cases (a) to (c) require multiple MCS's, while the last case (d) produces the atom mapping with a single MCS.

In [213, 182, 181] it is argued that edge-induced MCS is more suitable to characterise molecular similarity than induced MCS's, as it is the bonded interactions that are responsible for molecular properties and activities. However, most algorithms consider induced subgraphs due to their more favourable computational complexity.

In clique-based MCS the connection between an induced MCS between two graphs G_1 and G_2 and a maximum clique in the tensor product graph $G_\otimes$ of G_1 and G_2 is exploited [164, 51] (See Section 2.4). The clique problem is one of the six original basic NP-complete problems [89], with a vast literature [28, 200]. The Bron-Kerbosch algorithm [37] and its derivatives [148] are reported as fastest in practise by several authors [35, 94, 148]. Most recent advances have pruned the search space by detecting isomorphic



(a) CIMCS (b) CEMCS

(c) IMCS (d) EMCS

Figure 3.4: Four cases for maximum common subgraph on reaction `R05092: ent-Copalyl diphosphate <=> ent-Kaurene + Diphosphate`. The grey areas indicate the MCS in each case. Dashed lines highlight the non-trivial parts of the corresponding mapping.

cliques [46, 47]. Implementations by Koch [148] and Tonnelier [254] enumerate connected subgraphs, with Koch also utilising Whitney's theorem for edge-induced MCS [278].

The Whitney graph isomorphism theorem states that two connected

graphs are isomorphic if and only if their line graphs are isomorphic, with a sole exception of graphs K_3 and $K_{1,3}$. The K_3 is a three-vertex complete graph, and $K_{1,3}$ is a four vertex graph with a star topology. A line graph $L(G)$ of a graph G is a graph whose vertex set consists of the edge set of G. A pair of vertices of the line graph $L(G)$ are connected if the two corresponding edges in G are adjacent.

The theorem implies that with a sole exception of K_3 and $K_{1,3}$ graphs, an induced MCS on the line graphs corresponds to a edge-induced MCS on the original graphs. This property can be exploited to find edge-induced MCS using the clique-method by transformation into line graph domain [195, 155, 70, 148, 49].

The search-based MCS algorithms operate with the original graphs directly and are not restricted to any particular subgraph class. The method of Cuissart is a general MCS algorithm [57]. They construct a tree of connected induced subgraphs for both graphs. McGregor's algorithm is based on a correspondence matrix, which specifies which atom mappings are legal during the search [181].

The GMA algorithm by Xu [284] interprets one graph as query graph and the other graph as target graph. A permutation of the query graph vertices is produced, which is then used to walk on target graph with constrained back-tracking. If the walk completes, the permutation defines an isomorphism. The EMCSS algorithm extends GMA by using ideas from genetic algorithms [270].

3.3.2 Combinatorial algorithms

Akutsu presents several algorithms which involve graph partitioning by a *cut* of size C [5]. Blum and Kohlbacher extend these algorithms [26]. A chemical cut is done by removing at most C edges such that the endpoints of a removed edge belong to different components. The atom mapping problem is turned into one of finding a cut of size C on both reactant and product graphs, such that the resulting connected components are equivalent on both sides. These methods support only a subset of reactions by setting the maximum cut size to 1 for Akutsu and to 2 for Blum and Kohlbacher.

Recently, Crabtree and Mehta presented five combinatorial algorithms for minimisation Hamming cost [55]. They treat the problem as that of finding a minimum cut, that is, a set of edges to remove. They remove exhaustively edges from both sides of the reaction until the resulting partitions are isomorphic matches. Formed bonds are treated as removed bonds from the product side. Five variants of the method are presented with

formal analysis of computational complexities.

In the inexact graph matching community the A* algorithm is widely used. In A* a partial matching is updated and extended towards minimum cost direction according to heuristics on the cost of matching the remaining nodes. A global optimum is searched by using a best-first heuristic to guide the search process by evaluating

$$f(x) = g(x) + h(x),$$

where $f(x)$ is an estimated cost function of a *state* or partial matching x, $g(x)$ is the cost of the partial solution x so far, and $h(x)$ is the estimate of the cost to complete the partial solution. The algorithm is admissible, if $h(x)$ is a lower bound on the true future cost. If $h(x)$ is zero, the A^* is equivalent to Dijkstra's algorithm [65], where the next state is chosen based on only the cost so far. A priority queue is often used to keep a set of states to expand based on their $f(x)$ value.

The strength of A* lies in the ability to estimate the future costs $h(x)$. In branch-and-bound we use this information to prune unnecessary branches [96, 237]. In A* this information is used to choose the order of search tree traversal, which in general gives better runtime but at the cost of larger memory requirement [52]. Several A* algorithms and heuristics have been introduced for inexact graph matching [101, 69, 21, 20, 22, 255]

In Berretti et al. a *bipartite matching algorithm* is executed for a higher quality lower bound on the h [21, 20, 22]. The fastest algorithm for BPM is the Hungarian algorithm that is $O(n^3)$ [154, 282], and hence carries a substantial added computational cost to the A*. Other methods include greedily completing the partial matching to achieve an upper bound for the optimal solutions, which is then used to prune the search tree [111]. In Berretti et al. an incremental heuristic for future cost is used to avoid recomputation [22]. In Serratosa et al. a method is presented that exploits contextual information of the vertices [236].

In Paper **I** the bipartite matching algorithm and greedy completions are introduced for the reaction mapping as heuristics for the bounds. Atom neighbourhood information is used to distinguish between mappings of atoms of the same label.

3.3.3 Approximate algorithms

Suboptimal inexact graph matching algorithms find a locally optimal solution. Several approximate algorithms are search-based [74, 73]. In A* and in branch-and-bound algorithms the running time can be artificially limited and the best solution so far returned.

An alternative to search algorithms is to cast the problem of mapping discrete objects into one of continuous non-linear optimisation problem, for which many optimisation algorithms are documented. The problem is transformed into a continuous domain, solved using optimisation, and the results are finally converted back into discrete domain. In general these methods do not guarantee optimality. A family of methods determine the matching probabilistically [82]: a vertex is mapped to a distribution of vertices on the other graph. Then, iterative optimisation uses the current distributions and cost functions to update the distributions until convergence. A matching is determined by taking a maximum probability matches. A series of improvements and extensions have been proposed by multiple authors [122, 280, 50, 143], for instance into Bayesian measures [190] and Expectation-Maximisation optimisation [62, 171].

Other methods are based on neural networks [78], Kohonen maps [285] or genetic algorithms [56, 272]. An important part of genetic algorithms is the fitness function used to score candidates. Brown et al. count the number of preserved edges as a fitness function [39], while Wagener and Gasteiger also minimise the number of emerging components [268]. Fröhlich et al. study the use of parallel genetic algorithms [86].

Chapter 4

Kernels on molecular graphs

In this chapter we discuss the topic of kernel methods (See Chapter 2.6) as applied to biological graphs. Graph kernels are a natural match for both chemical and reaction graphs. We introduce graph kernels based on enumerating sequences in Section 4.2 and on subgraphs in Section 4.3. Then, in Section 4.4 we introduce the concept of R-convolution kernels, which forms an unifying framework for all structured kernels. We finish this Chapter by discussing kernels for chemical reactions in Section 4.5.

These concepts form the basis for original Paper **II**, where we introduce an efficient *path kernel* for classification of reaction graphs into functional categories.

4.1 Introduction

From a wider perspective, graphs – along with strings – are examples of structured data, which are characterised by lack of natural representation as vectors. Kernel methods are a natural choice for structured data, as no explicit vectorial representation is required. Instead, the learning is based on measuring similarity of graphs. Kernel methods for structured data are still under research [90, 15].

Graph kernels are kernels $K : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ where the objects $G \in \mathcal{G}$ are tuples $G = (V, E)$ of vertices V and edges E. This opens graphs objects to all kernel-based machine learning, as kernel methods only operate the input data through kernel functions.

The standard graph kernels are *spectrum kernels*, where an individual feature $\phi(G)_i$ of the feature vector $\phi(G) = (\phi(G)_1, \ldots, \phi(G)_d)^T$ of length d counts the number of times a subgraph $i \subset G$ occurs in the graph G (See

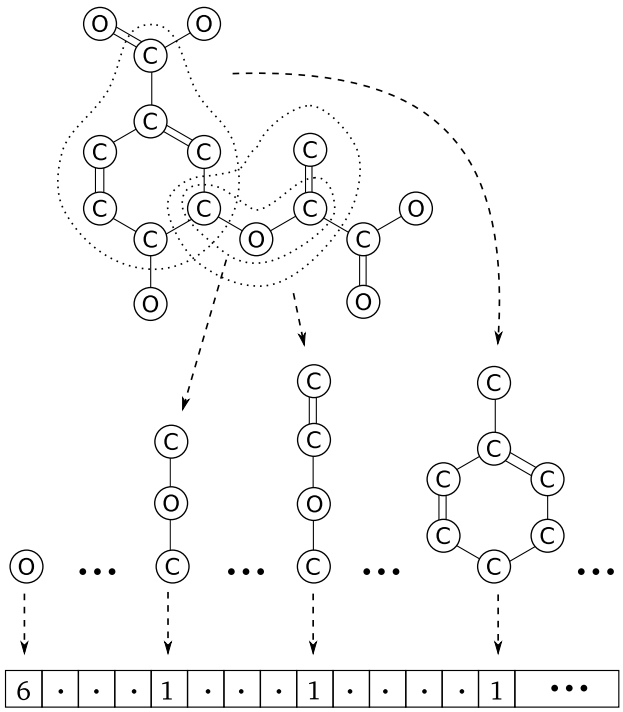


Figure 4.1: An illustration of a decay-less feature vector (bottom) of a Chorismate (top). Four substructures (middle) are highlighted, of which three are atom sequences. The numbers denote counts of the corresponding substructure.

Figure 4.1), often weighted by a feature-specific *decay* term λ_i:

$$\phi(G)_i = \sqrt{\lambda_i}|\{i \in \mathcal{I}(G)\}|,$$

where $\mathcal{I}(G)$ is the set of substructures of graph G of type $\mathcal{I}$. The decay term is important to down-weigh features if the feature set is not bounded. The kernel value is then

$$\begin{aligned} K(G, G') &= \langle \phi(G), \phi(G') \rangle \\ &= \sum_{i \in \mathcal{I}} \phi(G)_i \phi(G')_i \\ &= \sum_{i \in \mathcal{I}} \lambda_i |\{i \in \mathcal{I}(G)\}| \cdot |\{i \in \mathcal{I}(G')\}|, \end{aligned}$$

where $\mathcal{I}$ is the universe of substructures.

The similarity is thus measured as the number of matching substructures. This model only compares features that are exactly the same between two graphs. However, this is sometimes against what we desire. For instance, in an aromatic ring subsequent bonds are labelled by alternating single and double bonds. An alternative convention is to label all bonds with an order of "1.5". When comparing chemical graphs with mixed label conventions, there is no exact feature matches between the aromatic rings. Hence, the graphs appear erroneously dissimilar. In other cases, we might be interested to match also features that are not identical but are similar. In such cases, we introduce a *soft-matching* similarity function (itself a kernel) $\kappa : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ and match similar features together by

$$K(G, G') = \sum_{i \in \mathcal{I}(G)} \sum_{j \in \mathcal{I}(G')} \kappa(i, j) \phi(G)_i \phi(G')_j.$$

The more general soft-matching model corresponds to a tensor product feature space [238], where pairs of features are considered. Due to Definition 1 of Chapter 2.6, the graph kernel always implies a vectorial representation in some feature space $\mathcal{F}$.

Graph kernels, in general, follow either of these two models, with the former usually also explicitly stating the feature representation $\phi(G)$. Another division is based the class $\mathcal{I}$ of the subgraphs. Enumerating subgraphs introduces the problem of isomorphism [153]. Hence, more restrictive classes of subgraphs have been the focus of research, from vertex or edge sequences [263], to trees [239] and subgraphs of bounded size [153]. The trade-off between expressiveness and efficiency of the subgraph class has been discussed by Ramon and Gärtner [206].

The ultimate, yet unrealistic, goal is to have a one-to-one correspondence between the vector representation and the graph object. This is closely related to the *graph reconstruction* problem with a large literature [197].

In the machine learning community the focus has been on utilising 'all substructures' kernels, which enumerate exhaustively possibly millions of features for the kernel method to process and analyse. These models have achieved high predictive power [114], yet feature spaces of this magnitude offer little in terms of understanding what aspects or features of the graphs are important for the prediction task.

In contrast, the chemoinformatics literature puts emphasis on finding and using small sets of informative chemical descriptors that are most relevant for the prediction problem [226, 253, 269]. Widely used approaches are to use pre-defined substructures called functional groups [269], three-

dimensional conformation information [48, 117], frequent subgraph mining [64] or only small exhaustively enumerated substructures [269, 225, 81]. Kernelization of these features is straightforward and has led to improvements in classification performance [205, 117]. Another line of research concerns with representing molecular graphs as reduced graphs for a more compact and chemically informative representation [61].

Graph kernels have seen several applications in bioinformatics when the objects of interest are molecules, reactions or proteins. Graph kernels are an excellent match for molecular clustering [256], drug prediction [249, 48], and for molecular toxicity and mutagenicity prediction [249, 205]. Reaction kernels that are based on graphs have been used to predict whether a protein is an enzyme or not [30], and on reaction function prediction [227, 13, 12, 61, 30]. Graph kernels for proteins have been used to predict the class and function of proteins [240, 257, 157], as well as for disease outcome prediction [31].

4.2 Sequence based kernels

Sequence-based kernels enumerate labelled walks or paths of the graph as the representative structure of the graph.

4.2.1 Random walk kernels

Three kinds of walk models have been proposed. The random walk model by Gärtner et al. utilises adjacency matrix exponentials to count the number of random walks [91], while the general Markov chain model by Kashima et al. introduces a probabilistic random walker [135]. The *simple walk kernel*, which explicitly enumerates walk features, is discussed in the next section.

The Gärtner's model is based on the property of adjacency matrix exponentials where $[A^k]_{ij}$ denotes the count of k-length walks starting from vertex i and ending at vertex j given an adjacency matrix A of a graph G. The kernel is defined as the number of matching labelled walks in graphs G and G', modified by the length decay term λ,

$$K_{walks}(G, G') = \langle \phi(G), \phi(G') \rangle = \sum_{k=1}^{\infty} \sum_{w \in \mathcal{W}_k} \phi(G)_w \phi(G')_w,$$

where the feature representation is explicitly

$$\phi(G)_w = \sqrt{\lambda_w} |\{w \in \mathcal{W}(G)\}|,$$

where $\mathcal{W}(G)$ is the set of all walks in graphs G and $\mathcal{W}_k$ is the universe of walks of length k.

Gärtner proposed an elegant way to compute the kernel in the tensor product graph (See Section 2.4), where a walk corresponds to a pair of matching walks in the original graphs. Hence, we can count the number of matching walks in two graphs by counting the number of walks in the corresponding tensor product graph:

$$K_{walks}(G, G') = \sum_{i,j=1}^{|V_\otimes|} \left[\sum_{k=0}^{\infty} \lambda_k A_\otimes^k \right]_{ij},$$

where we count the number of matching walks up to infinite length between any two start i and end j vertices from the adjacency matrix $A_\otimes$ of the tensor product graph. Using geometric decay series of $\lambda_k = \lambda^k$ the kernel can be computed in roughly cubic time by inverting $(\mathbf{I} - \lambda A_\otimes)$, given that $\lambda < \frac{1}{\max_{v \in V_\otimes} deg(v)}$ [91].

For an exponential series $\lambda_k = \lambda^k / k!$ the kernel turns into an instance of a *diffusion kernel* with a natural down-weighing of longer walks [149, 238]

$$K_{diffusion} = \sum_{k=1}^{\infty} \frac{\lambda^k A_\otimes^k}{k!} = \exp(\lambda A_\otimes)$$

with unconstrained λ. The diffusion kernel has an interpretation as a covariance of a stochastic process on a random field. Also, diffusion kernels can be regarded as generalisations of Gaussian functions to discrete graphs [149].

The random walk kernel suffers from several shortcomings. The inversion method does not support soft-matching of walks naturally. To overcome this limitation, some methods have proposed to engineer the tensor product graph adjacency matrix to partially simulate soft-matching [91, 29]. Another problem is the inclusion of tottering walks, which do not provide any additional information.

To address these shortcomings, Kashima et al. [135] defined a Markov chain model where each walk $w = (v_1, \ldots, v_m)$ is assigned a probability

$$p(w|G) = p_s(v_1) \prod_{i=1}^{m-1} p_t(v_{i+1}|v_i) p_e(v_m),$$

where p_s, p_t and p_e are the starting, transition and stopping probabilities within graph G, respectively. The kernel is then a marginalisation

$$K_{marg}(G, G') = \sum_{\substack{w \in \mathcal{W} \\ w' \in \mathcal{W}}} \kappa(w, w') p(w|G) p(w'|G'),$$

where $\mathcal{W}$ is the universe of walks, and $\kappa(w, w')$ is a soft-matching label similarity function giving similarity of two walks, for instance based on their label similarities. The $\kappa(w, w')$ is positive only for walks of same length. For a hard-matching similarity $\kappa = \delta$ ($\delta(w, w') = 1$ if $w = w'$, 0 otherwise) only exactly matching walks are counted, and thus the other summation is omitted. The kernel itself is defined as the expectation of the κ over all walks, with the $p(w|G)$ term behaving as a feature extractor.

The kernel can be computed efficiently in the tensor product graph as

$$\pi_s^T (I - \Pi_t)^{-1} \pi_e,$$

where $\pi_s = (\pi_s(u, v))_{(v,u)\in V_\otimes}$ with $\pi_s(u, v) = p_t(u)p'_t(v)$; $\Pi_t = (\pi_t(v|u))_{(u,v)\in V_\otimes^2}$ collects transition probabilities $\pi_t((v_1, v_2)|(u_1, u_2)) = p_t(v_1|u_1)p'_t(v_2|u_2)$; and $\pi_e = (\pi_e(u, v))_{(u,v)\in V_\otimes}$ is the stopping probabilities $\pi_e(u, v) = p_e(u)p'_e(v)$.

A non-tottering variant was introduced subsequently by either using second order Markov chains, or by enriching the vertex labels to discourage it [174]. Runtime improvements for the marginalised kernel were reported by Vishwanathan et al. [263].

A unified framework of sequence kernels has proven the Gärtner's kernel to be a special case of the marginal kernel [263].

4.2.2 Simple walk kernels

The random walk kernels count the matching walks up to infinite length with parameter λ controlling the decay of longer walks. With $\lambda < 1$ the contribution of longer walks quickly becomes negligible and long walks have effectively no effect on the kernel value. This is sometimes against what we desire — longer walks may contain important information for e.g. graphs with repetitive substructures, where the walk length is required to surpass the diameter of the substructure to notice the repetition. Therefore we consider a finite-length walk kernels, where walks up to length m are constructed explicitly [61]. Working with explicit walks allows us to regard paths and non-tottering walks.

We count the number of matching m-length walks in two graphs by using following dynamic programming equations, defined over the tensor product graph

$$\begin{aligned} D_1(v) &= 1 \qquad \text{for all } v \in V_\otimes \\ D_k(v) &= \lambda_k \sum_{(u,v)\in E_\otimes} D_{k-1}(u) \end{aligned}$$

for each vertex $v_i \in V_\otimes$ where $n > 1$. The simple walk kernel is then

$$K_{simple}(G, G') = \sum_{v \in V_\otimes} D_m(v).$$

At limit $m \to \infty$ this corresponds to exponential walk kernel.

4.2.3 Path kernels

A path kernel is

$$K_{paths}(G, G') = \sum_{p \in \mathcal{P}} \phi(G)_p \phi(G')_p,$$

where $\mathcal{P}$ is the set of all labelled path sequences and the path counts $\phi(G)_p = \sqrt{\lambda^{|p|}}|\{p \in \mathcal{P}(G)\}|$ contain the decay λ as a non-negative exponential weighting term. The paths do not contain repetitive sequences, and hence can be regarded as more informative – and less numerous – than standard walks. For computational reasons, instead of full path kernels, a kernel based on shortest paths was introduced by Borgwardt et al. [30]. Here, we restrict the set of paths to only those which also coincide with a shortest sequence between the start and end vertices of the path. This radically limits the size of the path space.

In Paper **II** we introduce the first feasible path based graph kernel with dramatic improvement on classification performance over walks and shortest paths on reaction function prediction task.

In Figure 4.2 we highlight the difference between walk and path features, along with a reference method of unrestricted subgraphs. The actual feature vectors may also contain the decay λ, and possible tensor feature combinations in case of a soft-matching kernel.

4.2.4 Suffix trees for sequences

The path kernels – and to an extent, simple graph kernels – construct the sequence features explicitly. This raises a need to store the feature efficiently. The feature vectors are typically sparse, which can be exploited in data structures storing the features.

Let $\mathcal{A}$ be a finite alphabet with characters as elements. An empty string is denoted as ϵ. A string $S = s_1 \cdots s_m \in \mathcal{A}^m$ is a sequence of characters of length m. For a string $S = uvw$, the u is a prefix and w is a suffix of v, respectively.

A suffix tree T for a string S is a rooted tree with edge labels corresponding to strings. A path from root to leaves produces a distinct suffix

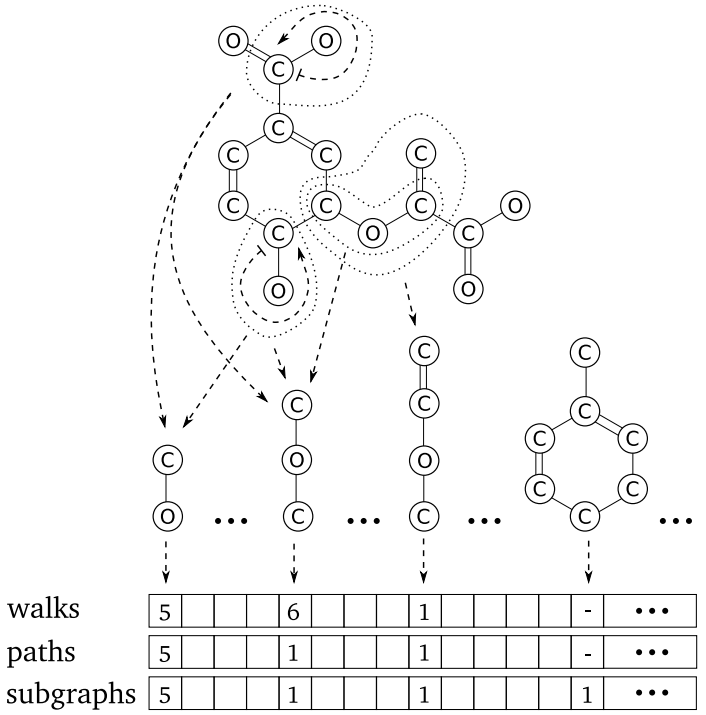


Figure 4.2: An illustration of counts of three feature types of Chorismate. The `C-O-C` feature of "walks" counts five tottering walks of `C-O` sequences and a single non-tottering `C-O-C` sequence. Only two of the `C-O`'s are highlighted for clarity. The "paths" features count only the unique non-repeating `C-O-C`. The last feature with a ring is not a sequence, and hence is counted only for "subgraph" type features.

$s_k \cdots s_m$ of S for some k. Each suffix is represented by a unique path. A suffix tree can be built in $\Theta(m)$ time for finite alphabets [97]. By following the paths from root the suffix tree can answer subsequence queries in $O(k)$ time for a subsequence of length k. A suffix array is an implementation of suffix tree in an array form with enhanced computational efficiency [176].

A generalised suffix tree represents all suffixes of a set of n strings $\{S_i, \ldots, S_n\}$ with varying length [251]. The construction relies on attaching a unique terminator symbol t_i to a string S_i and concatenating the strings together. The construction then follows the construction of a suffix tree, with analogous access operations. Generalised suffix trees are a standard approach in string kernels [163, 264]. Using generalised suffix trees to store sequences of graph kernels have been remarked by several authors [205, 249].

A compressed index is a data structure that combines compression and indexing, for instance, on suffix trees [80]. Even though the data structure is compressed, the data can still be accessed efficiently without complete decompression through special interface. In Paper **II** we introduce compressed indices for graph kernels to facilitate efficient storage of the paths of reaction graphs.

4.3 Subgraph kernels

An intuitive way to produce a feature mapping for graphs is to enumerate unrestricted subgraphs of the graph up to a size k, and to define the kernel as the number of common subgraphs. In chemical graphs small subgraphs are regarded as informative. For instance, an aromatic ring has a size of 6, a common carboxyl group a size of 4 and a phosphate group a size 5. Incidentally, proposed subgraph kernels count subgraphs up to size 7 [240, 153].

In the subgraph kernels $K_{subgraph}(G, G') = \langle \phi(G), \phi(G') \rangle$ an individual feature $\phi(G)_i$ is related to the occurrence of subgraph i in the graph G. Conventionally the subgraphs are defined as connected and induced subgraphs $i \subset G$ [153]. The problem of counting common subgraphs of unbounded size is NP-complete [91].

The *subgraph kernel* counts the number of isomorphic subgraphs between two graphs,

$$K_{subgraph}(G, G') = \sum_{\substack{H \subset G \\ H' \subset G'}} \lambda(H) \kappa_{\simeq}(H, H'),$$

where $\kappa_{\simeq} : \mathcal{G} \times \mathcal{G} \to \{0, 1\}$ is an isomorphism decision kernel and $\lambda(H)$ is a decay term relative to the size of subgraph H. Kriege and Mutzel proved that for a suitable λ, the subgraph kernel can be stated more simply as a *common subgraph isomorphism* (CSI) kernel counting the number of subgraph isomorphisms between two graphs G and G':

$$K_{CSI}(G, G') = \sum_{\varphi \in I(G, G')} \lambda(\varphi),$$

where φ is an isomorphism of all isomorphisms $I(G, G')$ between graphs G and G' [153]. In general, the CSI kernel counts the number of isomorphisms between subgraphs, while the subgraph kernel counts the number of isomorphic subgraphs. The former are, in general, more numerous, as several isomorphisms can exist between a pair of isomorphic subgraphs.

A soft-matching subgraph is called the *subgraph matching* (SM) kernel

$$K_{sm}(G_1, G_2) = \sum_{\varphi \in \mathcal{B}(G_1, G_2)} \lambda(\varphi) \prod_{v \in V(G_1)'} \kappa_V(v, \varphi(v)) \cdot \prod_{(u,w) \in V(G_1)' \times V(G_1)'} \kappa_E\big(e, (\varphi(u), \varphi(w))\big),$$

where $\mathcal{B}(G_1, G_2)$ is the set of all bijections between $V(G_1)' \subset V(G_1)$ and $V(G_2)' \subset V(G_2)$, the $V(G_1)'$ is the domain of the bijection φ, and κ_V and κ_E are the respective soft-matching kernels for mapping of vertices and edges.

The *graphlet kernel* is a subgraph kernel $\phi(G)^T \phi(G')$ where the feature vector $\phi(G)$ of a graph G is explicitly generated and normalised into a distribution [240].

Fröhlich et al. proposed an *alignment kernel* based on an maximal weighted bipartite matching π of a subgraph over the two graphs [85]. Assuming equal sized graphs G and G' for simplicity, the kernel is

$$K_{alignment}(G, G') = \max_{\pi} \sum_{v \in V(G)} \kappa\left(v, \pi(v)\right),$$

where κ is a soft-matching kernel. Contrary to author's claims, the kernel is not positive definite [263]. However, the alignment is still an informative measure of the similarity of the graphs [85].

4.4 R-Convolution kernels

On the seminal work of Haussler an unifying R-convolution kernel for structured objects was proposed, based on decompositions and combinations of decomposition kernels [107]. The formalism is flexible, and all kernels introduced during this Chapter can be interpreted as R-convolution kernels on graph structures.

The R-convolution kernel revolves around two concepts: (i) a decomposition of a structured object (for instance, a graph) in parts of different types (for instance, into walks, paths or subgraphs), and (ii) definition of a graph kernel through combinations of kernels on these parts. Formally, an R-convolution structure on a graph G is a triplet

$$\mathcal{R} = (\vec{G}, R, \vec{\kappa}),$$

where $\vec{G} = (G_1, \ldots, G_D)$ is a D-tuple of non-empty subsets $G_i \subset G$ of G, R is a decomposition relation on $G_1 \times \cdots \times G_D \times G$, and $\vec{\kappa} = (\kappa_1, \ldots, \kappa_D)$

is a D-tuple of Mercer kernel functions. Both kernels and parts are indexed by the *type* $i = 1, \ldots, D$, where $G_i \in \mathcal{G}_i$ is part of the universe of parts $\mathcal{G}_i$ of type i.

The relation $R(\vec{G}, G)$ is true iff $\vec{G}$ is a valid decomposition of G. Let $R^{-1}(G) = \{\vec{G} : R(\vec{G}, G)\}$ denote the multiset of all possible decomposition of G. An R-convolution kernel is then defined as

$$K_{\mathcal{R}}(G, G') = \sum_{\substack{\vec{G} \in R^{-1}(G) \\ \vec{G}' \in R^{-1}(G')}} \prod_{i=1}^{D} \kappa_i(G_i, G'_i).$$

All kernels introduced in this chapter are 'all-substructures kernels' that only use a single part type ($D = 1$) or two part types ($D = 2$), and are based either on an exact matching ($\kappa_i = \delta$) or on soft-matching kernels κ_i.

Menchetti et al. introduced the *weighted decomposition kernel* (WDK) highlighting the expressiveness of R-convolution framework [186, 48]. In the WDK the graph is decomposed into *selector* vertices $s \in V(G)$ and *context* subgraphs $Z \subset G$ around the selector. The inverse relation $R^{-1}(G) = \{(s, Z) : R(s, Z, G)\}$ enumerates all selector-context pairs found in the graph. The kernel requires an exact match of the labels of the selector vertices of two graphs G and G', but soft-matches the contexts according to their label distribution, disregarding the context structure. The kernel is

$$K_{WDK}(G, G') = \sum_{\substack{(s,Z) \in R^{-1}(G) \\ (s',Z') \in R^{-1}(G')}} \delta(s, s')\kappa(Z, Z').$$

The context is constrained to a fixed radius (authors propose 3) around the selector. The feature mapping is not obvious, however the convolution framework guarantees positive definiteness of the resulting kernel.

4.5 Chemical reaction classification

Chemical reaction classification concerns with automatic groupings of chemical reactions into categories according to their similarities and dissimilarities. The classification takes as input the reactant and products of the reaction.

A common prediction target is the EC function hierarchy, which defines a four-part hierarchical code *a.b.c.d* describing the reaction function. The first part consists of 6 main levels concerning with high-level functions, e.g. "ligases" and "transferases". The second level contains 63 categories,

```
1. Oxidoreductase reactions
2. Transferase reactions
3. Hydrolase reactions
4. Lyase reactions
5. Isomerase reactions
 5.1  Racemases and epimerases
 5.2  cis-trans-Isomerases
 5.3  Intramolecular oxidoreductases
 5.4  Intramolecular transferases
 5.5  Intramolecular lyases
 5.99  Other isomerases
 5.-
6. Ligase reactions
 6.1  Forming carbon-oxygen bonds
 6.2  Forming carbon-sulfur bonds
  6.2.1  Acid-thiol ligases
   6.2.1.1
    R00235  ATP + Acetate + CoA <=> AMP + Diphosphate + Acetyl-CoA
    R00236  Acetyl adenylate + CoA <=> AMP + Acetyl-CoA
    R00316  ATP + Acetate <=> Diphosphate + Acetyl adenylate
    R00925  ATP + Propanoate + CoA <=> AMP + Diphosphate + Propanoyl-CoA
    R00926  Propionyladenylate + CoA <=> AMP + Propanoyl-CoA
    R01354  ATP + Propanoate <=> Diphosphate + Propionyladenylate
   6.2.1.2
    R00389  ATP + Acid + CoA <=> AMP + Diphosphate + Acyl-CoA
    R01176  ATP + Butanoic acid + CoA <=> AMP + Diphosphate + Butanoyl-CoA
   ...
```

Figure 4.3: An excerpt of the Enzyme Nomenclature (EC) hierarchy.

which specify the reactant structures the reaction operates on. The third level contains 201 categories further specifying the structures. The last level is a running index that specifies the exact reactants (See Figure 4.3). For instance, a code 6.2.1.1 corresponds to "6 ligase reactions", "6.2 forming carbon-sulphur bonds", and finally "6.2.1 acid-thiol ligases". There are six known reactions of class 6.2.1.1 in KEGG.

In reaction function prediction a reaction is classified either to the six main categories, or the full EC-code is predicted (apart from the last part) [227]. The classification provides information on the function of an previously un-annotated reaction mechanisms. For instance, there is an estimated number of up to 200,000 plant metabolites [103]. Only a fraction of their reactions have been characterised [227]. Automatic function prediction can provide valuable clues to the function of the un-annotated reactions.

The E-zyme system by Kotera et al. analyses the correlations between an EC number and the alignment of reactants and products using MCS [152, 287]. Kernel methods are introduced by Saigo et al. [227] and Astikainen et al. [13]. Astikainen determines reaction similarity as the similarity between the participating molecules [13]. Saigo introduces a two-tiered reaction graph representation, where the relationships between participating

molecules have been annotated for additional information [227]. In Mu et al. certain idiosyncrasies of the EC hierarchy are discussed and an alternative reaction classification system is proposed consisting of 80 reaction classes [188].

In Paper **II** an atom-level reaction graph is proposed. The reaction graph utilises the full atom mapping, along with the reactant and product structures. A path kernel is applied to extract information from the reaction graph.

4.5.1 Reaction graph representations

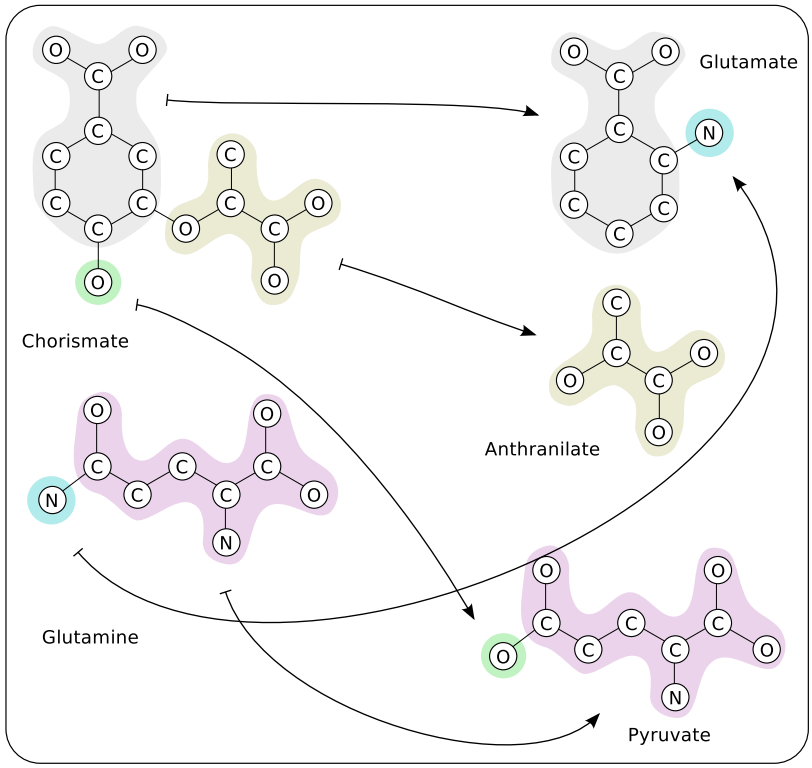
Reaction classification requires a representation of reactions. In the two-tiered reaction representation by Saigo et al. the outer tier is a graph with molecules as vertices and edges representing *interactions* between the molecules in the reaction. For instance, all reactants are joined into a clique to represent a reactant group. Reactants and products are joined by an edge only if the specific molecules have a relationship through the reaction. The interactions are determined with atom mappings using MCS and manual curation [151]. In the inner tier the molecule vertices are represented as standard chemical graphs.

Another formalism is the *reaction graph*, which results from the remark that the reaction (R, P) with reactants R and products P, along with the atom mapping f, contains overlapping information. For instance, the vertex sets of the R and P are identical: no atoms are lost or added during a balanced reaction. Also a majority of the edges are preserved across R and P as well.

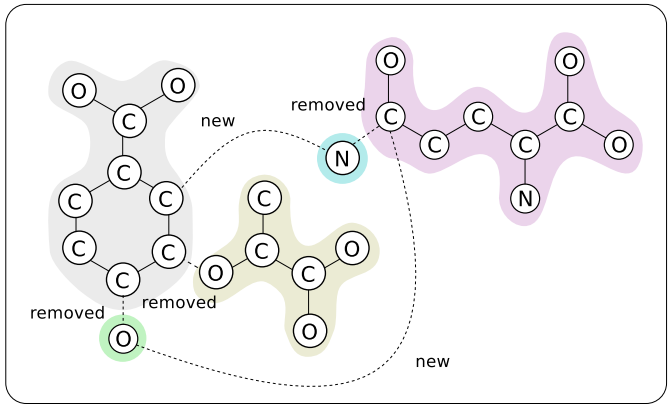
A reaction can be represented as a single graph through the atom mapping f. We define a *reaction graph* $G = (V, E)$ as a labelled undirected graph, where the vertices represent the atoms of the reactants and products, and the edges are labelled such that an edge common to both sides, through f, is labelled 0, an edge that is missing on reactants but exists on products as $+1$, and an edge that exists in reactants but is missing on products as -1. Thus, a set of formed new bonds are labelled with $+1$, while a set of broken bonds is labelled with -1 (See Figure 4.4).

This atom-level representation has several advantages. It represents the reaction as a single compressed graph that is both intuitive and easy to handle. It is unambiguous: there exists only a single reaction graph for a reaction triplet (R, P, f), which can be retrieved from the reaction graph.

Similar formalism has been discussed in a series of papers by Valiente et al. [76, 77, 75, 218, 217], while Yadav et al. discuss the potential benefits of such a representation [286]. They call these graphs *transformation graphs*



(a)



(b)

Figure 4.4: Reaction graph of reaction `R00986 Chorismate pyruvate-lyase` with EC code 4.1.3.27. (a) The atom mapping is highlighted with colour to indicate matching regions. (b) The reaction graph produced by taking the union of edges from both sides. The edges labelled "new" and "removed" correspond to -1 and $+1$ edges, respectively.

or *artificial chemistries*. In Valiente's notation, the transformation graph is applied on chemical graphs to transform them. However, the benefits of reaction graphs has been limited due to lack of high-quality atom mappings. In Paper **I** we introduce the reaction graph formalism through the computed optimal atom mappings on the KEGG database reactions.

4.5.2 Reaction kernels

The aforementioned two-tiered reaction representation by Saigo et al. was kernelized with a marginal random walk kernel within the upper-level reaction graph. A soft-matching inner kernel is introduced for vertices based on the molecules [227]. Astikainen et al. introduced three variants of reaction kernels acting directly on reactants and products, denoted as sum-of-reactants kernel, difference-of-reactants kernel and reactant matching kernel [13, 12].

In *sum-of-reactants* (SoR) kernel the feature vector of a reaction ρ is a sum of feature vectors of its reactants and products

$$\phi(\rho) = \sum_{M \in \rho} \phi(M)$$

with a kernel

$$K_{SoR}(\rho, \rho') = \sum_{\substack{M \in \rho \\ M' \in \rho'}} K_M(M, M'),$$

where $M \in \rho$ is a shorthand for both reactants and products of the reaction ρ.

Any kernel K_M can be used. The intuition behind this kernel is that two reactions are similar if the reactants they operate on are similar. The imminent drawback is that the kernel does not measure the reaction transformation in any obvious way. A *difference-of-reactants* (DoR) kernel takes the *change* of feature representation into account by defining

$$\phi(\rho) = \sum_{M \in \rho} \phi(M) - \sum_{M \in \rho} \phi(M)$$

with a kernel

$$\begin{aligned} K_{DoR}(\rho, \rho') = & \sum_{\substack{R \in R(\rho) \\ R' \in R(\rho')}} K_M(R, R') + \sum_{\substack{P \in P(\rho) \\ P' \in P(\rho')}} K_M(P, P') \\ & - \sum_{\substack{R \in R(\rho) \\ P' \in P(\rho')}} K_M(R, P') - \sum_{\substack{P \in P(\rho) \\ R' \in R(\rho')}} K_M(P, R'). \end{aligned}$$

The kernel picks up features that change during the reaction to highlight the transformative aspect of the reaction.

Both of the aforementioned kernels implicitly assume an underlying all-against-all matching between the reactant and product graphs. The atom mapping is taken into account by a tensor product between reactants and products

$$\varphi(\rho) = \sum_{M \in R(\rho)} \phi(M) \otimes \sum_{M \in P(\rho)} \phi(M),$$

which gives the *reactant matching* (RM) kernel

$$K_{RM}(\rho, \rho') = \left(\sum_{\substack{M \in R(\rho) \\ M' \in R(\rho')}} K_M(M, M') \right) \left(\sum_{\substack{M \in P(\rho) \\ M' \in P(\rho')}} K_M(M, M') \right).$$

All above kernels are unidirectional with straight-forward bidirectional variants [13].

Chapter 5

Metabolite identification with tandem mass spectrometry

Metabolite identification is the process of determining the metabolic contents of a cell sample, which is an important and prevalent step in biological experiments. Metabolite identification is also called structural elucidation in chemical literature. The qualitative knowledge of cell contents is a bottleneck for subsequent metabolic modelling and network analysis in metabolomics studies [194]. Yet in spite of its universality, the identification task is still both challenging and time-consuming due to measurement technologies that rarely are able to provide data to unambiguously identify the molecular structure. Mass spectrometry and tandem mass spectrometry are conventionally used for this task (see Section 2.3). The MS^1 spectrum often produces a wide view on the aggregate contents of the cells, while the MS/MS spectrum indicates a distinctive fragment pattern of a specific chemical specie isolated from MS^1.

In this Chapter we discuss the topic of metabolite identification with tandem mass spectrometry. We begin with a problem definition and introduction in Section 5.1, and continue by mass spectral analysis techniques in Section 5.2. We review the current de-facto approach of basing the identifications on a reference database of spectra in Section 5.3. We survey two alternative identification approaches. In Section 5.4 we introduce computational identification of product ions of MS/MS, and their usage in assisting metabolite identification. Finally, machine learning approaches to the problem are surveyed in Section 5.5.

The concepts of this Chapter form a basis for original Papers **III** and **IV**, where algorithms and models for fragment identification problem are introduced, as well as for original Paper **V**, where the first high-resolution kernel-based machine learning framework for metabolite identification is

proposed.

These two problems are closely related. The main difference is that in metabolite identification we aim to identify the unknown precursor metabolite based on the – also unknown – fragment peaks. In fragment identification the precursor metabolite is assumed to be known, and the identity of the fragments of the tandem mass spectrum are determined as substructures of the precursor.

For a general discussion of structural elucidation using mass spectrometry we refer to a review by Kind and Fiehn [142]. Neumann and Böcker [194], and Werner et al. [276] give excellent reviews on computational metabolite identification. Finally, for a readable introduction to machine learning of mass spectra we refer the reader to a review by Varmuza and Werther [262].

5.1 Introduction

The acquisition process of mass spectrometric data has been described with clarity by Neumann and Böcker [194].

From computational point-of-view, we model the tandem mass spectrum (See Figure 5.1) of a molecule $M = (V, E)$ as a collection $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{X}$ of 2-dimensional peak vectors $\mathbf{x}_i = (mass, int)^T$. We assume that each peak i of the spectrum is generated by a pool of a fragment structure $F_i = (V', E')$ of the molecule M. Usually a fragment is a substructure of the molecule, i.e. $F_i \subset G$, for all $i = 1, \ldots, k$. A peak consists of a mass of a fragment, and intensity possibly normalised to a percentage of the highest peak within the spectrum. A tandem mass spectrum dataset $S \subset \mathcal{X}^n$ is a set of n spectra measured with the same device. The size of each spectrum χ is the number of peaks it has, which varies within the dataset. Each mass measurement in the dataset contains an absolute assumed error ε inherent to the device. Hence, a peak's $mass$ is only an approximation to the true mass assumed to lie in the range $[mass - \varepsilon, mass + \varepsilon]$. For simplicity, we do not model the error through, for instance, Gaussian distributions.

The tandem mass spectrum χ originates from a precursor ion selected from a mass spectrum MS^1, which is usually also available along with any isotopic peaks for further constraints on the elemental composition. We denote the precursor peak $\mathbf{x}_{prec}$, which is always visible in the MS^1 spectrum, and sometimes also in MS^2 spectrum as an unfragmented ion (See Figure 5.1).

We present a formalised version of the metabolite identification problem as:

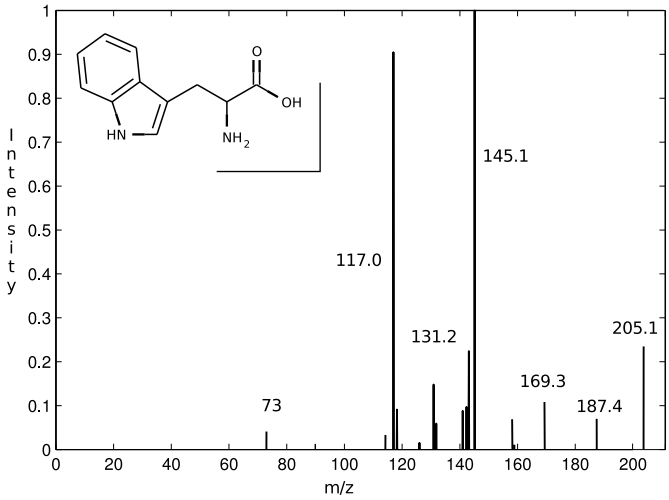


Figure 5.1: The MS/MS spectrum of a tryptophan (mass 204.23 u). The precursor peak at 205 u (the ionisation adds a proton with mass approximately 1 u) is visible. Possible isotope patterns are visible on peaks 117.0 u and 131.2 u.

Problem 2 (Metabolite identification)**.** *Given a tandem mass spectrum* $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *of an unknown metabolite* M *with a precursor peak* $\mathbf{x}_{prec}$ *and with an error* ε*, determine the structure of* M*.*

First clues to the identity of the metabolite are obtained from *mass spectral analysis* of the precursor peak pattern at MS^1. The peak masses and isotopic patterns provide constraints to the unknown metabolite's mass and elemental composition. The methods are equally applicable to MS/MS data, where an elemental composition of the fragment structures is linked to the precursor compound.

After spectral analysis, several identification approaches are in use. Matching the observed tandem mass spectrum to a database of reference spectra is a reliable identification method, given that the spectrometers used are similar and the reference database contains the measured metabolite [194]. If a match is not found from databases, additional information on metabolite structure is sought with interpretation of the fragmentation process. This includes prediction of the fragment structures and reconstruction of a fragmentation tree to hypothesise the likely occurred fragmentation reactions. Another line of evidence is gained by using pattern recognition algorithms to learn correspondences between the MS/MS peak patterns and the structural properties of the unknown metabolite. In prac-

tise, all three approaches are complemented with expert knowledge and manual reasoning.

Initial results on the ultimate goal of a fully automatic non-database based framework have been discussed very recently [230, 210]. We introduce an automatic framework for metabolite identification in Paper **V** using a machine learning approach and a novel statistical candidate ranking scheme.

Weissberg and Dagan give an illustrative example of organic compound identification using successively spectral analysis, reference databases, fragmentation prediction and candidate scoring along with manual analysis [275].

5.2 Spectral analysis

The metabolite identification problem is usually initiated with a spectral analysis producing elemental composition constraints for the possible metabolite structures, irrespective of the subsequent identification methodology. The same methods are used also to analyse the fragments, or even the MS^n spectra [283]. Main approaches are computing feasible elemental formulas [214], analysis of isotope patterns [27] and generation of isomers [159].

In elemental formula decomposition we search valid elemental formulas which sum up to the given *mass* with a mass error ε. The solution is

$$mass - \varepsilon < \sum_i n_i m_i < mass + \varepsilon,$$

where m_i is the mass of an element of type i and n_i is the count of element type i. The element set is often constrained to $\{C, H, N, O, P, S\}$ as these are most common in metabolites.

This is an instance of the *integer Knapsack* problem [59, 137]. The problem is NP-complete [89]. We search for all solution vectors $n = (n_1, \ldots, n_d)$, with non-negative coefficients, that satisfy the equation. A common solution to the problem is to formulate an integer problem and use an appropriate integer problem solver. An alternative is to use linear Diophantine equations [121]. The algorithm of SIRIUS software is based on transformation of the elemental masses into integer domain [27].

Not all elemental compositions can produce a chemically feasible molecule. A set of seven "golden" rules for valid elemental compositions are presented by Kind and Fiehn [141].

The constraints on the elemental formula depend only on the mass accuracy error ε. Additional constraints can be obtained by isotopic analysis. Each atom has a set of possible *isotopes* with a varying number of

neutrons in the nucleus. Hence, each atom induces multinomial distributions of masses, and subsequently a specific elemental composition has a specific isotopic distribution. The candidate elemental composition's simulated isotope pattern can be matched against the observed isotopic pattern for additional evidence [27].

The fastest known method to generate simulated isotopic peak patterns for a large set of elemental compositions is the Fourier Transform method [215, 27]. Next, the simulated isotopic patterns can be scored against the observed one with a fully Bayesian model [294, 293, 27] or with a simple euclidean distance measure [291].

Finally, the obtained candidate elemental compositions can be expanded into structures, which are denoted *isomers*. Isomer generation is a well-known problem in chemoinformatics literature [279]. The de-facto approach is the MOLGEN software which uses exhaustive generation with structural constraints [159]. The count of isomers of non-trivial elemental compositions is often impractically large.

5.3 Identification based on reference databases

One of the most common methods for compound identification is to query the observed tandem mass spectrum against a reference database of standard spectra [276]. This method is efficient and reliable as long as the database contains a corresponding spectrum, and the query and reference spectra are measured with compatible, or ideally, identical mass spectrometers with closely matching operating parameters [194].

Chemistry has a long tradition of storing measured mass spectra in databases. Several libraries of both MS^1 and MS/MS spectra are available, reviewed by Borland et al. [32]. Most prominent ones are the National Institute of Standards and Technology (NIST) database and the Wiley registry of mass spectral data, both of which contain both types of spectra measured on various spectrometers. Both databases are commercial. A prominent open alternative is the MassBank database, which focuses solely on MS/MS spectra [119].

Matching of the query spectrum against reference spectra in a spectral library necessitates a definition of a similarity function to score the retrieved candidates [38]. The similarity functions have been reviewed by Gower [100] and Stein [247, 244].

Typically up to tens of reference spectra share some peaks with the query spectrum. The simplest similarity measure is to count the number, or ratio, of matching peaks to achieve a peak count measure. Several meth-

ods include either raw or logarithmic peak intensities with experimentally defined weights [119, 71, 281], probabilistic measures [202, 198, 246] or Bayesian models [129]. An interesting approach by Lebedev and Cabrol-Bass computes the maximum common substructures of the best hits, and uses the estimated spectra of these substructures for additional comparison targets of the query spectrum [161].

The intensities in general are not regarded as very informative [194]. The X-Rank algorithm ranks the intensities within a single spectrum and only compares the intensities through the ranks [191].

As a concrete example, MassBank's query engine uses the Pearson correlation between query $\mathbf{w}^{(q)}$ and target $\mathbf{w}^{(t)}$ spectra vectors with elements

$$w_i = I_i^{\alpha} i^{\beta},$$

where i is the integral mass, I_i is the intensity of mass i, and constants α and β weight the importance of intensities and masses. The vector $\mathbf{w}$ contains zeroes for indices without a corresponding peak. The constants are set to, for instance, $\alpha = 0.5$ and $\beta = 2$ [119, 246].

The imminent drawback of the vector angle similarity measures is the alignment problem. Peaks are binned into bins of width 1 for nominal mass, and into narrower bins for high-resolution mass spectrometers. However, a peak can be placed into a wrong bin because of the measurement error, and thus misaligned. As a heuristic approach, mismatch is allowed so that close mass values are nevertheless compared together. MassBank uses a default value of 0.3 u mismatch [120].

5.4 Identification of product ions

In cases where the reference database does not contain the spectrum of the unknown compound, identification inevitably fails. However, a hitlist is still produced, possibly with highly compatible – and hence misleading – spectral matches.

For verification and additional structural clues, several methods analyse and predict the MS/MS fragmentation for additional evidence of the correct structure. The main idea is to estimate fragmentation reactions and cleavage sites of candidate metabolite structures with either rule-based [115, 1], combinatorial [116, 281, 113, 112, 250, 283, 17] or quantum chemical [23, 53, 165] approaches to see whether the simulated spectrum is compatible with the observed one. In fragment identification we explicitly assume the precursor structure to be known.

In original Paper **III** we introduce a combinatorial method for fragment identification using energy-based cost functions. The fragmentation tree framework is also proposed. In Paper **IV**, the methods introduced in Paper **III** have been implemented as software and further experiments are conducted.

5.4.1 Basic concepts

In a tandem mass spectrometer the selected compound pool is subjected to fragmentation and the resulting spectrum indicates peaks corresponding to these fragments. A common question is then, to identify which structures the resulting peaks correspond to. The major application of product ion identification is in metabolite identification [145], which has been the major context for fragment identification research.

The question is interesting also independent of metabolite identification. Theoretically the problem is interesting in the study of fragmentation mechanisms and theory [53, 6]. In drug metabolism studies, determination of the fragmentation of a pharmacologically active metabolite is an essential step in the characterisation of its biotransformations and the structures of resulting metabolites [18, 291, 203]. Product ion structures are also required to infer biotransformation sites [178]. A specific application is the $^{1}3$C flux analysis, where knowledge of the fragmentation patterns is beneficial for atom-level modelling [207].

While the structural information of product ions would be useful, it is nontrivial to obtain, both in theory and in practise. The fragmentation of a molecular ion in a tandem mass spectrometer is a complex, stochastic process that depends on e.g. the structure and chemical properties of precursor ions, the collision energy used, and the probabilities of the decomposition reactions as a function of the internal energy of an ion [184, 241].

The fragment structures F are generated from the parent compound through fragmentation reactions, which consist of bond cleavages and rearrangements, which form new bonds on the structures. Modelling of rearrangement fragments is especially difficult for combinatorial approaches and is still an open problem. We restrict ourselves to bond cleavages in this thesis. Hence, we can assume that the fragments are edge-induced subgraphs of M.

We are now ready to define the fragment identification problem:

Problem 3 (Fragment identification)**.** *Given a tandem mass spectrum χ of a known parent molecule $M = (V, E)$, identify the fragment structures $\{F_1, \ldots, F_k\}$ corresponding to $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, such that $F_i \subset M$ and $m(F_i) \in [mass_i \pm \varepsilon]$ for all $i = 1, \ldots, k$.*

In a weighted fragment identification problem we ask for fragment structures that minimise the total cost

$$c(\{F_1, \ldots, F_k\})$$

of the fragments. A cost function can, for instance, measure the number of bond removals necessary to produce said fragments or measure the energy landscapes of such operations. If all fragments are assumed to form independently, the cost turns into

$$c(\{F_1, \ldots, F_k\}) = \sum_{i=1}^{k} c(F_i).$$

This is a valid assumption if all fragments originate directly from the precursor compound without any interactions between fragments. In some mass spectrometers this is a realistic assumption. In combinatorial algorithms this assumption is made and fragment candidates are generated as subgraphs of the parent compound.

A "true" cost function would correspond to a quantum-chemical model that simulates the physico-chemical processes to a high accuracy [196, 4, 216]. Quantum-chemical modelling is feasible for fragmentation simulation of only very small molecular structures [6]. Apart from those methods, heuristic costs are commonly used to rank candidate fragments.

5.4.2 Combinatorial methods

Combinatorial methods determine possible fragment structures as subgraphs of the precursor chemical graph. These methods enumerate subgraphs that have matching mass to some of the measured peaks, and then use heuristic cost functions to score the candidates.

A major line of research concerns with bounded enumeration of candidate structures with a mass constraint. A common approach is to cut the parent graph with all possible combinations up to k bonds, and subsequently check the resulting fragment masses [242, 116, 36, 281]. Exhaustive enumeration is feasible for small values of k, with a value of 4 being common. The number of combinations of k edges out of m is $\binom{m}{k}$, where each combination of k removed edges can induce at most $k + 1$ fragments. Efficient combination generation algorithms are discussed by Knuth [146].

Generation of subgraphs of non-bounded size is an NP-hard problem. A simple graph traversal algorithm to generate exhaustively all induced and connected subgraphs was proposed by Rücker and Rücker [223, 222]. An interesting framework of *reverse search* with applications in subgraph

generation is presented by Avis and Fukuda [14] with a thorough discussion of the method by Kiyomi [144]. Both algorithms use topological orderings to prevent enumeration of same subgraph multiple times. However, neither algorithm regards isomorphism between the enumerated subgraphs, and hence might enumerate multiple subgraphs that are isomorphic, and hence equivalent. An enumeration algorithm by McKay supports enumeration of only non-isomorphic subgraphs [183].

In Paper **III** we adapt the Rücker's algorithm to fragment enumeration as the first non-bounded algorithm for the problem.

MASSPEC [242] and Spectool [36] both use an alternative representation by contracting fixed substructures of the chemical graph into non-breakable connected components represented by a single vertex, called *superatoms*. For instance, an aromatic ring of 6 atoms is contracted into a single atom with 6 edges. Only bonds between superatoms are allowed to break, which decreases the search space of bond cleavage combinations. A similar approach is in the decomposition method by Sweeney [250]. A minimum number of partitions are searched in such a way that maximum number of peaks can be explained as a combination of partitions. The method transforms the masses into integers and then represents the peak masses as a system of linear equations of the partitions. In practise the method produces easily unsolvable systems and Monte Carlo optimisation was employed.

Different cost heuristics are employed to rank the generated fragment candidates according to the feasibility of a certain fragment occurring in the mass spectrometer. A heuristic cost function by EPIC is

$$c_{EPIC} = |\Delta H| w_H + \sum_{e \in E_c} h_e w_e,$$

where $|\Delta H|$ is the number of hydrogen operations and w_H is the cost of a hydrogen operation, E_C is the set of broken bonds relative to the precursor, h_e is 1 for carbon-carbon bonds and 0.5 otherwise, and w_e is a bond type specific cost. For instance, phenyl bond carries a cost of 8, while an aromatic bond only 6.

In Papers **III** and **IV** a standard bond energy (BE) based cost function was introduced as a more realistic alternative to the heuristic costs of EPIC. A bond dissociation energy (BDE) is an accurate approximation of a bond strength and is defined as the standard enthalpy change when a bond is cleaved by homolysis at absolute zero [25]. BDE is dependent on the structural context around the bond. For instance, a methyl `C-H` bond has a BDE of 439 kJ/mol, while a benzylic `C-H` bond has a BDE of

Atom	H	C			N			O		S			P		
Order	–	–	=	≡	–	=	≡	–	=	–	=	≡	–	=	≡
H	-	412	-	-	388	-	-	463	-	338	-	-	322	-	-
C	412	348	612	837	305	613	890	360	743	272	573		264		
N	388	305	613	890	163	409	944	201	607						
O	463	360	743	1080	201	607	-	146	496	364	522	-	335	544	-
S	338	272	573					364	522	226	425			335	
P	322	264						335	544		335		205	351	489

Table 5.1: Examples of standard bond energies, in unit kJ/mol [288]. A dash denotes an impossible bond in normal conditions.

377 kJ/mol. However, BDE's are known for only the most common structures, and are hence unavailable for measuring bond strength in general chemical structures.

Instead, the standard BE can be used, which is the average BDE for a bond type. For instance, the `C-H` bond has a BE of 412 kJ/mol. The energy values for most bond types are listed in chemical handbooks [288] (See Table 5.1). BE has proven an useful approximation to bond dissociation energy and is correlated with the difficulty of breaking bonds [281, 112]. The cost function of a fragment is then

$$c(F) = \sum_{e \in E_c} BE(e),$$

where $BE(e)$ is the bond energy of a bond corresponding to an edge e.

The current state-of-the-art combinatorial metabolite identification method MetFrag uses bond energies and also additionally gives more weight to peaks of larger mass [281]. MetFrag draws metabolite candidates from chemical databases, such as KEGG or PubChem, based on the precursor peak's $\mathbf{x}_{prec}$ mass. Then, for each candidate metabolite, the lowest-cost fragments are estimated and their coverage of the observed tandem mass spectrum is measured. MetFrag ranks those metabolites higher that produce chemically less expensive fragments.

5.4.3 Fragmentation tree models

The combinatorial methods assume that all fragments are produced independently from the precursor, and use a corresponding cost function. However, in CID fragmentation secondary fragments are a common occurrence

[184]. In this case the cost function

$$c(\{F_1, \ldots, F_k\}) = \sum_{i=1}^{k} c(F_i)$$

errs as the fragment can be a substructure of another fragment, instead of the precursor compound (See Figure 5.2).

In Paper **III** we introduce a more difficult – and arguably more realistic – variant of the fragment identification problem, where the parent ions of each fragment should be determined. A fragment is formed by a cleavage reaction on another fragment, or on the parent compound. A *fragmentation tree* $T = (\{F_1, \ldots, F_k\}, E)$ forms where vertices F_i represent fragments and edges $(F_i, F_j) \in E$ denote fragmentation reactions (See Figure 5.2). The tree is rooted by the precursor M. The cost function then corresponds to the costs on the tree edges

$$c(\{F_1, \ldots, F_k\}) = \sum_{e \in E(T)} c(e).$$

The fragmentation tree is a subgraph of the *fragmentation graph*, which encodes all candidate fragments $\mathbf{F} = \{F_i^j : \forall i = 1, \ldots, k,\ \forall j = 1, \ldots, n_i\}$, where n_i is the count of candidate fragments for peak i. Let a directed fragmentation graph $G_F = (\mathbf{F}, E)$ be a tuple, where the vertices consist of all candidates for all peaks, and the edges connect any fragment F_i^j to a fragment $F_{i'}^{j'}$ iff the fragment F_i^j is a subfragment of the fragment $F_{i'}^{j'}$, i.e. $F_i^j \subset F_{i'}^{j'}$. An edge $(F_i^j, F_{i'}^{j'}) \in E$ has a cost $c(F_i^j, F_{i'}^{j'})$ of producing a fragment F_i^j from fragment $F_{i'}^{j'}$.

The fragmentation tree is now a *colourful subtree* of the fragmentation graph. A colourful subtree is a connected subtree of G_F, such that exactly 1 fragment candidate is chosen for each peak index (colour) i. We are interested in finding the fragmentation tree with a minimum cost over its edges:

Problem 4 (Minimum colourful subtree)**.** *Given a vertex-coloured edge-weighted fragmentation graph G, find the colourful subtree of G that has minimal*[1] *cost $c(T) = \sum_{e \in E(T)} c(e)$.*

[1] In Böcker and Rasche the problem is defined as a maximum colourful subtree problem with edge weights representing scores. We follow the notation that edge weights are costs associated with cleavages of fragments, and hence use a minimum colourful subtree formulation.

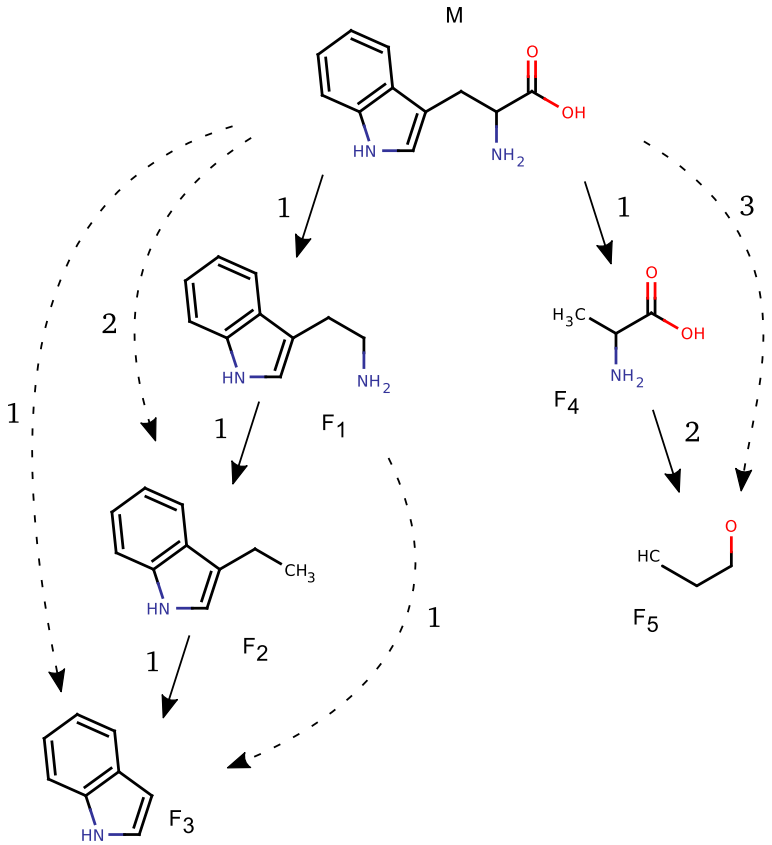


Figure 5.2: Fragmentation tree of a metabolite M with five fragment structures corresponding to five fragment peaks. The lines indicate possible fragmentation reactions with the number of bond cleavages in the reaction. The solid lines indicate a minimum Hamming cost explanation of the fragmentation process. The costs are dependent on the parent: for instance, the F_5 fragment requires cleavages of `C-N` and `C-O` bonds if the parent is F_4. However, if the fragment F_4 originates from the parent compound M, an additional cleavage of `C-C` is necessary. The total cost of the minimum cost tree is 6. The cost is 8 without any subfragment relations.

Böcker and Rasche provide formal algorithmic analysis of the problem [43, 44]. The problem is NP-hard by a reduction from SAT problem, even in the unweighted case [43]. The problem does not admit to constant factor approximations [66, 212]. Discussion of practical algorithms for the problem are presented by Rauf et al. [212].

Exact algorithms for minimum colourful subtree include mixed integer linear programming (MILP), dynamic programming, brute-force and branch-and-bound algorithms. In Paper **III** MILP models were introduced to find the optimal fragmentation trees. MILP is a general constrained optimisation algorithm, where a set of linear constraints are imposed on a set of variables that come from both integer and continuous domains. In MILP approach a binary variable is defined for each edge and vertex of the fragmentation graph. The constraints then force that exactly one vertex variable for each colour in the solution is set to 1 with at least one of the incoming edges set to 1 as well. Additional constraint ensures connectivity of the final tree.

In a dynamic programming approach we recursively optimise the minimal score $W(v, S)$ of a colourful tree with root v and colour set S. The computation begins with a colour set including only the parent compound with zero cost. In practise dynamic programming was reported feasible for only spectra with 10 to 15 peaks [211].

For large fragmentation graphs the exact algorithms have been proven infeasible. A greedy algorithm considers inclusion of edges in order of their costs. Another greedy strategy considers colours in some order and adds a fragment of that colour that promises the least increase in total cost. A hybrid strategy dynamic programming is used for only a subset of peaks to get a partial optimal solution. This is then completed with greedy heuristics. According to one study, experts regarded the fragmentation trees by heuristic methods as inaccurate [211].

Hufsky et al. has extended the fragmentation tree computations to multiple input spectral measurements of the same precursor [228]. Another method measures the change of intensities of the fragment peaks when subjected to ramping up the collision energy, which can give clues to whether the peak is produced by a primary or a secondary fragment [17].

In Böcker and Rasche, the fragmentation tree from Paper **III** is adapted to hold molecular formulas, instead of candidate fragment structures, as vertices [43]. Instead of generating candidate fragments, they generate feasible molecular formulas. The method provides no fragment structures, but does not require the precursor structure to be known. Hence, a fragmentation tree is used for evidence to metabolite identification problem with an

unknown precursor. The method is also robust with rearrangements as it does not take the structure into account.

5.4.4 Metabolite identification via fragmentation trees

A recent metabolite identification method computes optimal fragmentation trees where the fragments are represented by molecular formulas instead of candidate structures [211]. Molecular formulas can be deduced from mass spectra only, with no information of the parent compound. Hence more reliable molecular formula peak annotations are given as dependencies between peaks are taken into account.

The method begins by acquisition of tandem mass spectra with various collision energies and subsequent merging of the spectra. For each peak, the possible elemental compositions are estimated with SIRIUS using mass measurement error, isotopic pattern and MS/MS data. A fragmentation graph is constructed over the elemental compositions and least-cost fragmentation tree computed. The resulting molecular formula annotations are analysed and verified by domain experts, MS^n experiments, rule-based mass spectral simulation and comparison of different fragmentation trees. The metabolite identification is still ultimately manually done, but with help of the deduced information [123].

Later the model was extended with alignments of fragmentation trees and hierarchical clustering for a more automatic metabolite identification with promising results [210]. A fragmentation tree is aligned against a collection of known fragmentation trees. Instead of comparing tandem mass spectra, we compare fragmentation trees for a more insightful comparison. Further analysis is done by two-way hierarchical clustering of the alignment targets and chemical similarity of the underlying chemical structures.

5.4.5 Rule-based methods

An alternative to the combinatorial fragmentation tree search is the rule-based method, where the fragmentation tree is constructed by simulation of the fragmentation process, starting from the parent compound. The rule-based methods utilise the decades of knowledge of fragmentation to form deterministic rules, which state which bonds of the ions are subject to cleavage. The collection of cleavage rules is encoded in a database and a system is devised to decide which fragmentation occurs in case of alternating fragmentation rules [184].

The fragmentation rules are roughly categorised into *general* and *structure-specific* rules. The general rules are often credited to the seminal book by

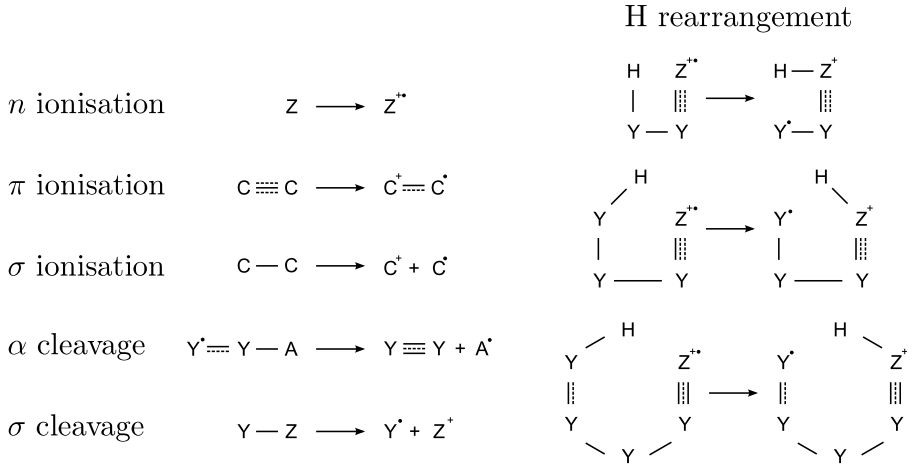


Figure 5.3: General fragmentation rules. Pluses indicate positive charge and dots indicate radicals. "A" is any atom, "Y" is any atom except hydrogen and "Z" is an atom with a free electron pair. The dashed bonds indicate alternative orders.

McLafferty and Turecek [184].

The general rules include n, π and σ-ionisation reactions, α and σ cleavage reactions and H-rearrangement reactions (See Figure 5.3). The fragmentation proceeds by first choosing a suitable ionisation reaction. Then, cleavage and rearrangement reactions are executed recursively. Uncharged fragments are discarded after each step. The general mechanisms tend to generate many false positives [93].

The structure-specific rules apply for a specific structure. These are usually determined from literature and can contain wildcards. Two widely used, and proprietary, software of rule-based fragmentation are MassFrontier by HighChem and ACD/MS Fragmenter by ACD/Labs. For instance, MassFrontier uses both general and structure-specific rules. It contains a manually curated, proprietary database of 19,000 fragmentation mechanisms as reported in thousands of publications in mass spectrometric journals.

Open alternatives to these include MOLGEN-MSF [231] and Jchem Fragmenter [166]. MOLGEN-MSF implements the general fragmentation rules as described above [139]. In case of alternative rules a scoring function examines the intensities of the peaks of the corresponding fragment products. A fragment with higher intensity is preferred. JChem Fragmenter contains a set of RECAP rules and accepts any custom rules in SMARTS

format [166].

In studies by Schymanski et al. the subpar performance of using MassFrontier, ACD/MS Fragmenter and MOLGEN-MSF for metabolite identification is reported [231, 232]. All three methods showed structural bias towards asymmetric structures [231], while the match values given by ACD/MS Fragmenter to candidates based on their fragment identifications were reported close to random [231, 232]. In Paper **IV** the identification accuracy of MassFrontier was reported low and categorically failing any negatively charged peaks.

5.5 Machine learning

In machine learning based metabolite identification we study and exploit the relationships between tandem mass spectral data and the metabolite structures. We assume that the tandem mass spectral signals correlate with the substructures present in the measured compound, and thus prediction of structural patterns – or even the structure itself – is possible. Studies by Varmuza [260] and Demuth et al. [63] discuss, and end up supporting, the validity of this assumption.

The problem of predicting the precursor metabolite directly is very challenging, and hence a standard approach is to instead predict substructures, chemical features or molecular class memberships [260].

In mathematical terms, the former problem of direct metabolite identification is a *structured prediction* problem: given a spectrum $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{X}$, learn a function $f : \mathcal{X} \to \mathcal{M}$, where $\mathcal{M}$ is a space of molecules. By representing molecules as chemical graphs, the problem is transformed into one of predicting graph objects. Currently this task is unrealistic. The structured prediction problem is under ongoing research [15].

The latter approach corresponds to learning a set of mappings $f_i : \mathcal{X} \to \{0, 1\}$, each of which predicts whether a substructure or property i is present in the unknown metabolite. We denote prediction targets y_i as *fingerprints* and, in Paper **V**, call the whole approach the *fingerprint model*, in contrast to the structured prediction model (See Figure 5.4). In general there is no guarantee that predicting a particular subtask f_i is feasible based on the information on the tandem mass spectra. Some properties of the metabolites do not reflect to the spectrum (such as charge localisation), while others might be masked by noise and measurement errors.

Computationally the metabolite identification problem is now rendered substantially more feasible, however this introduces a new problem of reconstructing the metabolite from the fingerprint predictions $f_1(\chi), \ldots, f_m(\chi)$.

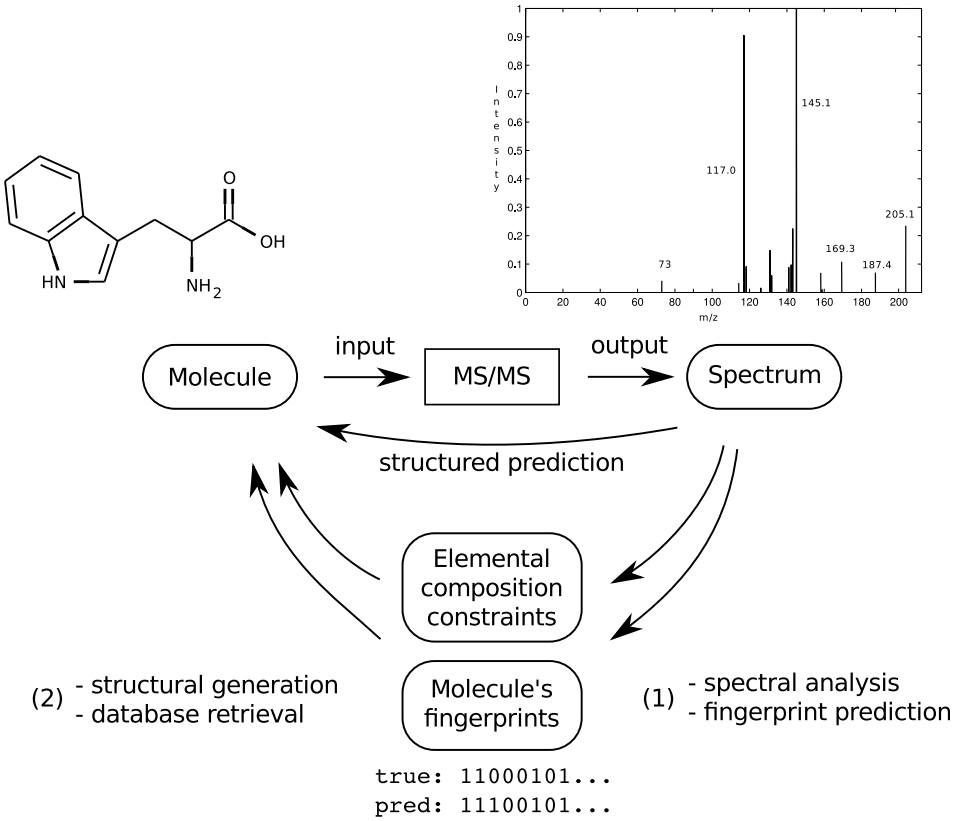


Figure 5.4: Metabolite identification overview through machine learning. In structured prediction a molecular graph is directly predicted. In the fingerprint model we utilise fingerprints and other constraints.

In chemical literature, this problem is often not addressed. Instead, it is assumed that a domain expert is able to elucidate the structure based on the individual predictions $f_i(\chi)$, or the original purpose of fingerprint prediction was other than precursor identification [262].

A MOLGEN-MS framework was proposed by Kerber et al. as complete pipeline for metabolite identification [138, 139]. The method uses isotope peaks for elemental composition constraints and predicts fingerprints from MS/MS data [262]. Both results are used to constrain isomer generation [159]. Finally, the candidate isomers are validated using rule-based simulation to see whether the simulated spectrum matches the observed one. The framework has not been completely published.

Recently a framework implementing the general idea of Kerber et al. was brought forth [232]. Schymanski et al. implemented the first three

steps of the framework to progressively eliminate false candidates from consideration [232]. In another approach the elimination of candidates is replaced by an ensemble scoring scheme and addition of chemical feasibility calculations [230]. Additionally, the simulation phase was replaced by MetFrag, which uses combinatorial fragment generation and energy-based scoring [281].

The current methods for fingerprint prediction universally only support nominal mass spectra, or higher accuracy spectra with fixed bin widths [260, 230]. Hence, high-resolution mass spectrometers and their accurate peak mass values are not exploited in the machine learning task. In Paper **V** we introduce a machine learning framework for metabolite identification with the first high-resolution mass spectral kernels.

5.5.1 Using fingerprints for metabolite identification

Prediction of fingerprints consists of three components: (i) mass spectral input features $\phi(\chi)$, (ii) substructure output features $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$, and (iii) the prediction algorithm f_i.

Input features. The purpose of mass spectral input features is to obtain a set of features that are informative to the molecular substructures to be predicted. The features $\phi(\chi)_i$ are either linear or non-linear functions of the selected peak intensities, denoted as I_m for the intensity of peak of integral mass m. A majority of the following features were proposed early in the pioneering work on STIRS software [60] and have been acknowledged since [72, 58, 277, 109].

The simplest feature is directly the intensities of all peaks,

$$\phi_{int}(\chi)_i = I_i.$$

A common approach is to take logarithms over the intensities, due to large variance of the intensity. The usage of binary intensities are discussed from information theoretic point-of-view by Scott [233, 234]. Small peaks below up to 5% intensity are usually discarded as unreliable [63].

The neutral loss features

$$\phi_{nloss}(\chi)_i = I_{m_{prec}-i},$$

where m_{prec} is the mass of the parent ion, measure the neutral losses from the parent compound [72]. The neutral loss features set the zero index at parent ion mass, with all other peak masses relative to it. It can detect a loss of a particular mass invariant of the parent ion mass.

Intensity ratios

$$\phi_{ir}(\chi)_i = \frac{I_i}{I_{i+\Delta m}}$$

measure the ratio of intensity of peaks that are Δm apart. Δm is often between 1 and 14, which measures isotopic ratios [262].

A common pattern in metabolites is the $\mathtt{CH_2}$ carbon group, which is easily cleaved from the end of a carbon chain. The carbon group has integral mass of 14. The cleavage of carbon groups can be detected by a intensity sum feature

$$\phi_{is}(\chi) = \sum_{m \in \{m_1, \ldots, m_d\}} I_m,$$

where masses at intervals of 14 are summed, i.e. $mod(|m_i - m_j|, 14) = 0$ for all $i, j \in 1, \ldots, d$. Other intervals include the geometric series of masses $I_{m/c}$ to link structurally identical ions that have different charges c.

Autocorrelation features

$$\phi_{ac}(\chi) = \frac{\sum I_m I_{m+\Delta m}}{\sum I_m I_m}$$

measure the mass differences between peaks up to some Δm range. Finally, peak combinations

$$\phi_{comb}(\chi) = \prod_m I_m$$

are alternative to intensity sums.

Various feature selection procedures have been applied to choose an informative subset of spectral features: from Fisher ratio selection [67, 259], greedy forward selection [168, 259], and genetic algorithms [289] to heuristic data mining [292] and principal component analysis [262]. The feature size decreases usually to some tens of discriminative features to facilitate statistical analysis [262].

All aforementioned input features assume integral mass values, and hence do not support accurate peak measurements.

Output features. The output features $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ are the binary fingerprint targets of the prediction. They are either substructures, memberships of molecular classes, or physico-chemical properties [243]. A review of various fingerprints and their implementations has been conducted by Steffen et al. [243].

The substructures are denoted as fingerprints: they are functional groups or small chemically meaningful substructures. Examples include the aromatic ring $\mathtt{C_6H_6}$, a phosphate group $\mathtt{PO_4}$, a carbon-carbon double bond $\mathtt{C{=}C}$,

a primary alcohol RCH_2OH, or a phenyl C_6H_5. The membership of molecular classes can be encoded as single substructure fingerprint. For instance, a phenyl is characterised with an existence of a phenyl group.

Pharmacophoric fingerprints model the structures and chemical properties related to the pharmacological action of drug molecules [179]. In practise the models focus on molecular binding including spatial information, charges, hydrophobicity and donor/acceptor relations. Physico-chemical fingerprints model, for instance, lipophilicity, polarizability, charge and other properties [199].

Commonly a set of tens or hundreds of high-quality fingerprints are used. STIRS contained already 600 fingerprints [60]. A standard set of 1365 substructures contains both systematically generated small substructures as well as expert defined functional groups [261, 235]. Another common fingerprint set is the 881-bit fingerprint set standardised by the PubChem database [271]. The usefulness of an individual fingerprint is dependent on its predictability and its discriminatory power with respect to the task at hand.

The fingerprint set is closely related to chemical similarity measures. An ideal fingerprint set can distinguish, e.g. metabolites solely on the fingerprint vector without concerning with the actual structure explicitly. Hence fingerprints provide a prime target for classification for metabolite identification.

Mathematically we wish to find a smallest fingerprint set that maximises the joint entropy of the fingerprint distribution. Often some fingerprints in combination represent the same information. For instance, an aromatic ring fingerprint necessarily implies an carbon-carbon double bond fingerprint. Full information theoretic analysis of joint entropies of a small number of fingerprints have been conducted [158].

In Steffen et al. the performance of different molecular fingerprints are compared for biological profile data [243]. Several studies have compared different fingerprints for virtual screening applications [98, 19] with a critical evaluation of these comparisons by Hawkins et al. [108].

In the machine learning context, the discriminatory properties of the fingerprints are accompanied with the prediction accuracies of said fingerprints. The combined effect of these two factors is an open problem and has been addressed in Paper **V**.

Classifiers. Early work on predicting fingerprints is characterised by K-nearest neighbour algorithms [60, 201], the linear discriminant analysis [266] and rule-based systems [167, 267]. The pioneering work by Breiman et

al. applied a decision tree to the problem of detecting a single chlorine substructure [34]. A high cross-validation accuracy of 95% was reported with a decision tree of 1,500 internal nodes. This line of research has been continued with random forests and boosting combined with both neural networks and decision trees [124, 95, 110, 109, 290]. Neural networks have been a popular choice of classifier due to their high predictive accuracy [58, 72]. Other methods include the partial least squares discriminant analysis (PLS-DA) [292, 289] and SVM with Gaussian kernels on top of direct dot products of the feature vectors [290].

Several studies report comparisons of the classification methods. Boosting improves the classifiers without exception when applied to both neural networks and decision trees [110, 290]. SVM's performance is on par with AdaBoost decision trees in the study by Yu-Xi et al. [290]. The comparison by Werther et al. shows neural networks producing slightly better classifiers than LDA or KNN [277], which is expected as neural networks can handle non-linear features to a degree. KNN was reported to surpass LDA and least squares methods in predicting oxygen content of small hydrocarbons [180]. Feature selection is shown to have a small effect with LDA and PLS-DA [289].

5.5.2 Mass spectral kernels

Tandem mass spectrometry has been used with kernel methods in classification of pesticides [290] and in validation of phosphopeptide identifications [169]. In the former a direct kernelization of the mass spectral features is used, while in the latter 8 features are constructed representing, for instance, the ratio between highest peak and precursor peak, and the average intensity.

A mass spectral kernel should take the peak error ε into account. A soft-matching kernel is a natural choice. We next introduce two such kernel families: a standard soft-matching kernel over the peaks, and a probabilistic probability product kernel.

A soft-matching mass spectral kernel is

$$K(\chi, \chi') = \sum_{\substack{(mass,int)\in\chi \\ (mass',int')\in\chi'}} \kappa_m(mass, mass') \cdot \kappa_i(int, int'),$$

where $\kappa_m(mass, mass')$ is any soft-matching kernel and κ_i is a kernel for intensities. A natural kernel for intensities is the product kernel $\kappa_i(int, int') = int \cdot int'$. The peak masses are aligned with the κ_m. Any choice of a Mercer

kernel is suitable, for instance a Gaussian radial basis function kernel:

$$\kappa_{RBF}(u, v) = \exp\left(-\frac{|u - v|^2}{2\sigma^2}\right),$$

where σ acts as a width parameter.

Probability product kernel

In a seminal work by Kondor and Jebara a density estimation kernel was proposed for sets [149]. They later extended the kernel into a probability product kernel, which allows kernelization of any probabilistic model [128]. The probability product kernel is introduced as a mass spectral kernel in the Paper **V**.

The main idea behind probability kernel is simple, yet powerful. We associate a probabilistic model for each datum, and then define a kernel as a similarity measure of these densities instead of the original data. This is beneficial, for instance, when the objects in question are sets, such as in mass spectra. The far-reaching consequence of the probability kernels is that *any* set of probabilistic models is open for classification and regression through kernel methods.

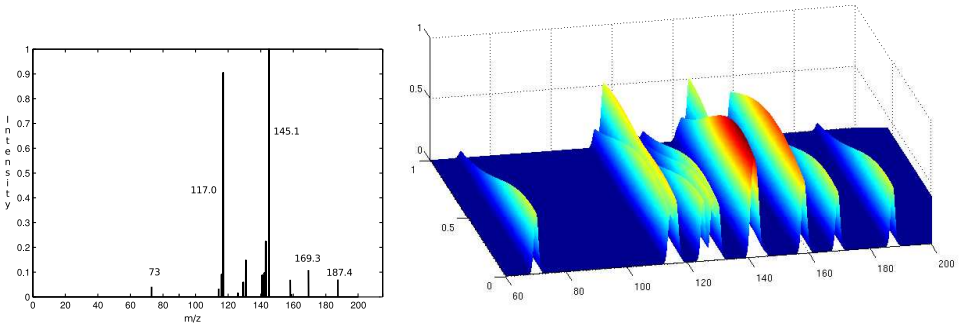
Let the dataset of n objects $\chi \in \mathcal{X}$ be either singleton sets $\chi = \{\mathbf{x} \in \mathbb{R}^d\}$ or sets with several data points $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_k : \mathbf{x}_i \in \mathbb{R}^d\}$, where data points are vectors from $\mathbb{R}^d$. The number of data points k per object χ is allowed to vary. We assume that for each χ there is underlying distribution generating the data points $\{\mathbf{x}_i\}$. We then define a kernel between χ and χ' by estimating corresponding distributions p and p' that represent the underlying distributions. The *probability product kernel* is then between the estimates

$$K_{pp}(\chi, \chi') = K_\beta(p, p') = \int_{\mathbb{R}^d} p(\mathbf{x})^\beta p'(\mathbf{x})^\beta d\mathbf{x} = \langle p^\beta, p'^\beta \rangle_{L_2}.$$

The kernel is positive definite as a dot product. The feature map is a $L_2(\mathcal{X})$ space. However, in practise the data lies on a manifold spanned by the data points with a maximum dimensionality of n.

The kernel has interesting properties for special values of β. For $\beta = 1$ the kernel is the expectation of one distribution under the other, and hence called the *expected likelihood kernel*

$$K_{el}(\chi, \chi') = \int_{\mathbb{R}^d} p(\mathbf{x}) p'(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{p'}[p(\mathbf{x})] = \mathbb{E}_p[p'(\mathbf{x})].$$



(a) A spectrum (b) The kernel density estimation

Figure 5.5: The kernel density estimation representation of a spectrum. A gaussian density is placed at each peak in a two-dimensional space.

Another variant is encountered at $\beta = \frac{1}{2}$ as

$$K_{bh}(\chi, \chi') = \int_{\mathbb{R}^d} \sqrt{p(\mathbf{x})}\sqrt{p'(\mathbf{x})}d\mathbf{x},$$

which is known as the Bhattacharyya's measure of affinity between distributions [2, 24]. We call this kernel the *Bhattacharyya kernel*. The kernel satisfies the normalisation property $K_{bh}(\chi, \chi) = 1$.

The conventional measure of distribution similarity is the Kullback-Leibler divergence

$$D(p||p') = \int_{\mathbb{R}^d} p(\mathbf{x}) \log p(\mathbf{x})d\mathbf{x} - \int_{\mathbb{R}^d} p(\mathbf{x}) \log p'(\mathbf{x})d\mathbf{x},$$

which is not symmetric and hence not a Mercel kernel [127].

Efficient methods to compute the probability product kernel have been introduced for various distributions, such as exponential, Gaussian, Bernoulli, multinomial and gamma distributions over the data χ. Kernels were defined also for situations where the underlying generative distribution is assumed to be a Hidden Markov Model, Bayesian Network or linear Gaussian model [128].

As a specific example we define the kernel computation explicitly for an expected likelihood kernel with a Gaussian mixture model

$$p = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(\mu_i, \Sigma_i)$$

of χ, where an individual data point $\mathbf{x}_i$ of χ is represented with a Gaussian of mean μ_i and covariance Σ_i. With spectral data, a straightforward approach is to center the mean $\mu_i = \mathbf{x}_i$ at the peak (See Figure 5.5), and thus

$$p = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(\mathbf{x}_i, \Sigma_i).$$

The probability product kernel then measures the similarity between two spectra χ and χ' as the similarity between the corresponding probabilistic models p and p':

$$\begin{aligned} K(\chi, \chi') &= K(p, p') \\ &= \int_{\mathbb{R}^d} p(\mathbf{x}) p'(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbb{R}^d} \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(\mathbf{x}_i, \Sigma_i) \cdot \frac{1}{k'}\sum_{j=1}^{k'} \mathcal{N}(\mathbf{x}_j, \Sigma_j) d\mathbf{x} \\ &= \frac{1}{k}\frac{1}{k'}\sum_{i=1}^{k}\sum_{j=1}^{k'} \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}_i, \Sigma_i) \cdot \mathcal{N}(\mathbf{x}_j, \Sigma_j) d\mathbf{x}. \\ &= \frac{1}{k}\frac{1}{k'}\sum_{i=1}^{k}\sum_{j=1}^{k'} \int_{\mathbb{R}^d} z_\dagger \mathcal{N}(\mathbf{x}_\dagger, \Sigma_\dagger) d\mathbf{x}, \end{aligned} \tag{5.1}$$

where $\Sigma_\dagger = (\Sigma_i^{-1} + \Sigma_j^{-1})^{-1}$, $\mathbf{x}_\dagger = \Sigma_\dagger(\Sigma_i^{-1}\mathbf{x}_i + \Sigma_j^{-1}\mathbf{x}_j)$ and

$$z_\dagger = \frac{1}{(2\pi)^{1/2}|\Sigma_i + \Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\Sigma_i + \Sigma_j)^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right).$$

The product of two gaussians is then another, unnormalized gaussian [3]. As the integral of a gaussian distribution is one, we have

$$\int_{\mathbb{R}^d} z_\dagger \mathcal{N}(\mathbf{x}_\dagger, \Sigma_\dagger) d\mathbf{x} = z_\dagger.$$

By plugging this result into 5.1 we have a general solution to the kernel as (See Paper **V**)

$$K(\chi, \chi') = \frac{1}{k}\frac{1}{k'}\sum_{i,j}^{k,k'} \frac{1}{(2\pi)^{1/2}|\Sigma_i + \Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\Sigma_i + \Sigma_j)^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right).$$

Chapter 6

Conclusions

In this thesis algorithmic methods have been presented to four bioinformatics and chemoinformatics problems involving small molecules. In the original publications we have proposed efficient and state-of-the-art computational solutions to these problems. The problems fall under two major themes of atom-level descriptions of organic reactions, and on identification of structures based on tandem mass spectrometric measurements.

We discussed the problem of computational reaction mapping, which belongs to graph matching problems. The de-facto method of using maximum common subgraphs suffers from several practical and theoretical shortcomings. Hence, we introduce formalism to determine the optimality criteria of reaction mappings using graph edit distance, and propose an accompanying A* algorithm to find optimal mappings.

We reviewed the field of graph kernel methods for structured prediction on biological graphs. A large array of graph kernels have been proposed that base their feature representation on enumerating substructures of the molecular graphs. However, a path-based graph kernel has been missing. The introduced path kernel's features combine the simplicity of sequence features and high information content for state-of-the-art classification performance. We also discussed the application of graph kernels to organic reactions, where the reaction mappings can be utilised for a compact reaction graph representation to facilitate machine learning. The reaction graph concept warrants subsequent research, as a more robust alternative, on all fields of metabolomics where reactions are represented as transformations.

We then discussed identification of structures in tandem mass spectrometric measurements. A fragment identification problem was reviewed with applications. In addition to quantum-chemical simulations, the main classes of proposed algorithms are rule-based methods and combinatorial methods. We extend the combinatorial methods to model the dependen-

cies between fragments, resulting in fragmentation trees. A proposed cost function using bond energies provides a formal method to assess feasibility of the proposed fragments.

Finally we discussed computational metabolite identification. The problem is challenging and of high importance in metabolomics studies. We surveyed the main approaches of utilising reference databases, fragment identifications and machine learning for the metabolite identification. The introduced kernel method defines a formal model to account for mass measurement accuracy and provides a state-of-the-art performance of kernel methods to metabolite identification. Further research is warranted on both exploiting fragmentation trees and advancements of structured prediction to metabolite identification.

References

[1] ACD/Labs. ACD/MS Fragmenter. `http://www.acdlabs.com`, 2005.

[2] F. Aherne, N. Thacker, and P. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32:1–7, 1997.

[3] P. Ahrendt. The multivariate gaussian probability distribution. Technical report, Technical University of Denmark, 2005.

[4] J. Aichelin, G. Peilert, A. Bohnet, A. Rosenhauer, H. Stöcker, and W. Greiner. Quantum molecular dynamics approach to heavy ion collisions: Description of the model, comparison with fragmentation data, and the mechanism of fragment formation. *Physical Review*, 37:2451–2468, 1988.

[5] T. Akutsu. Efficient extraction of mapping rules of atoms from enzymatic reaction data. *Journal of Computational Biology*, 11:449–462, 2004.

[6] A. Alex, S. Harvey, T. Parsons, F.S. Pullen, P. Wright, and J-A. Riley. Can density functional theory (DFT) be used as an aid to a deeper understanding of tandem mass spectrometric fragmentation pathways? *Rapid Communications in Mass Spectrometry*, 23:2619–2627, 2009.

[7] H. Almohamed. A linear programming approach for the weighted graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:522–525, 1993.

[8] J. Apostolakis, O. Sacher, R. Korner, and J. Gasteiger. Automatic determination of reaction mappings and reaction center information. 2. Validation on a biochemical reaction database. *Journal of Chemical Information and Modeling*, 48:1190–1198, 2008.

[9] M. Arita. Graph modeling of metabolism. *Journal of Japan Society of Artificial Intelligence*, 15:703–710, 2000.

[10] M. Arita. In silico atomic tracing of substrate-product relationships in *Escherichia coli* intermediary metabolism. *Genome Research*, 13(11):2455–2466, 2003.

[11] M. Arita. Introduction to the ARM database: Database on chemical transformation in metabolism for tracing pathways. In *Metabolomics, The Frontier of Systems Biology*, volume 4, pages 193–210. Springer Tokyo, 2005.

[12] K. Astikainen, L. Holm, E. Pitkänen, S. Szedmak, and J. Rousu. Structured output prediction of novel enzyme function with reaction kernels. In *Biomedical Engineering Systems and Technologies*, pages 367–378. Springer, 2011.

[13] K. Astikainen, E. Pitkänen, J. Rousu, L. Holm, and S. Szedmak. Reaction kernels: Structured output prediction approaches for novel enzyme function. In *BIOINFORMATICS*. Valencia, Spain, 2010.

[14] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65:21–46, 1996.

[15] G. Bakir. *Predicting Structured Data*. MIT Press, 2007.

[16] V. Balaz, V. Kvasnicka, and J. Pospichal. Two metrics in a graph theory modeling of organic chemistry. *Discrete Applied Mathmematics*, 35:1–19, 1992.

[17] M. Bandu, J. Wilson, R. Vachet, D. Dalpathado, and H. Desaire. STEP (statistical test of equivalent pathways) analysis: A mass spectrometric method for carbohydrates and peptides. *Analytical Chemistry*, 77:5886–5893, 2005.

[18] P. Baranczewski, A. Stańczak, A. Kautiainen, P. Sandin, and Per-Olof Edlund. Introduction to early *in vitro* identification of metabolites of new chemical entities in drug discovery and development. *Pharmacological Reports*, 58:341–352, 2006.

[19] A. Bender and R. Glen. A discussion of measures of enrichment in virtual screening: Comparing the information content of descriptors with increasing levels of sophistication. *Journal of Chemical Information and Modeling*, 45:1369–1375, 2005.

[20] S. Berretti, A. Del Bimbo, and E. Vicario. The computational aspect of retrieval by spatial arrangement. In *Proceedings of the International Conference on Pattern Recognition*, pages 1047–1051, 2000.

[21] S. Berretti, A. Del Bimbo, and E. Vicario. A look-ahead strategy for graph matching in retrieval by spatial arrangement. *International Conference on Multimedia and Expo*, pages 1721–1724, 2000.

[22] S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1089–1105, 2001.

[23] R. Bettens and A. Lee. A new algorithm for molecular fragmentation in quantum chemical calculations. *Journal of Physical Chemistry A*, 110:8777–8785, 2006.

[24] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of Calcutta Mathematical Society*, 1943.

[25] S. Blanskby and G. Ellison. Bond dissociation energies of organic molecules. *Accounts of chemical research*, 36:255–263, 2003.

[26] T. Blum and O. Kohlbacher. Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. *Journal of Computational Biology*, 15:565–576, 2008.

[27] S. Böcker, M. Letzel, Z.S. Liptak, and A. Pervukhin. Sirius: decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25:218–224, 2009.

[28] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[29] K. Borgwardt and H-P. Kriegel. Shortest-path kernels on graphs. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-2005)*. Houston, USA, 2005.

[30] K. Borgwardt, C. Ong, S. Schönauer, S. Vishwanathan, A. Smola, and H-P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56, 2005.

[31] K. Borgwardt, S. Vishwanathan, N. Schraudolph, and H-P. Kriegel. Graph kernels for disease outcome prediction from protein-protein interaction networks. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 4–15, 2007.

[32] L. Borland, M. Brickhouse, T. Thomas, and A. Fountain. Review of chemical signature databases. *Analytical and Bioanalytical Chemistry*, 397:1019–1028, 2010.

[33] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[34] L. Breiman, J. Friedman, C. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[35] A. Brint and P. Willett. Algorithms for the identification of three-dimensional maximal common substructures. *Journal of Chemical Informatics in Computer Science*, 2:311–320, 1987.

[36] T. Brodmeier, A. Gloor, M. Cadisch, R. Bürgin, and E. Pretsch. Hypermedia tools for the interpretation of mass spectra. *Analytica Chimica Acta*, 277:297–304, 1993.

[37] C. Bron and J. Kerbosch. Finding all cliques of an undirected graph. *Communications of the ACM*, 16:575–577, 1971.

[38] M. Brown, W.B. Dunn, P. Dobson, Y. Patel, C.L. Winder, S. Francis-McIntyre, P. Begley, K. Carroll, D. Broadhurst, A. Tseng, N. Swainston, I. Spasic, R. Goodacre, and D.B. Kell. Mass spectrometry tools and metabolite-specific databases for molecular identification in metabolomics. *Analyst*, 134:1322–1332, 2009.

[39] R. D. Brown, G. Jones, P. Willett, and R. Glen. Matching two-dimensional chemical graphs using genetic algorithms. *Journal of Chemical Information and Modeling*, 34:63–70, 1994.

[40] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18:6689–694, 1997.

[41] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922, 1999.

[42] H. Bunke. Graph matching: Theoretical foundations, algorithms, and applications. In *Proceedings of Vision Interface*, pages 82–88, 2000.

[43] S. Böcker and F. Rasche. Towards *de novo* identification of metabolites by analyzing tandem mass spectra. In *Bioinformatics 24, ECCB*, pages T49–T55, 2008.

[44] S. Böcker, F. Rasche, and T. Steijger. Annotating fragmentation patterns. In *Proceedings of the WABI*, volume 5724 of *Lecture Notes in Bioinformatics*, pages 13–24. Springer, 2009.

[45] R. Caspi, T. Altman, K. Dreher, C. A. Fulcher, P. Subhraveti, I. M. Keseler, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, A. Pujar, A. G. Shearer, M. Travers, D. Weerasinghe, P. Zhang, and P. D. Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 40:D742–D753, 2012.

[46] F. Cazals and C. Karande. An algorithm for reporting maximal c-cliques. *Theoretical Computer Science*, 349:484–490, 2005.

[47] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407:564–568, 2008.

[48] A. Ceroni, F. Costa, and P. Frasconi. Classification of small molecules by two- and three-dimensional decomposition kernels. *Bioinformatics*, 23:2038–2045, 2007.

[49] C. Chen and D. Yun. Discovering process models from execution history by graph matching. In *International Conference on Systems, Signals, Control, Computers*, 1998.

[50] W. J. Christmas, J. Kittler, and M. Petrou. Sturctural matching in computer vision using probabilistic relaxation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 17:749–764, 1995.

[51] M. Cone, R. Venkataraghaven, and F. McLafferty. Molecular structure comparison program for the identification of maximal common substructures. *Journal of the American Chemical Society*, 99:7668–7671, 1977.

[52] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition & Artificial Intelligence*, 18:265–298, 2004.

[53] K. Coombes, J. Koomen, K. Baggerly, J. Morris, and R. Kobayashi. Understanding the characteristics of mass spectrometry data through the use of simulation. *Cancer Informatics*, 1:41–52, 2005.

[54] L. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomoprhism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1367–1372, 2004.

[55] J.D. Crabtree and D.P. Mehta. Automatic reaction mapping. *ACM Journal of Experimental Algorithms*, 13:15, 2009.

[56] A. Cross, R. Wilson, and E. Hancock. Genetic search for structural matching. In *Computer Vision - ECCV*, volume 1064 of *Lecture Notes in Computer Science*, pages 514–525. Springer, 1996.

[57] B. Cuissart and J-J. Hebrard. A direct algorithm to find a largest common connected induced subgraph of two graphs. In *Graph-Based Representations in Pattern Recognition*, volume 3434 of *Lecture Notes in Computer Science*, page 133. Springer-Verlag Berlin Heidelberg, 2005.

[58] B. Curry and D. Rumelhart. MSnet: A neural network that classifies mass spectra. *Tetrahedron Computer Methodology*, 3:213–237, 1990.

[59] G. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5:266–277, 1957.

[60] H. Dayringer, G. Pesyna, R. Venkataraghavan, and F. McLafferty. Computer-aided interpretation of mass spectra. Information on substructural probabilities form STIRS. *Organic Mass Spectrometry*, 11:529–542, 1976.

[61] A. Demco. *Graph Kernel Extension and Experiments with Application to Molecule Classification, Lead Hopping and Multiple Targets.* PhD thesis, University of Southampton, 2009.

[62] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series*, B39:1–38, 1977.

[63] W. Demuth, M. Karlovits, and K. Varmuza. Spectral similarity versus structural similarity: Mass spectrometry. *Analytica Chimica Acta*, 516:75–85, 2004.

[64] M. Deshpande. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engingeering*, 17:1036–1050, 2005.

[65] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[66] R. Dondi, G. Fertin, and S. Vialette. Maximum motif problem in vertex-colored graphs. In *CPM*, volume 5577 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2009.

[67] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley, 1973.

[68] J. Dugundji and I. Ugi. An algebraic model of constitutional chemistry as a basis for chemical computer programs. *Topics in Current Chemistry*, 39:19–64, 1973.

[69] A. C. M. Dumay, R. J. van der Geest, J. J. Gerbrands, E. Jansen, and J. H. C. Reiber. Consistent inexact graph matching applied to labeling coronary segments in arteriograms. In *Proceedings of the International Conference on Pattern Recognition*, pages 439–442, 1992.

[70] P. Durand, R. Pasari, J. Baker, and C. Tsai. An efficient algorithm for similarity analysis of molecules. *Internet journal of Chemistry*, 2:1, 1999.

[71] J.P. Dworzanski, A.P. Snyder, R. Chen, H. Zhang, D. Wishart, and L. Li. Identification of bacteria using tandem mass spectrometry combined with a proteome database and statistical scoring. *Analytical Chemistry*, 76:2355–2366, 2004.

[72] A. Eghbaldar, T. Forrest, and D. Cabrol-Bass. Development of neural networks for identification of structural features from mass spectral data. *Analytica Chimica Acta*, 359:283–301, 1998.

[73] M. A. Eshera and K. S. Fu. A graph distance measure for image analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 14:398–408, 1984.

[74] M. A. Eshera and K. S. Fu. A similarity measure between attributed relational graphs for image analysis. In *Proceedings of the International Conference on Pattern Recognition*, pages 75–77, 1984.

[75] H. Felix, F. Rossello, and G. Valiente. Artificial chemistries and metabolic pathways. In *Proceedings of the Annual Spanish Bioinformatics Conference*, pages 56–59, 2004.

[76] H. Felix, F. Rossello, and G. Valiente. Optimal artificial chemistries and metabolic pathways. In *Proceedings of the Mexican International Conference on Computer Science*, pages 298–305. IEEE Computer Science Press, 2005.

[77] L. Felix and G. Valiente. Efficient validation of metabolic pathway databases. In *Proceedings of the International Symposium on Computational Biology and Genome Informatics*, pages 1209–1212, 2005.

[78] J. Feng, M. Laumy, and M. Dhome. Inexact matching using neural networks. In *Pattern Recognition in Practise IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, pages 177–184, 1994.

[79] M-L. Fernandez and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22:753–758, 2001.

[80] P. Ferragina and G. Manzini. An experimental study of a compressed index. *Information Sciences*, 135:13–28, 2001.

[81] J. Ferreira and F. Couto. Semantic similarity for automatic classification of chemical compounds. *PLoS Computational Biology*, 6, 2010.

[82] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computing*, 22:67–92, 1973.

[83] J. Frey. Using InChI. *Chemistry International*, pages 14–15, 2006.

[84] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[85] H. Fröhlich, J. Wegner, F. Sieker, and A. Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of ICML*, 2005.

[86] H. Fröhlich, A. Kosir, and B. Zajc. Optimization of FPGA configurations using parallel genetic algorithm. *Information Sciences*, 133:195–219, 2001.

[87] J. Gao, S. Ma, D.T. Major, K. Nam, J. Pu, and D.G. Truhlar. Mechanisms and free energies of enzymatic reactions. *Chemical Review*, 106:3188–3209, 2006.

[88] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis & Applications*, 13:113–129, 2010.

[89] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[90] T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5:49–58, 2003.

[91] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines, 16th Annual Conferences on Learning Theory and 7th Kernel Workshop*, volume 2843, pages 129–143. Springer-Verlag, 2003.

[92] J. Gasteiger and T. Engel, editors. *Chemoinformatics: A Textbook*. Wiley, 2003.

[93] J. Gasteiger, W. Hanebeck, and K-P. Schulz. Prediction of mass spectra from structural information. *Journal of Chemical Information and Computer Sciences*, 32:264–271, 1992.

[94] L. Gerhards and W. Lindenberg. Clique detection for nondirected graphs: Two new algorithms. *Computing*, 21:295–322, 1979.

[95] P. Geurts, M. Fillet, D. de Seny, M-A. Meuwis, M. Malaise, M-P. Merville, and L. Wehenkel. Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics*, 21:3138–3145, 2005.

[96] D. E. Gharaman, A. K. C. Wong, and T. Au. Graph optimal monomorphism algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 10:181–188, 1980.

[97] R. Giegerich and S. Kurtz. From Ukkonen to McCreight and Weiner: A unifying view of linear-time suffix tree construction. *Algorithmica*, 19:331–353, 1997.

[98] J. Godden, F. Stahura, and J. Bajorath. Anatomy of fingerprint search calculations on structurally diverse sets of active compounds. *Journal of Chemical Information and Modeling*, 45:1812–1819, 2005.

[99] R. Goodacre, S. Vaidyanathan, W. B. Dunn, G. G. Harrigan, and D. B. Kell. Metabolomics by numbers: Acquiring and understanding global metabolite data. *Trends in Biotechnology*, 22:245–252, 2004.

[100] J. Gower and P. Legendre. Metric and euclidian properties of dissimilaity coefficients. *Journal of Classification*, 3:5–48, 1998.

[101] L. Gregory and J. Kittler. Using graph search techniques for contextual colour retrieval. In *Proceedings of the Joint IAPR International Workshops SSPR and SPR*, pages 186–194, 2002.

[102] R. Grossman, D. Hamelberg, P Kasturi, and B. Liu. Experimental studies of the universal chemical key (UCK) algorithm on the NCI database of chemical compounds. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB)*, pages 244–250, 2003.

[103] R. Hall, M. Beale, O. Fiehn, N. Hardy, L. Summer, and R. Bino. Plant metabolomics: The missing link in functional genomics strategies. *The Plant Cell*, 14:1437–1440, 2002.

[104] S. Hammes-Schiffer and S. Benkovic. Relating protein motion to catalysis. *Annual Review of Biochemistry*, 75:519–541, 2006.

[105] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125:11853–65, 2003.

[106] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa. Heuristics for chemical compound matching. *Genome Informatics*, 14:144–153, 2003.

[107] D. Haussler. Convolution kernels on discrete structures. Technical report, University of California, Santa Cruz, 1999. UCSC-CRL-99-10.

[108] P. Hawkins, G. Warren, G. Skillman, and A. Nicholls. How to do an evaluation: Pitfalls and traps. *Journal of Computer-Aided Molecular Design*, 22:179–190, 2008.

[109] P. He, K-T. Fang, Y-Z. Liang, and B-Y. Li. A generalized boosting algorithm and its application to two-class chemical classification problem. *Analytica Chimica Acta*, 543:181–191, 2005.

[110] P. He, C-J. Xu, Y-Z. Liang, and K-T. Fang. Improving the classification accuracy in chemistry via boosting technique. *Chemometrics and intelligent laboratory systems*, 70:39–46, 2004.

[111] M. Heinonen, S. Lappalainen, T. Mielikäinen, and J. Rousu. Computing atom mappings for biochemical reactions without subgraph isomorphism. *Journal of Computational Biology*, 18:43–58, 2011.

[112] M. Heinonen, A. Rantanen, T. Mielikäinen, T. Kokkonen, J. Kiuru, R.A. Ketola, and J. Rousu. Fid: a software for *ab initio* structural identification of product ions from tandem mass spectrometric data. *Rapid Communications in Mass Spectrometry*, 22:3043–3052, 2008.

[113] M. Heinonen, A. Rantanen, T. Mielikäinen, E. Pitkänen, J. Kokkonen, and J. Rousu. *Ab Initio* prediction of molecular fragments from tandem mass spectrometry data. In *German Conference on Bioinformatics*, volume P-83 of *Lecture Notes in Informatics (LNI)*, pages 40–53. GI, 2006.

[114] M. Heinonen, N. Välimäki, V. Mäkinen, and J. Rousu. Efficient path kernels for reaction function prediction. In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms*, pages 202–207, 2012.

[115] HighChem. HighChem Mass Frontier 4.0. `http://www.highchem.com`, 2005.

[116] A. Hill and R. Mortishire-Smith. Automated assignment of high-resolution collisionally activated dissociation mass spectra using a systematic bond disconnection approach. *Rapid Communications in Mass Spectrometry*, 19:3111–3118, 2005.

[117] G. Hinselmann, N. Fechner, A. Jahn, M. Eckert, and A. Zell. Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments. *Neurocomputing*, 74:219–229, 2010.

[118] T. Hogiri, C. Furusawa, Y. Shinfuku, N. Ono, and H. Shimizu. Analysis of metabolic network based on conservation of molecular structure. *BioSystems*, 95:175–178, 2008.

[119] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima, Y. Oda, Y. Kakazu, M. Kusano, T. Tohge, F. Matsuda, Y. Sawada, M. Y. Hirai, H. Nakanishi, K. Ikeda, N. Akimoto, T. Maoka, H. Takahashi, T. Ara, N. Sakurai, H. Suzuki, D. Shibata, S. Neumann, T. Iida, K. Tanaka, K. Funatsu, F. Matsuura, T. Soga, R. Taguchi, K. Saito, and T. Nishioka.

Massbank: A public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry*, 45:703–714, 2010.

[120] H. Horai, M. Arita, and T. Nishioka. Comparison of ESI-MS spectra in massbank database. In *Proceedings of the International Conference on BioMedical Engineering and Informatics*, pages 853–857, 2008.

[121] C. Hsu. Diophantine approach to isotopic abundance calculations. *Analytical Chemistry*, 56:1356–1361, 1984.

[122] B. Huet and E. R. Hancock. Shape recognition from large image libraries by inexact graph matching. *Pattern Recognition Letters*, 20:1259–1269, 1999.

[123] F. Hufsky, M. Rempt, F. Rasche, G. Pohnert, and S. Böcker. *De novo* analysis of electron impact mass spectra using fragmentation trees. *Analytica Chimica Acta*, 739:67–76, 2012.

[124] J. Hummel, N. Strehmel, J. Selbig, D. Walther, and J. Kopka. Decision tree supported substructure prediction of metabolites from GC-MS profiles. *Metabolomics*, 6:322–333, 2010.

[125] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999.

[126] C. James, D. Weininger, and J. Delany. Daylight theory manual, SMARTS theory. Technical report, Daylight, 2000.

[127] T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. *Learning theory and kernel machines*, 2777:57–71, 2003.

[128] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[129] J. Jeong, X. Shi, X. Zhang, S. Kim, and C. Shen. An empirical Bayes model using a competetion score for metabolite identification in gas chromatography mass spectrometry. *BMC Bioinformatics*, 12:392, 2011.

[130] C. Jochum, J. Gasteiger, and I. Ugi. The principle of minimal chemical distance (pmcd). *Angewandte Chemie International Edition*, 19:495–505, 1980.

[131] C. Jochum, J. Gasteiger, I. Ugi, and J. Dugundji. The principle of minimal chemical distance and the principle of minimum structure change. *Zeitschrift für Naturforschung*, 37B:1205–1215, 1982.

[132] I. Joliffe. *Principal Component Analysis*. Springer, 1986.

[133] D. Justice and A. Hero. A binary linear programming formulation of the graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1200–1214, 2006.

[134] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28:27–30, 2000.

[135] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 321–328, 2003.

[136] Douglas Kell. Metabolomics and systems biology: Making sense of the soup. *Current Opinion in Microbiology*, 7:296–307, 2004.

[137] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Heidelberg, 2004.

[138] A. Kerber, R. Laue, M. Meringer, and K. Varmuza. MOLGEN-MS: Evaluation of low resolution electron impact mass spectra with MS classification and exhaustive structure generation. *Advances in Mass Spectrometry*, 15:939–940, 2001.

[139] A. Kerber, M. Meringer, and C. Rücker. CASE via MS: Ranking structure candidates by mass spectra. *Croatica Chemica Acta*, 79:449–464, 2006.

[140] T. Kind and O. Fiehn. Metabolomic database annotations via query of elemental compositions: Mass accuracy is insufficient even at less than 1 ppm. *BMC Bioinformatics*, 7:234–244, 2006.

[141] T. Kind and O. Fiehn. Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*, 8:105–125, 2007.

[142] T. Kind and O. Fiend. Advances in structure elucidation of small molecules using mass spectrometry. *Bioanalytical Review*, 2:23–60, 2010.

[143] J. Kittler and E. R. Hancock. Combining evidence in probabilistic relaxation. *International Journal of Pattern Recognition and Artificial Intelligence*, 3:29–51, 1989.

[144] M. Kiyomi. *Studies on Subgraph and Supergraph Enumeration Algorithms*. PhD thesis, Department of Informatics, The Graduate University for Advanced Studies, Japan, 2006.

[145] K. Klagkou, F. Pullen, M. Harrison, A. Organ, A. Firth, and J. Langley. Approaches towards the automated interpretation and prediction of electrospray tandem mass spectra of non-peptidic combinatorial compounds. *Rapid Communications in Mass Spectrometry*, 17:1163–1168, 2003.

[146] D. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 3*. Addison-Wesley, 2005.

[147] J. Köbler, U. Schöning, and J. Toran. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhauser Verlag, Switzerland, 1994.

[148] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250:1–30, 2001.

[149] R. Kondor and T. Jebara. A kernel between sets of vectors. In *ICML*, 2003.

[150] R. Korner and J. Apostolakis. Automatic determination of reaction mappings and reaction center information. 1. The imaginary transition state energy approach. *Journal of Chemical Information and Modeling*, 48:1181–1189, 2008.

[151] M. Kotera, M. Hattori, M-A. Oh, R. Yamamoto, T. Komeno, J. Yabuzaki, K. Tonomura, S. Goto, and M. Kanehisa. RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Informatics*, 15:P062, 2004.

[152] M. Kotera, Y. Okuno, M. Hattori, S. Goto, and M. Kanehisa. Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *Journal of the American Chemical Society*, 126:16487–16498, 2004.

[153] N. Kriege and P. Mutzel. Subgraph matching kernels for attributed graphs. In *Proceedings of ICML*, pages 1015–1022, 2012.

[154] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2:83–97, 1955.

[155] V. Kvasnicka and J. Pospichal. Maximal common subgraphs of molecular graphs. *Reports in Molecular Theory*, 1, 1990.

[156] V. Kvasnicka and J. Pospichal. Chemical and reaction metrics for graph-theoretical model of organic chemistry. *Journal of Molecular Structure*, 73:17–42, 1991.

[157] G. Lanckriet, M. Deng, N. Christianini, M. Jordan, and W. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 9, pages 300–311, 2004.

[158] M. Landon and S. Schaus. JEDA: Joint entropy diversity analysis. An information-theoretic method for choosing diverse and representative subsets from combinatorial libraries. *Molecular Diversity*, 10:333–339, 2006.

[159] R. Laue, T. Grüner, M. Meringer, and A. Kerber. Constrained generation of molecular graphs. *DIMACS Series in Discrete Mathematics And Theoretical Computer Science*, 69:319–332, 2005.

[160] A. Leach and V. Gillet. *An Introduction to Chemoinformatics*. Kluwer Academic Publishers, 2003.

[161] K. Lebedev and D. Cabrol-Bass. New computer aided methods for revealing structural features of unknown compounds using low resolution mass spectra. *Journal of Chemical Information and Computer Sciences*, 38:410–419, 1998.

[162] M. Leber, V. Egelhofer, I. Schomburg, and D. Schomburg. Automatic assigment of reaction operators to enzymatic reactions. *Bioinformatics*, 25:3135–3142, 2009.

[163] C. Leslie, E. Eskin, and S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[164] G. Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9:1–12, 1972.

[165] K. Levsen, H-M. Schiebel, J. Terlouw, K. Jobst, M. Elend, A. Preiss, H. Thiele, and A. Ingendoh. Even-electron ions: a systematic study of the neutral species lost in the dissociation of quasi-molecular ions. *Journal of Mass Spectrometry*, 42:1024–1044, 2007.

[166] X. Lewell, D. Judd, S. Watson, and M. Hann. RECAP - retrosynthetic combinatorial analysis procedure: A powerful new technique

for identifying privileged molecular fragments with useful applications in combinatorical chemistry. *Journal of Chemical Information and Computer Sciences*, 38:511–522, 1998.

[167] R. Lindsay, B. Buchanan, E. Feigenbaum, and J. Lederberg. DENDRAL: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61:209–261, 1993.

[168] H. Lohninger. Feature selection using growing neural networks: The recognition of quinoline derivatives from mass spectral data. *Software Development in Chemistry*, 7:25–37, 1993.

[169] B. Lu, C. Ruse, T. Xu, S. Park, and J. Yates III. Automatic validation of phosphopeptide identifications from tandem mass spectra. *Analytical Chemistry*, 79:1301–1310, 2007.

[170] E. Luks. Isomorphism of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25:42–65, 1982.

[171] B. Luo and E. R. Hancock. Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 23:1120–1136, 2001.

[172] M. Lynch. Storage and retrieval of information on chemical structures by computer. *Endeavour*, 27:68–73, 1968.

[173] M. Lynch and P. Willett. The automatic detection of chemical reaction sites. *Journal of Chemical Information and Computer Sciences*, 18:154–159, 1978.

[174] P. Mahe, N. Ueda, T. Akutsu, J-L. Perret, and J-P. Vert. Extensions of marginalized graph kernels. In *Proceedings of the 21st International Conference on Machine Learning (ICML2004)*. Banff, Canada, 2004.

[175] P. Mahe, N. Ueda, T. Akutsu, J-L. Perret, and J-P. Vert. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *Journal of Chemical Information and Modeling*, 45:939–951, 2005.

[176] U. Manber and G. Myers. Suffix arrays: A new method for online string searches. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 319–327, 1990.

[177] J. Mariethoz and S. Bengio. A max kernel for text-independent speaker verification systems. In *Second Workshop on Multimodal User Authentication, MMUA*, 2006.

[178] M. Marull and B. Rochat. Fragmentation study of imatinib and characterization of new imatinib metabolites by liquid chromatography-triple-quadrupole and linear ion trap mass spectrometers. *Journal of Mass Spectrometry*, 41:390–404, 2006.

[179] J. Mason, I. Morize, P. Menard, D. Cheney, C. Hulme, and R. Labaudiniere. New 4-point pharmacophore method for molecular similarity and diversity applications: Overview of the method and applications, including a novel approach to the design of combinatorial libraries containing priviliged substructures. *Journal of Medicinal Chemistry*, 42:3251–3264, 1999.

[180] J. McGill and B. Kowalski. Classification of mass spectra via pattern recognition. *Journal of Chemical Information and Modeling*, 18:52–55, 1978.

[181] J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, 12:23, 1982.

[182] J. McGregor and P. Willett. Use of a maximal common subgraph algorithm in the automatic identification of the ostensible bond changes occurring in chemical reactions. *Journal of Chemical Information and Computer Sciences*, 21:137–140, 1981.

[183] B. McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 26:306–324, 1998.

[184] F. McLafferty and F. Turecek. *Interpretation of mass spectra*. University Science Books, 4rd edition, 1993.

[185] A. McNaught. The IUPAC international chemical identifier: InChI – A new standard for molecular informatics. *Chemistry International*, pages 12–14, 2006.

[186] S. Menchetti, F. Costa, and P. Frasconi. Weighted decomposition kernels. In *Proceedings of the 22nd international conference on Machine learning*, pages 585–592, 2005.

[187] B. Menküc, C. Gille, and H. Holzhütter. Computer aided optimization of carbon atom labeling for tracer experiments. *Genome Informatics*, 20:270–276, 2008.

[188] F. Mu, C. Unkefer, P. Unkefer, and W. Hlavacek. Prediction of metabolic reactions based on atomic and molecular properties of small-molecule compounds. *Bioinformatics*, 27:1537–1545, 2011.

[189] F. Mu, R. Williams, C. Unkefer, P. Unkefer, J. Faeder, and W. Hlavacek. Carbon-fate maps for metabolic reactions. *Bioinformatics*, 23:3193–3199, 2007.

[190] R. Myers, R. C. Wilson, and E. R. Hancock. Bayesian graph edit distance. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 22:628–635, 2000.

[191] R. Mylonas, Y. Mauron, A. Masselot, P-A. Binz, N. Budin, M. Fathi, V. Viette, D. Hochstrasser, and F. Lisacek. X-Rank: A robust algorithm for small molecule identification using tandem mass spectrometry. *Journal of Mass Spectrometry*, 44:494–502, 2009.

[192] G. Neglur, R. Grossman, B. Liu, B. Ludäscher, and L. Raschid. Assigning unique keys to chemical compounds for data integration: Some interesting counter examples. In *Data Integration in Life Sciences*, volume 3615 of *Lecture Notes in Computer Science*, page 735. Springer-Verlag Berlin Heidelberg, 2005.

[193] M. Neuhaus and Bunke. H. A probabilistic approach to learning costs for graph edit distance. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 389–393, 2004.

[194] S. Neumann and Böcker. Computational mass spectrometry for metabolomics: Identification of metabolites and small molecules. *Analytical and Bioanalytical Chemistry*, 398:2779–2788, 2010.

[195] V. Nicholson, C. Tsai, M. Johnson, and M. Naim. A subgraph isomorphism theorem for molecular graphs. *Graph Theory and Topology in Chemistry*, page 226, 1987.

[196] S. Nordholm and S. Rice. A quantum ergodic theory approach to unimolecular fragmentation. *Journal of Chemical Physics*, 62:157–168, 1975.

[197] V. Nydl. Graph reconstruction from subgraphs. *Discrete Mathematics*, 235:335–341, 2001.

[198] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H.C. Köfeler. On the instrument and the inter-laboratory transferability of a tandem mass spectral reference library: 2. Optimization and characterization of the search algorithm. *Journal of Mass Spectrometry*, 44:494–502, 2009.

[199] T. Oprea and J. Gottfries. Chemography: The art of navigating in chemical space. *Journal of Combinatorial Chemistry*, 3:157–166, 2001.

[200] P. Pardalos. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[201] M. Passlack and W. Bremser. IDIOTS - A structure-oriented data bank system for the identification and interpretation of infrared spectra. *Computer-Supported Spectroscopic Databases*, pages 92–116, 1986.

[202] M. Pavlic, K. Libiseller, and H. Oberacher. Combined use of ESI-QqTOF-MS and ESI-QqTOF-MS/MS with mass-spectral library search for qualittative analysis of drugs. *Analytical and Bioanalytical Chemistry*, 386:69–82, 2006.

[203] S. Peterman, N. Duczak, A. Kalgutkar, M. Lame, and J. Soglia. Application of a linear ion trap/orbitrap mass spectrometer in metabolite characterization studies: Examination of the human liver microsomal metabolism of the non-tricyclic anti-depressant nefazodone using data-dependent accurate mass measurements. *Journal of American Society for Mass Spectrometry*, 17:363–375, 2006.

[204] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[205] L. Ralaivola, S.J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18:1093–1110, 2005.

[206] J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

[207] A. Rantanen, H. Maaheimo, E. Pitkänen, J. Rousu, and E. Ukkonen. Equivalence of metabolite fragments and flow analysis of isotopomer distributions for flux estimation. In *Transactions on Computational Systems Biology, Vol. 1, Lecture Notes in Bioinformatics*, volume 4220, pages 198–220, 2006.

[208] A. Rantanen, J. Rousu, P. Jouhten, N. Zamboni, H. Maaheimo, and E. Ukkonen. An analytic and systematic framework for estimating metabolic flux ratios from 13C tracer experiments. *BMC Bioinformatics*, 9:266–285, 2008.

[209] Ari Rantanen, Taneli Mielikäinen, Juho Rousu, Hannu Maaheimo, and Esko Ukkonen. Planning optimal measurements of isotopomer distributions for estimation of metabolic fluxes. *Bioinformatics*, 22(10):1198–1206, 2006.

[210] F. Rasche, K. Scheubert, F. Hufsky, T. Zichner, M. Kai, A. Svatos, and S. Böcker. Identifying the unknowns by aligning fragmentation trees. *Analytical Chemistry*, 84:3417–3426, 2012.

[211] F. Rasche, A. Svatos, R. Maddula, C. Böttcher, and S. Böcker. Computing fragmentation trees from tandem mass spectrometry data. *Analytical Chemistry*, 83:1243–1251, 2011.

[212] I. Rauf, F. Rasche, F. Nicolas, and S. Böcker. Finding maximum colorful subtrees in practice. In *Research in Computational Molecular Biology*, volume 7262 of *Lecture Notes in Computer Science*, pages 213–223. Springer Berlin, 2012.

[213] J. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16:521–533, 2002.

[214] T. Reemtsma. Determination of molecular formulas of natural organic matter molecules by (ultra-)high-resolution mass spectrometry: status and needs. *Journal of Chromatography A*, 1215:3687–3701, 2009.

[215] A. Rockwood, J. Van Orman, and D. Dearden. Isotopic compositions and accurate masses of single isotopic peaks. *Journal of American Society of Mass Spectrometry*, 15:12–21, 2004.

[216] F. Rogalewicz, Y. Hoppilard, and G. Ohanessian. Fragmentation mechanisms of α-amino acids protonated under electrospray ionization: A collision activation and ab initio theoretical study. *International Journal of Mass Spectrometry*, 195/196:565–590, 2000.

[217] F. Rossello and G. Valiente. Analysis of metabolic pathways by graph transformation. In *Proceedings of the International Conference on Graph Transformation*, volume 3256 of *Lecture Notes in Computer Science*, pages 70–82, 2004.

[218] F. Rossello and G. Valiente. Chemical graphs, chemical reaction graphs, and chemical graph transformation. In *Proceedings of International Workshop on Graph-Based Tools*, volume 127, pages 157–166, 2005.

[219] J. Rousu, A. Rantanen, H. Maaheimo, E. Pitkänen, K. Saarela, and E. Ukkonen. A method for estimating metabolic fluxes from incomplete isotopomer information. In *Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 88–103. Springer, 2003.

[220] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006.

[221] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Efficient algorithms for max-margin structured classification. In *Predicting Structured Data*, pages 105–129. MIT Press, 2007.

[222] C. Rücker and G. Rücker. On finding nonisomorphic connected subgraphs and distinct molecular substructures. *Journal of Chemical Information and Computer Science*, 41:314–320, 2001.

[223] G. Rücker and C. Rücker. Automatic enumeration of all connected subgraphs. *MATCH Communications in Mathematical and in Computer Chemistry*, 41:145–149, 2000.

[224] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Letters to Nature*, 323:533–536, 1986.

[225] M. Rupp and G. Schneider. Graph kernels for molecular similarity. *Molecular Informatics*, 29:266–273, 2010.

[226] M. Rupp, P. Schneider, and G. Schneider. Distance phenomena in high-dimensional chemical descriptor spaces: Consequences for similarity-based approaches. *Journal of Computational Chemistry*, 30:2285–2296, 2009.

[227] H. Saigo, M. Hattori, H. Kashima, and K. Tsuda. Reaction graph kernels predict EC numbers of unknown enzymatic reactions in plant secondary metabolism. *BMC Bioinformatics*, 11:S31, 2010.

[228] K. Scheubert, F. Hufsky, F. Rasche, and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. In *Research in Computational Molecular Biology*, volume 6577 of *Lecture Notes in Computer Science*, pages 377–391. Springer Berlin, 2011.

[229] B. Schölkopf, A. Smola, and K-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10:1299–1319, 1998.

[230] E. Schymanski, C. Gallampois, M. Krauss, M. Meringer, S. Neumann, T. Schulze, S. Wolf, and W. Brack. Consensus structure elucidation combining GC/EI-MS, structure generation, and calculated properties. *Analytical Chemistry*, 84:3287–3295, 2012.

[231] E. Schymanski, M. Meringer, and W. Brack. Matching structures to mass spectra using fragmentation patterns: Are the results as good as they look? *Analytical Chemistry*, 81:3608–3617, 2009.

[232] E. Schymanski, M. Meringer, and W. Brack. Automated strategies to identify compounds on the basis of GC/EI-MS and calculated properties. *Analytical Chemistry*, 83:903–912, 2011.

[233] D. Scott. Determination of chemical classes from mass spectra of toxic organic compounds by simca pattern recognition and information theory. *Analytical Chemistry*, 58:881–890, 1986.

[234] D. Scott. Effects of binary encoding on pattern recognition and library matching of spectral data. *Chemometrics and Intelligent Laboratory Systems*, 4:47–63, 1988.

[235] H. Scsibrany, M. Karlovits, W. Demuth, F. Müller, and K. Varmuza. Clustering and similarity of chemical structures represented by binary substructure descriptors. *Chemometrics and Intelligent Laboratory Systems*, 67:95–108, 2003.

[236] F. Serratosa, R. Alquezar, and A. Sanfeliu. Function-described graphs: A fast algorithm to compute a sub-optimal matching measure. In *Proceedings of the IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, pages 71–77, 1999.

[237] L. G. Shapiro and R. M. Haralick. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:504–519, 1981.

[238] J. Shawe-Taylor and N. Christianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[239] N. Shervashidze and K. Borgwardt. Fast subtree kernels on graphs. In *Proceedings of NIPS*, volume 22, page 1660, 2009.

[240] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proceedings of AISTATS*, 2009.

[241] Anil Shukla and Jean Futrell. Tandem mass spectrometry: Dissociation of ions by collisional activation. *Journal of Mass Spectrometry*, 35:1069–1090, 2000.

[242] M. Siegel and G. Gill. MASSPEC: A graphics-based data system for correlating a mass spectrum with a proposed structure. *Analytica Chimica Acta*, 237:459–472, 1990.

[243] A. Steffen, T. Kogej, C. Tyrchan, and O. Engkvist. Comparison of molecular fingerprint methods on the basis of biological profile data. *Journal of Chemical Information and Modeling*, 49:338–347, 2009.

[244] S. Stein. Chemical substructure identification by mass spectral library searching. *Journal of American Society for Mass Spectrometry*, 6:644–655, 1995.

[245] S. Stein, S. Heller, D. Tchekhovskoi, and I. Pletnev. *InChI Technical Manual*, 2011.

[246] S.E. Stein. Estimating probabilities of correct identification from results of mass spectral library searches. *Journal of The American Society for Mass Spectrometry*, 5:316–323, 1994.

[247] S.E. Stein and D. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *Journal of American Society of Mass Spectrometry*, 5:859–866, 1994.

[248] E. Sussenguth. A graph-theoretical algorithm for matching chemical structures. *Journal of Chemical Documentation*, 5:36–43, 1965.

[249] S. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21:i359–i368, 2005.

[250] D. Sweeney. Small molecules as mathematical partitions. *Analytical Chemistry*, 75:5362–5373, 2003.

[251] W. Szpankowski. Probabilistic analysis of generalized suffix trees. In *Proceedings of the Combinatorial Pattern Matching*, pages 1–14. Springer Verlag, 1992.

[252] R. Tarjan. Graph algorithms in chemical computation. In R. Christofferson, editor, *Algorithms for Chemical Computations*, pages 1–20. American Chemical Society, 1977.

[253] R. Todeschini and V. Consonni. *Handbook of Molecular Descriptors*. John Wiley & Sons, 2000.

[254] C. Tonnelier, P. Jauffret, T. Hanser, and G. Kaufman. Machine learning of generic reactions: 3. An efficient algorithm for maximal common substructure determination. *Tetrahedron Computer Methodology*, 3:351–358, 1990.

[255] W-H. Tsai and K-S. Fu. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 9:757–768, 1979.

[256] K. Tsuda and T. Kudo. Clustering graphs by weighted substructure mining. In *Proceedings of ICML*, pages 953–960, 2006.

[257] K. Tsuda, H. Shin, and Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21:ii59–ii65, 2005.

[258] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:695–703, 1988.

[259] K. Varmuza. *Pattern Recognition in Chemistry*. Springer-Verlag, 1980.

[260] K. Varmuza. Recognition of relationships between mass spectral data and chemical structures by multivariate data analysis. *Analytical Sciences*, 17:i467, 2001.

[261] K. Varmuza, M. Karlovits, and W. Demuth. Spectral similarity versus structural similarity: Infrared spectroscopy. *Analytica Chimica Acta*, 490:313–324, 2003.

[262] K. Varmuza and W. Werther. Mass spectral classifiers for supporting systematic structure elucidation. *Journal of Chemical Information and Computer Sciences*, 36:323–333, 1996.

[263] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

[264] S. V. N. Vishwanathan and J. Smola. Fast kernels for strings and tree matching. *Advances in Neural Information Processing Systems*, 15, 2003.

[265] G. E. Vleduts. Concerning one system of classification and codification of organic reactions. *Information Storage and Retrieval*, pages 117–146, 1963.

[266] K. Voorhees, P. Harrington, T. Street, S. Hoffman, S. Durfee, J. Bonelli, and C. Firnhaber. Approaches to pyrolysis / mass spectrometry data analysis of biological materials. *Computer-Enhanced Analytical Spectroscopy*, 2:259–275, 1990.

[267] A. Wade, P. Palmer, K. Hart, and C. Enke. Development of algorithms for automated elucidation of spectral feature/substructure relationships in tandem mass spectrometry. *Analytica Chimica Acta*, 215:169–186, 1988.

[268] M. Wagener and J. Gasteiger. The determination of maximum common substructures by a genetic algorithm: Application in synthesis design and for the structural analysis of biological activity. *Angewandte Chemie International Edition in English*, 33:1994, 1994.

[269] N. Wale, I. Watson, and G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and information systems*, 14:347, 2008.

[270] T. Wang and J. Zhou. EMCSS: A new method for maximal common substructure search. *Journal of Chemical Information and Modeling*, 37:828–834, 1997.

[271] Y. Wang, J. Xiao, T. Suzek, J. Zhang, J. Wang, and S. Bryant. Pubchem: A public information system for analyzing bioactivities of small molecules. *Nucleic Acids Research*, 37:W623–W633, 2009.

[272] Y-K. Wang, K-C. Fan, and J-T. Horng. Genetic-based search for error-correcting graph isomorphism. *IEEE Transactions on Systems, Man and Cybernetics*, 27:588–597, 1997.

[273] D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28:31–36, 1988.

[274] D. Weininger, A. Weininger, and J.L. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29:97–101, 1989.

[275] A. Weissberg and S. Dagan. Interpretation of ESI(+)-MS-MS spectra – Towards the identification of "unknowns". *International Journal of Mass Spectrometry*, 299:158–168, 2011.

[276] E. Werner, J-F. Heilier, C. Ducruix, E. Ezan, C. Junot, and J-C. Tabet. Mass spectrometry for the identification of the discriminating signals from metabolomics: Current status and future trends. *Journal of Chromatography B*, 871:143–164, 2008.

[277] W. Werther, H. Lohninger, F. Stancl, and K. Varmuza. Classification of mass spectra. A comparison of yes/no classification methods for the recognition of simple structural properties. *Chemometrics Intelligent Laboratory Systems*, 22:63–76, 1994.

[278] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54:150–168, 1932.

[279] T. Wieland, A. Kerber, and R. Laue. Principles of the generation of constitutional and configurational isomers. *Journal of Chemical Information and Computer Science*, 36:413–419, 1996.

[280] R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 19:634–648, 1997.

[281] S. Wolf, S. Schmidt, M. Müller-Hannemann, and S. Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinformatics*, 11:148, 2010.

[282] M. B. Wright. Speeding up the Hungarian algorithm. *Computers & Operations Research*, 17:95–96, 1990.

[283] Q. Wu. Multistage accurate mass spectrometry: A "basket in a basket" approach for structure elucidation and its application to a compound from combinatorial synthesis. *Analytical Chemistry*, 70:865–872, 1998.

[284] J. Xu. GMA: A generic match algorithm for structural homomorphism, isomorhism, and maximal common substructure match and its applications. *Journal of Chemical Information and Modeling*, 36:25–34, 1996.

[285] L. Xu and E. Oja. Improved simulated annealing, Boltzmann machine, and attributed graph matching. In *Neural Networks*, volume

412 of *Lecture Notes in Computer Science*, pages 151–161. Springer, 1990.

[286] M. Yadav, B. Kelley, and S. Silverman. The potential of a chemical graph transformation system. In *Graph Transformations*, volume 3256 of *Lecture Notes in Computer Science*, pages 83–95, 2004.

[287] Y. Yamanishi, M. Hattori, M. Kotera, S. Goto, and M. Kanehisa. E-zyme: Predicting potential EC numbers from the chemical transformation pattern of substrate-product pairs. *Bioinformatics*, 25:179–186, 2009.

[288] YR. Yao. *Handbook of Bond Dissociation Energies in Organic Compounds*. CRC Press, Boca Raton, FL, USA, 2003.

[289] H. Yoshida, R. Leardi, K. Funatsu, and K. Varmuza. Feature selection by genetic alogirhtms for mass spectral classifiers. *Analytica Chimica Acta*, 446:485–494, 2001.

[290] Z. Yu-Xi, X. Qing, Y. Gang, and L. Meng-Long. Computer-assisted prediction of the classification of the pesticide chemical structure in mass spectra. *Chinese Journal of Analytical Chemistry*, 35:1449–1454, 2007.

[291] J. Zhang, W. Gao, J. Cai, S. He, R. Zeng, and R. Chen. Predicting molecular formulas of fragment ions with isotope patterns in tandem mass spectra. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:217–230, 2005.

[292] K. Zhang, Y. Liang, and A. Chen. Selection of neutral losses and characteristic ions for mass spectral classifier. *Analyst*, 134:1717–1724, 2009.

[293] N. Zhang, R. Aebersold, and B. Schwikowski. Probid: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, 2:1406–1412, 2002.

[294] W. Zhang and B. Chait. Profound: An expert system for protein identification using mass spectrometric peptide mapping information. *Analytical Chemistry*, 72:2482–2489, 2000.