universität wien

DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

„Bulk Processing of Molecule Patent Associations“

verfasst von / submitted by

Patrick Penner

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magister der Pharmazie (Mag. Pharm.)

Wien, 2017 / Vienna, 2017

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 449
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Diplomstudium Pharmazie
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Thierry Langer
Mitbetreut von / Co-Supervisor:	

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Thierry Langer for the opportunity of this project and the many things I learned in the course of it. It was an enlightening experience and a very welcome chance to apply knowledge.

Furthermore, I would like to thank Gökhan Ibis for his continued support and guidance in software development.

I would also like to thank Dr. Thomas Seidel for his expertise in cheminformatics and the specific pointers he gave me along the way.

Miriam Penner deserves mentioning for her graphical design work on the KNIME Node icon.

Lastly I would like to thank my patient proofreaders: Katharina Penner, Miriam Penner, Arthur Garon, Clara van Hoey and Markus Wieder.

Kurzfassung

Die Suche nach Molekülen in chemischen Patenten ist schon seit Jahrzehnten eine Herausforderung. Die Ungenauigkeit von Moleküldarstellungen, die Extraktion von Strukturen aus Patenten und die große Anzahl veröffentlichter Patente erschweren dieses Unterfangen. Dieses Projekt widmet sich einem bis jetzt wenig beachteten Aspekt der Patentsuche, nämlich einer automatisierten Suche nach Patenten für größere Molekülmengen. Dazu wird ein Framework geschaffen, das von zwei Endnutzerapplikationen implementiert wird.

Als Datenquelle wurde der SureChEMBL Datensatz verwendet. Dieser wurde, zur besseren Handhabung, in eine relationale Datenbank eingespielt, die, auf einer mitgelieferten Vorlage aufbauend, der Nutzung entsprechend modifiziert wurde. Es wurde außerdem ein Interface programmiert, das die Verbindung der Datenbank mit einer Applikation steuert und die Nutzung erleichtert. Auf diesem Interface basierend, wurden zwei Programme in zwei verschiedenen Nutzeroberflächen implementiert: ein Knoten für die Workflowumgebung KNIME und ein Command Line Tool.

Anschliessend wurde eine Reihe von Experimenten durchgeführt, welche die Funktionalität des Systems beschreiben sollen. Die Fähigkeit des Systems Moleküle in patentierte und nicht patentierte Moleküle zu sortieren, wurde an der KNIME Node geprüft. Das System trennte den gegebenen Datensatz mit hoher Genauigkeit auf. Einzelne Fehlentscheidungen des Systems und grundsätzliche Probleme mit dem Versuchsaufbau wurden anschließend diskutiert. Außerdem beschreiben weitere Experimente mit verschiedener Thread Anzahl, mit verschieden großen Datensätzen und mit unterschiedlich stark patentierten Verbindungen die Performance des Systems. Das System braucht im Durchschnitt etwa eine Sekunde, um ein Molekül hinsichtlich der Patentsituation zu charakterisieren. Zwei praktische Beispiele illustrieren mögliche Anwendungen des Systems und zuletzt wird noch auf zwei mögliche zukünftige Features eingegangen.

Abstract

Searching through molecules in chemical patents has presented a challenge for decades. The ambiguity of representing molecules, their extraction from patents, and the sheer number of published patents all contribute to the complexity of the subject. This project attempts to fill a gap left by other patent searching systems. It creates a framework to allow automated processing of patent molecule associations. In addition, two user facing applications using this framework will be implemented.

A data source had to be found to provide the associations, and the data set generated by the SureChEMBL platform was used. The data then had to be represented in a format that would allow for easy retrieval. A relational database was designed based on a provided template and modified to suit the needs of the intended functionality. Furthermore, an interface was required that would bridge the gap between a user facing application and the database backend. This interface was then used to create separate implementations in two different environments. One being a KNIME node extension and the other being a command line tool.

A series of experiments was performed to determine whether the intended functionality had been achieved. The KNIME node implementation was used to test whether the node would correctly classify patented and not patented molecules. It was found that it did so with high accuracy. The reasons for incorrect molecule classification are discussed as well as some problems surrounding the test set up. Performance experiments describe the efficiency of the functionality. This varies depending on thread usage, size of the query data set, and the number of patents molecules are associated with. In general, approximately 1 second of average query time per molecule can be expected. Two use cases are included for illustration purposes and possible future features briefly discussed.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Significance

Patents accompany many forms of industry and research. They guarantee an inventors right to profit from an invention. In a pharmaceutical context the invention is usually a drug. This drug is protected by several chemical and pharmaceutical patents, giving it a period of market exclusivity. This guarantee of market exclusivity makes patents so important.

Chemical patent information has been a challenge for many years. The entities that chemical patents wish to protect have always had some form of ambiguity associated with them. Chemical entities are expressed in several different ways, for example via systematical names. Attempts to establish systematical chemical nomenclature have regrettably been met with only limited success[10] and with the introduction of Markush structures, by the eponymous legal case in 1924, even structural representations of molecules in patents have been very broad in scope.

Patents, however, require innovation. Innovation that can only be proven as such by showing its distinction from prior art. In other words, prior art has to be searchable to exclude it. Thus, developing methods to make searching prior art for chemical structures easier has been around for a while, with some of the first systems having been developed with punch cards in mind.[11] Yet after half a century of software development, most systems still suffer from severe limitations.[12]

With the advent of open patent data the field of patent searching has become accessible to many more researchers who have found applications for patent searching beyond prior art examination. Bregonje et al. describe patents as a unique source for chemical information potentially not found in classical scientific literature.[13] Projects such as the SCRIPDB try to make this information freely and easily usable by researchers.[14] In this

manner a field traditionally associated with industry and monopolized by commercial analysis has found innovative use in research.

The commercial importance of patents in chemistry associated industry and specifically pharmaceutical industry has steadily increased. With the threat of generic entry into the market looming over every drug, the pharmaceutical industry has devised a number of strategies to avoid what is known as the "Patent Cliff".[3][15][16][17] Development of new drugs is heavily influenced by patents and so methods of navigating the patent space still hold great importance.

A new addition to the field of chemical patent searching are automated extraction workflows that find molecules in patents and save these associations.[12] Automation is a very necessary next step to keep up with the increasing amount of published patents.

Access to the extracted information is often still limited to inputting one molecule at a time. Querying data sets on a larger scale is therefore very time intensive. Considering that modern drug development workflows operate on large data sets, there is a definite need to query patent molecule associations in an automated way.

1.2 Goals

This project addresses the topic of bulk processing patent molecule associations in three different aspects. The first aspect is, of course, a functional classification algorithm that splits input molecules into groups of patented molecules and not patented molecules. The data set used to determine this comes from the automated extraction pipeline implemented by the SureChEMBL platform.[8]

While a deterministic analysis of correct classification into patented molecules and not patented molecules is difficult, this system will be compared to current systems providing a similar service.

The second aspect is the automation of patent molecule association. The main goal of this project is providing a platform where users can input a list of molecules instead of singular entities. The ability of the system to query molecules in bulk will be evaluated mainly by its performance and speed.

A third aspect is the accessibility of this functionality to a user. Two implementations, each directed at a different user base, will be described. Use cases of both implementations will be presented and differences in functionality highlighted.

1.3 Structure

This thesis contains three different parts, in addition to this introduction. The following part, the background section, will discuss general information surrounding the project. It will highlight aspects of patents and their legislation, describe a variety of challenges that had to be met in the course of this project, will name some of the tools used to overcome

said challenges, and include an overview of existing solutions, too which this project will be compared.

The implementation section will describe the process of software development. Special attention will be paid to the different parts of the overall functionality. This section will contain the design decisions made to achieve the aforementioned goals. These design decisions will be explained and some of the implications for the user discussed.

The discussion section will analyze the functionality to determine whether the goals were met and will give examples as to the usage. It will feature a number of experiments and some use case demonstrations. It will also include an outlook on features that the functionality would benefit from.

CHAPTER 2

Background

2.1 Patents

This section will attempt to briefly define what a patent is and then describe elements of a patent. It will also highlight the role patents play in the pharmaceutical industry.

2.1.1 Definition of a Patent

The World Intellectual Property Organization (WIPO) defines a patent as follows[18]:

> A patent is an exclusive right granted for an invention, which is a product or a process that provides, in general, a new way of doing something, or offers a new technical solution to a problem. To get a patent, technical information about the invention must be disclosed to the public in a patent application.

While patent legislation often varies, this definition does encompass the general intention and legislative nature of patents. Patents are usually published as patent documents. These patent documents are heavily standardized and legislated by national and international authorities, resulting in some variability.

One organization attempting international standardization of patent documents is the WIPO[19]. The WIPO is a specialized agency of the United Nations and currently comprises 186 UN member states, as well as a few non-member states. One of its most important contributions to patent documentation is the list of standards including: recommendations for the minimum data elements needed to identify a patent document as such[20], a list of two letter codes for different patent issuing entities[21], and the contents of announcements in patent gazettes[22], among others.

Because of the USA's special position in the pharmaceutical market[23], several references will be made to the United States Patent and Trademark Office (USPTO) as a national

legislative body that issues patents. These are examples meant to represent the national aspect of patent legislation. Specific laws governing the term of patents or the types of patents issued will vary nationally.

Patents are granted and recognized on a national level. In some cases, such as the European Patent Office (EPO), treaties exist to allow centralized examination of patents for several countries. Nevertheless, most countries will require examination by a national patent office. Contrarily, while some patent documents are associated to the WIPO these are not legally binding patents. WIPO patent documents are an administrative tool to allow easier application for patents in the member states.

Some countries maintain a registration system, in which a patent is registered without thorough examination and eventually proven valid when challenged in court. The more common examination system requires a patent be evaluated according to specific guidelines. One such guideline is the USPTO Manual of Patent Examining Procedure.[24]

2.1.2 Elements of a Patent Document

A few different patent sections or fields have become international standard requirements. The particularities differ nationally, so legislation laid down by the USPTO will be used to illustrate them. The full arrangement of a patent application the USPTO requires, is discussed in [25].

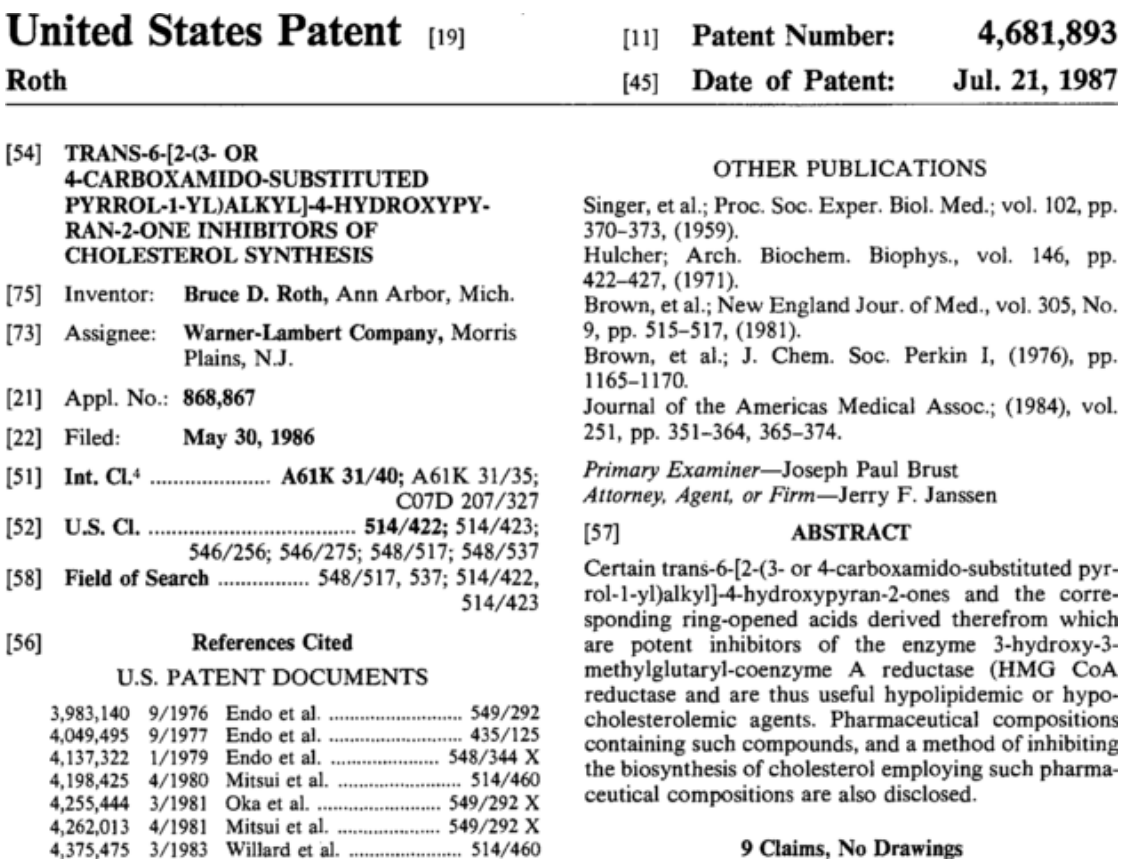
The first few elements of a patent are administrative in nature, for example information about the stated inventor. This information is taken from the application data sheet the USPTO requires at application.[26] The USPTO also requires a so-called information disclosure statement.[27] This contains patents and publications relevant to the invention.

Patents ordinarily provide an abstract[28] that summarizes the contents of the claims section. Abstracts are not very regulated and some patent examiners only check for adherence to length limitations. Some countries also require a dedicated summary that is considerably longer than an abstract.

To be examined, a patent application requires sufficient disclosure of the invention as well as a list of carefully chosen claims. According to the USPTO, the description of the invention, also referred to as the specification[29]:

> "[...]shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor or joint inventor of carrying out the invention."

The statement "best mode" in the statute set forth by the USPTO requires the inventor to disclose the most efficient parameters to reproduce his invention. It is included with the



United States Patent [19]

Roth

[11] **Patent Number: 4,681,893**

[45] **Date of Patent: Jul. 21, 1987**

[54] **TRANS-6-[2-(3- OR 4-CARBOXAMIDO-SUBSTITUTED PYRROL-1-YL)ALKYL]-4-HYDROXYPYRAN-2-ONE INHIBITORS OF CHOLESTEROL SYNTHESIS**

[75] Inventor: **Bruce D. Roth**, Ann Arbor, Mich.

[73] Assignee: **Warner-Lambert Company**, Morris Plains, N.J.

[21] Appl. No.: **868,867**

[22] Filed: **May 30, 1986**

[51] **Int. Cl.**[4] **A61K 31/40**; A61K 31/35; C07D 207/327

[52] **U.S. Cl.** **514/422**; 514/423; 546/256; 546/275; 548/517; 548/537

[58] **Field of Search** 548/517, 537; 514/422, 514/423

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,983,140	9/1976	Endo et al.	549/292
4,049,495	9/1977	Endo et al.	435/125
4,137,322	1/1979	Endo et al.	548/344 X
4,198,425	4/1980	Mitsui et al.	514/460
4,255,444	3/1981	Oka et al.	549/292 X
4,262,013	4/1981	Mitsui et al.	549/292 X
4,375,475	3/1983	Willard et al.	514/460

OTHER PUBLICATIONS

Singer, et al.; Proc. Soc. Exper. Biol. Med.; vol. 102, pp. 370–373, (1959).
Hulcher; Arch. Biochem. Biophys., vol. 146, pp. 422–427, (1971).
Brown, et al.; New England Jour. of Med., vol. 305, No. 9, pp. 515–517, (1981).
Brown, et al.; J. Chem. Soc. Perkin I, (1976), pp. 1165–1170.
Journal of the Americas Medical Assoc.; (1984), vol. 251, pp. 351–364, 365–374.

Primary Examiner—Joseph Paul Brust
Attorney, Agent, or Firm—Jerry F. Janssen

[57] **ABSTRACT**

Certain trans-6-[2-(3- or 4-carboxamido-substituted pyrrol-1-yl)alkyl]-4-hydroxypyran-2-ones and the corresponding ring-opened acids derived therefrom which are potent inhibitors of the enzyme 3-hydroxy-3-methylglutaryl-coenzyme A reductase (HMG CoA reductase and are thus useful hypolipidemic or hypocholesterolemic agents. Pharmaceutical compositions containing such compounds, and a method of inhibiting the biosynthesis of cholesterol employing such pharmaceutical compositions are also disclosed.

9 Claims, No Drawings

Figure 2.1: Title page of the primary Lipitor (Atorvastatin) patent[1]

intention to prevent the publication of very general descriptions that hinder reproduction of the invention. This "best mode" disclosure is an example of patent legislation that varies from country to country.

It may be an inventor's intention to disclose as little as possible or to obfuscate the patented subject matter. This is often done to deter competitors from copying the invention. However, patents should provide a basis to reproduce the invention after its term has expired, so sometimes legislation is employed to disallow certain obfuscation.

The claims section of a patent is the legally binding part, defining the scope of the granted exclusive rights.[29] It is arguably the most important part of every patent document and the most fought over. On one hand, the claims section attempts to be as broad as possible and include as many aspects of the invention as possible, on the other, it must avoid everything considered to be prior art or risk denial of application. Patents require a provably and completely novel aspect to be granted as such.[30]

The patent is also required to contain drawings where they are necessary to understand the invention.[31] In a chemical and pharmaceutical context these drawings often comprise

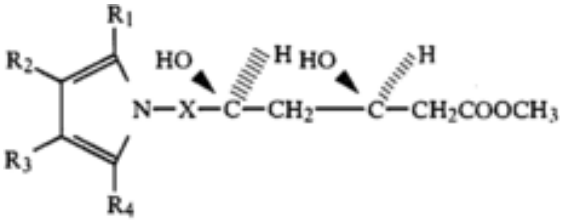


Figure 2.2: Example of a Markush structure comprising Atorvastatin, modified from [1]

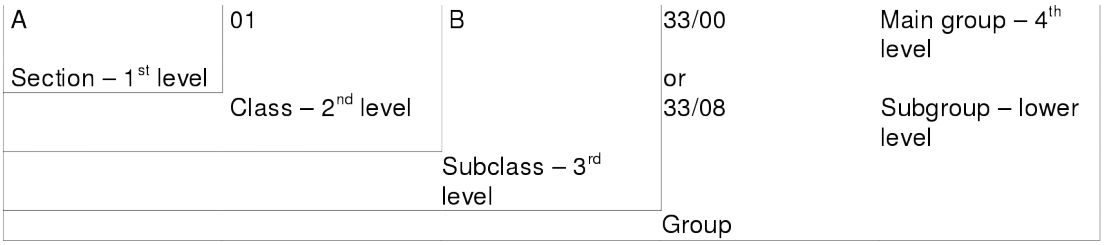


Figure 2.3: IPC patent classification divided into its parts, modified from [2]

specific compounds and can be included in the claims section. These compounds are commonly represented as Markush structures (figure 2.2) that encompass multiple compounds by different types of variable structures, connected to a central structure. Not all of these compounds have necessarily been synthesized. Those that have not been synthesized are referred to as "prophetic compounds". They are included as protection for the invention, to prevent someone from slightly varying a compound to achieve a very similar activity or property.[11]

Some patents also have attachments. In 2001 the USPTO started its Complex Working Unit (CWU) Pilot Program[32] to allow more complex entities, such as chemical structure drawings, mathematical formulae, or protein crystal structures to be submitted as attachments in their source format.

2.1.3 Patent Classification

Patent classifications attempt to group patents by their field of application. A number of classification systems exist, managed by different organizations.

International Patent Classification (IPC)

The IPC is the most widely used patent classification system and is administered by the WIPO. It consists of a number of symbols describing several sections, groups, and subgroups. The symbols denote hierarchical relationships of the groups each splitting up the previous one into smaller, more precise parts. An example is given in figure 2.3.

The section symbol "A" means human necessities. The class symbol "01" narrows the classification further to several fields associated with agriculture. The subclass and

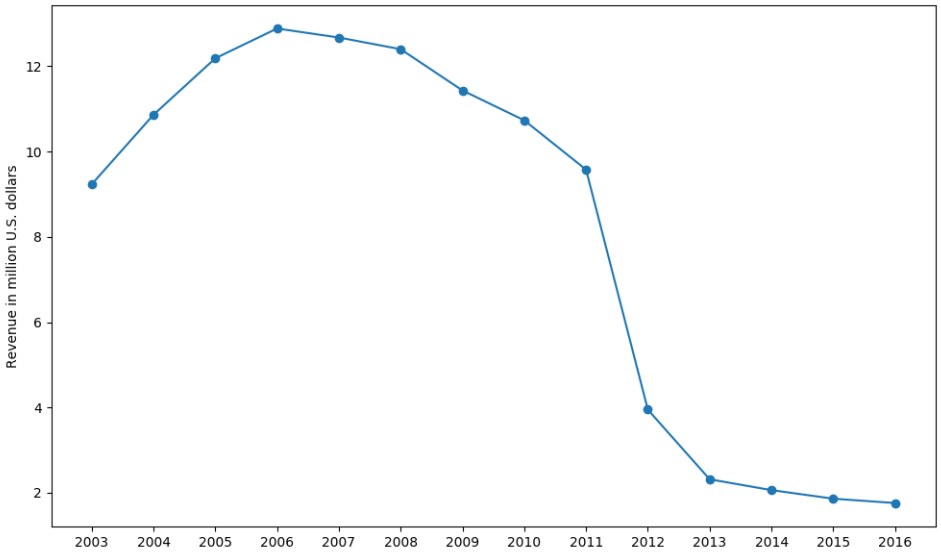


Figure 2.4: Revenue of Lipitor (Atorvastatin)[3]

group symbols then specify that this patent classification describes a patent for tilling implements with rotary driven tools.[33]

The IPC was initially developed as a tool to be used for paper based filing. So with the advent of computer systems, the IPC underwent a larger reform that is sometimes called the IPCR. In 2009 the original IPC classification system was discontinued.

Cooperative Patent Classification (CPC)

The CPC is a joint effort of the USPTO and the EPO to combine both of their internal classification systems into one. It is intended to be an extension of the IPC to allow for more specific separation. This eliminates the need for both offices to reclassify the patents of the other into their own system. Seeing as it is an extension of the IPC the structure of the CPC is very similar. Documentation concerning the additions to the IPC that the CPC provides can be found on the CPC website[34].

2.1.4 Patents in the Pharmaceutical Industry

Patents are a vital part of the pharmaceutical industry as it is now. The current "blockbuster business model" absolutely requires flagship drugs to be under patent protection to be profitable.[3] This is exemplified very clearly in the sharp decline of profitability when a molecule loses its patent protection.

One example is the drop of revenue Lipitor experienced after its patent protection expired. Figure 2.4 shows the revenue for Lipitor since 2003 and the steep decline in revenue since

2011. This is referred to as the "Patent Cliff" In 2011 Lipitor's extended patent term ended, making generic entry possible.

The end of patent protection for a molecule usually means an almost immediate appearance of generics and often a significant loss of market share. Another example of this is Prozac (Fluoxetine), which lost 70% of its market share after generic market entry.[35]

With R&D costs continually rising throughout the pharmaceutical industry[36], the need to protect these investments also increases. The period of market exclusivity achieved by patent protection becomes a vital instrument to ensure a return on those expenses. Thus many companies employ strategies to prolong that period[3] through strategic use of secondary patents or, more specifically, using concepts such as chiral switching

Secondary patents are filed late in the life cycle of a patented drug to lengthen its patent protection. Specific aspects of the drug are separately claimed, avoiding the claims of the previous patent. The previous patent is considered prior art at this point and reclaiming parts of it would invalidated the new patent. Claims include different crystal structures, different formulations, and even different usage.[17] In chiral switching it is a specific enantiomer that is claimed to replace a previously racemic claim. However, the claimant must prove that there is a significant difference between the racemic composition and the single enantiomer composition.[16]

2.2 Challenges

Associating molecules and patents is not without its challenges. Two of the challenges will be detailed in this section and some solutions discussed.

2.2.1 Molecule Representation

Molecule representation has always presented a challenge, especially in an informatics context. For example, a study of several small molecule databases revealed that dramatic inconsistencies existed between the same molecules in different databases.[37] Therefore the problem of molecule representation remains a relevant one. There are many ways of representing molecules in a computer readable format, yet not all provide the same functionality. Differences exist in the areas of:

- stereochemical information
- conformational information
- data size
- human readability
- uniqueness

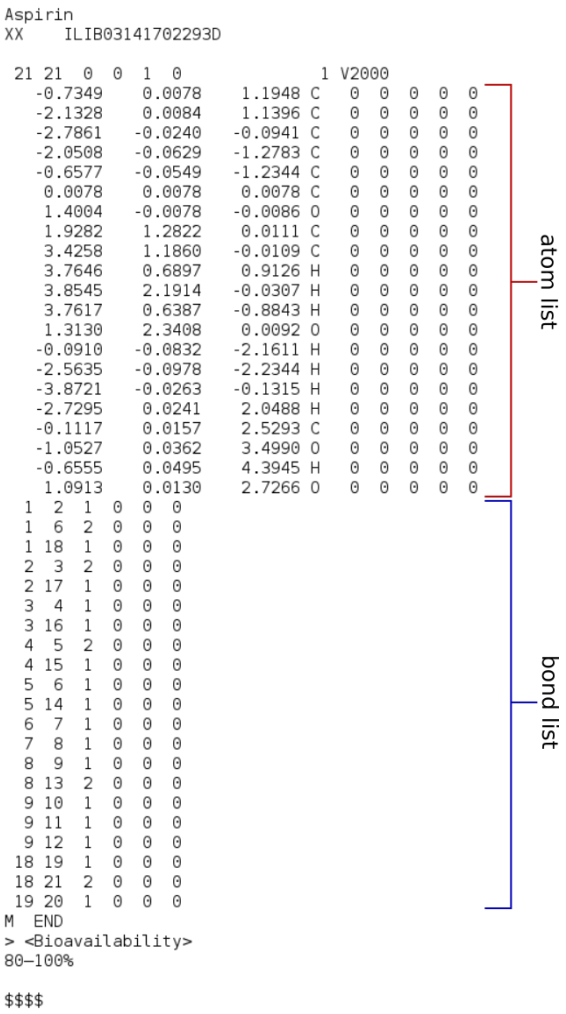


Figure 2.5: Aspirin in an SD file format, generated by the LigandScout SDF writer KNIME node

This section will discuss several different approaches to molecule representation, in the light of the aforementioned characteristics.

Structural Representations

A very intuitive approach to represent molecules in a digital way is a list of atom types with 3D coordinates and a list of bonds with defined bond orders. Such a format is implemented in a molfile or its extension, the structure data file (SDF). An example is given in figure 2.5. Information is stored in the form of a connection table. The most prominent parts of this connection table are the atom block and the bond block. The atom and bond block also contain the stereochemical information.[38]

Structural representations are unique in the fact that they can also convey conformational data. Through the use of 3D coordinates and point to point bonds, the bond angles and the torsion angles become variable. While the bond angles themselves will stay fairly consistent, depending on the chemical environment, the torsion angles can be manipulated at will. This is absolutely necessary in applications such as virtual screening.[39] This feature is, however, not necessary when trying to uniquely identify a molecule and may even confuse such attempts.

The main drawback of structural representations is their data size. Whether on file or in memory, a structural representation is a significant memory investment. This makes the use of structural representations as unique identifiers of molecules problematic, especially in larger data sets.

Human readability is not the intention of a structural representation of a molecule. Every attempt at visualization is connected to a translation effort onto a 3D or 2D environment. A structural file is difficult to interpret without visualization software.

While structural formats are very specific in their representation of a molecule, they may be too specific for use as unique identifiers. For example, conformational differences in two representations should not lead to mismatches when absolute configuration would match. This could be avoided if a conformational analysis would be done at every read in but this would result in massive overhead. Furthermore, minute differences in 3D coordinates, for example caused by different handling of the float data type, could also result in false mismatches and would necessitate a certain fuzziness in the comparison.

String Representations

There are several formats that attempt to represent molecules as strings. The most common and most popular is the Simplified Molecular Input Line Entry Specification (SMILES) format.[40] SMILES strings are a very simple way of representing molecules in a human readable and easy to use fashion. Writing molecules as SMILES string does not require automated processing, but can be done by anyone with a little practice.

A SMILES string, however, contains significantly less information than a structural representation. A SMILES string will only contain information about atom types, connectivity, bond orders, as well as some limited stereochemistry. Projection into a 3D space is achieved by using knowledge based interpretations of functional groups and bonds to specify bond lengths and torsion angles. Thus all conformational data is lost at read out in exchange for consensus data.

The main advantages of a SMILES string are its minimal data usage and its human readability. While not perfectly intuitive, SMILES strings are a lot more interpretable to the human eye than a structural file.

The use of SMILES strings as unique identifiers, while less data intensive, still remains problematic. There are usually multiple ways to write molecules in SMILES formats and there are multiple variations on canonicalization algorithms.[41] As such, it is hard to

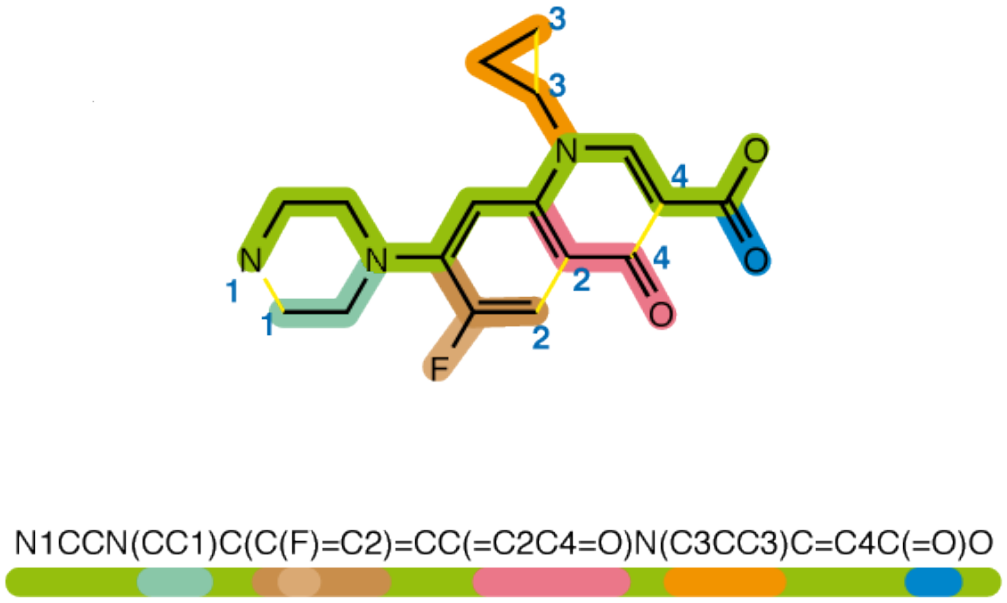


Figure 2.6: Generation of a SMILES string for Ciprofloxacin, modified from [4]

tell whether query SMILES strings will be generated the same way as SMILES strings in a databases. Different generation algorithms result in the same molecules being written differently. This can result in false mismatches when querying.

IUPAC International Chemical Identifier

A comparatively recent addition to string representation is the IUPAC International Chemical Identifier (InChI). It was developed by a group associated with the International Union of Pure and Applied Chemistry (IUPAC) to provide the following relevant functionality[42]:

- hierarchical
- structure based
- unique
- applicability to the domain of "classic organic chemistry"
- a default level of specificity to ensure interoperability between databases

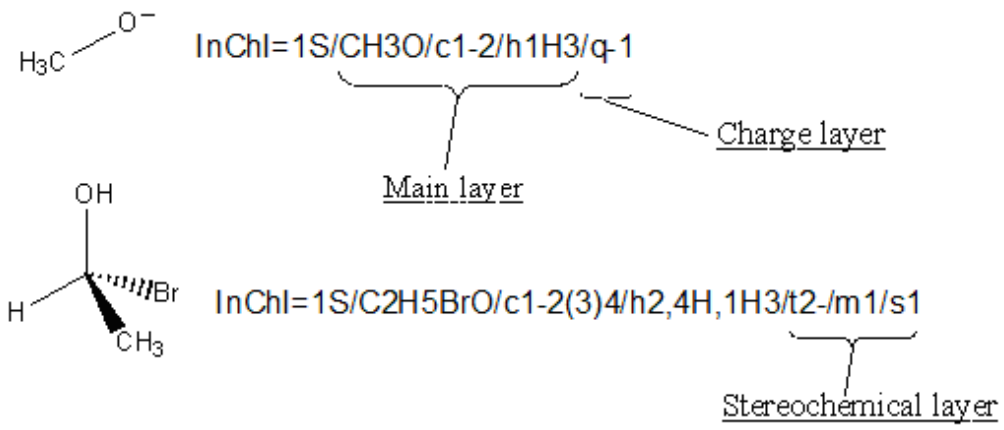


Figure 2.7: Standard InChI layers modified from [5]

The InChI uses the structure of the molecule to generate a layered identifier. Each layer contains a different type of molecule information, for example, a layer describing tautomers or a layer showing different levels of protonation.

Generating an InChI can lead to different results, depending on the options used. Flexibility in the generation can be seen as positive in terms of customizability, however, it can lead to known problems in communication of different databases with each other.[37]

The solution is the standard InChI. In its generation the standard InChI always considers the same three aspects of atom composition, connectivity, stereochemistry, and isotopic composition. This results in a system that always generates the same string for the same molecule. As such, the standard InChI is uniquely suited for use in databases.

The main drawback of the InChI is the return to a need for automated processing. An InChI is very hard to read for a human and even harder to generate. In this sense, it sacrifices human readability and simplicity for standardization, while retaining molecular structure.

Its structure based nature can lead to an InChI being a long string, depending on size and complexity of an encoded molecule. As a response to the need for a more compact identifier, the InChI key was developed.[42] It is a hash based derivative of the InChI, resulting in a 27 character string. The mathematical transformation into a hashed form does incur a loss of information and so an InChI key is not as easily converted back into a structure. Moreover, it also leads to the loss of the InChIs convenient hierarchical structure, making it impossible to cleanly separate information layers.

2.2.2 Patent Molecule Association

With the advent of open patent data, the development of extraction tools and databases for patent data has accelerated. Patent data is not usually an obvious source for scientific data, however some studies conclude that a large percentage of compounds published in patents are never published in scientific journals.[13] This elevates patent data from a position of patent avoidance, to an interesting target for data mining. A few of these attempts have already been published.[43]

Several problems arise when initially associating patents with molecules. This section will briefly discuss chemical extraction and relevance sorting.

Extracting Chemistry from Patents

Usually it is the Markush structures that contain most of the chemistry in a patent. 4 different points of variance in Markush structures are described by Downs et al.[11]:

- substituent variation – a list of alternative substituents
- position variation – a variable point of attachment (for example on a benzene)
- frequency variation – repetition of a substituent or variable group
- homology variation – substituents are only defined by a generic class related to each other by homology

This level of possible variation can lead to Markush structures so broad that enumeration becomes very problematic, whether automated or by hand. This has become so time consuming that the USPTO has attempted to limit the scope of Markush structures.[44]

Chemical entities explicitly disclosed outside of Markush structures may be hard to interpret as well. Many compounds are described through ambiguous trivial names that could describe a whole cluster of compounds. It is standard practice to describe compounds in the form of systematic names, however manually generated systematic names are often incorrect, so that deriving structures from them becomes impossible.[45] This leads to even more loss in extraction.

While some of these problems can be overcome in manual curation by the dynamic use of common sense, automated extraction is often not capable of such flexibility.

Determining Relevance

Having extracted chemistry from a patent, one is left with the question of the patent's relevance for the extracted molecules. A common approach to relevance scoring of patents for automatically extracted molecules is by position and frequency of the molecule in the patent.[46] Position in this case refers to the section under which it was found.

Scoring by frequency of occurrence follows the simple rationale that relevant molecules should be mentioned more often. This, while a starting point, is a bit simplistic. One way to refine this is by dissecting the patent and sorting by frequency in the different sections. A molecule that appears often in the claims may be more central than a molecule that only appears in the description. A molecule in the description may simply be exemplifying a form of activity displayed by the actual subject of the patent.

Depending on the use case of the patent search, another approach could be a rough pre-filter, sorting patents into possibly still valid and likely invalid by age of the patent. Especially in areas of development, the validity of a patent will be of higher importance than a compound's occurrence frequency. Even so such a pre-filter can only be considered rough, as the specific durations of patent validity vary by patent type and national legislation.

In general, it can be said that determining the relevance of a patent for a compound is a non-deterministic problem, especially when the possibility of litigation and patent challenge is considered.[15]

Bulk processing

Early stages of modern drug discovery and development often involve screening methods such as high throughput screening or, more recently, virtual screening. Screening methods involve compound libraries, not singular compounds and high levels of automation.[47] Thus, it stands to reason that tools involved in the drug discovery and development process should support the processing of compound libraries in an automated way.

2.3 Tools

This section will introduce a few tools that will be mentioned later in the implementation chapter. Specific details of their usage will be discussed there.

2.3.1 Relational Databases

Relational databases have been a very valuable tool in dealing with large amounts of data. They allow for fast retrieval and provide specialized functionality depending on the Relational Database Management System (RDBMS).

Data in a relational database is stored in tables. These tables consist of rows and columns. Every row consists of a number of fields corresponding to the number of columns. The rows aggregate the different fields associated with one record, while the columns define the data types of these fields. Supported data types vary by RDBMS. Columns can also have a number of constraints on them, requiring them to be unique or to correspond to another table. While the number of columns usually remains static, the number of rows is extendable.

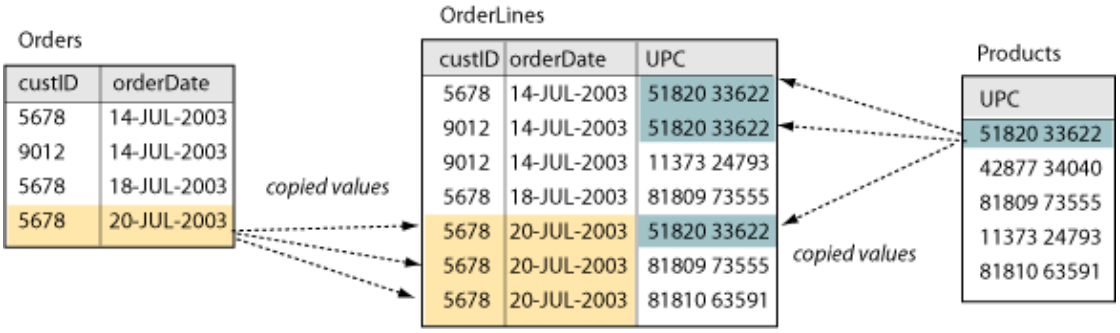


Figure 2.8: Example of a many-to-many relationship of orders and products[6]

To describe more complex data relationships often more than one table is needed. For example, one row may be connected to many different rows containing other data on another table. This would be a one-to-many relationship. To describe a relationship between different tables a relational database uses primary and foreign keys. A primary key represents a unique identifier for one specific row on one specific table. If this primary key is referenced on a different table to show a relationship between two rows it is a foreign key on this table as it originates from a different table and is foreign to this one.

Rows on two different tables can be connected to each other in many-to-many relationships as well. In that case single rows on one table are connected to many rows on the other, and vice versa. These kinds of many-to-many relationships are usually described on a mapping table that contains two foreign keys, one for each entity mapped, and information about the mapping. This way a many-to-many relationship can be expressed without duplicating data on one of the tables.

RDBMS

The infrastructure surrounding a relational database is the RDBMS. A variety of RDBMSs exist with different leanings and priorities in their implementations. From simple ones, designed for light usage, to heavy duty enterprise facing solutions, they all have different advantages and disadvantages. A proper comparison of RDBMSs is beyond the scope of this thesis. A starting point can be found in [48].

Structured Query Language (SQL)

SQL is a querying language used by virtually all relational databases. SQL was designed for use with relational database and has since become an ISO standard.[49] SQL attempts to be easy to use while still being capable of very specific querying. SQL is also used for a number of commands within databases such as table creation or transformation. Large parts of SQL are standardized, however, many RDBMS implement their own "dialect" of SQL that provides further functionality. This can lead to compatibility issues when using more specific SQL queries or scripts.

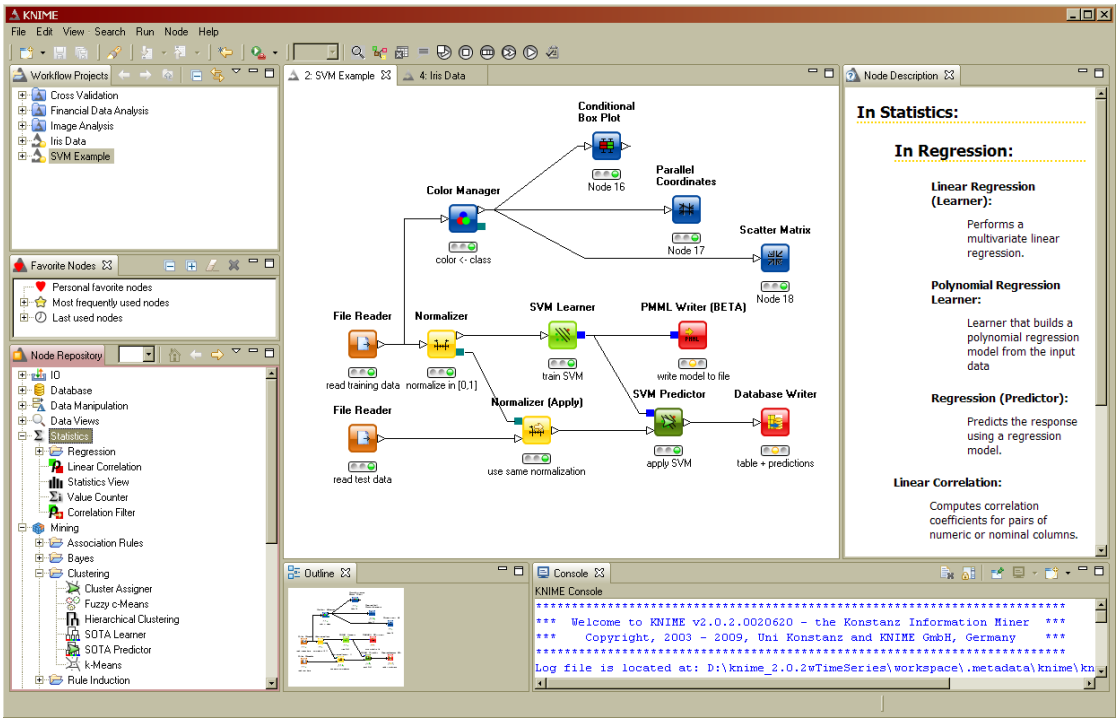


Figure 2.9: Screenshot of the KNIME example workflow, taken from [7]

2.3.2 KNIME

KNIME is an open source workflow program, implemented in Java, which allows users without knowledge of programming or scripting to use and combine graphical representations of scripts in a user friendly environment. The name derives from Konstanz Information Miner and refers to its roots as a tool for data mining and statistical analysis.[7] It is also widely used in cheminformatics and drug discovery in a competitive role to the commercial solution Pipeline Pilot.[50]

In KNIME functional units, called nodes, are dragged and dropped on a grid and connected to each other by ports that define input and output. Usually data is transferred between these ports in the form of tables, however, more specialized nodes do require and transmit other data types. This leads to graphical representation of a process.

KNIME brings two groups together within a common framework. It is dependent on developers to expand its array of functionality and tries to make this intuitive to use for the average user. It achieves this by providing an easily extendable framework and by defining guidelines to promote consistency across the nodes.[51]

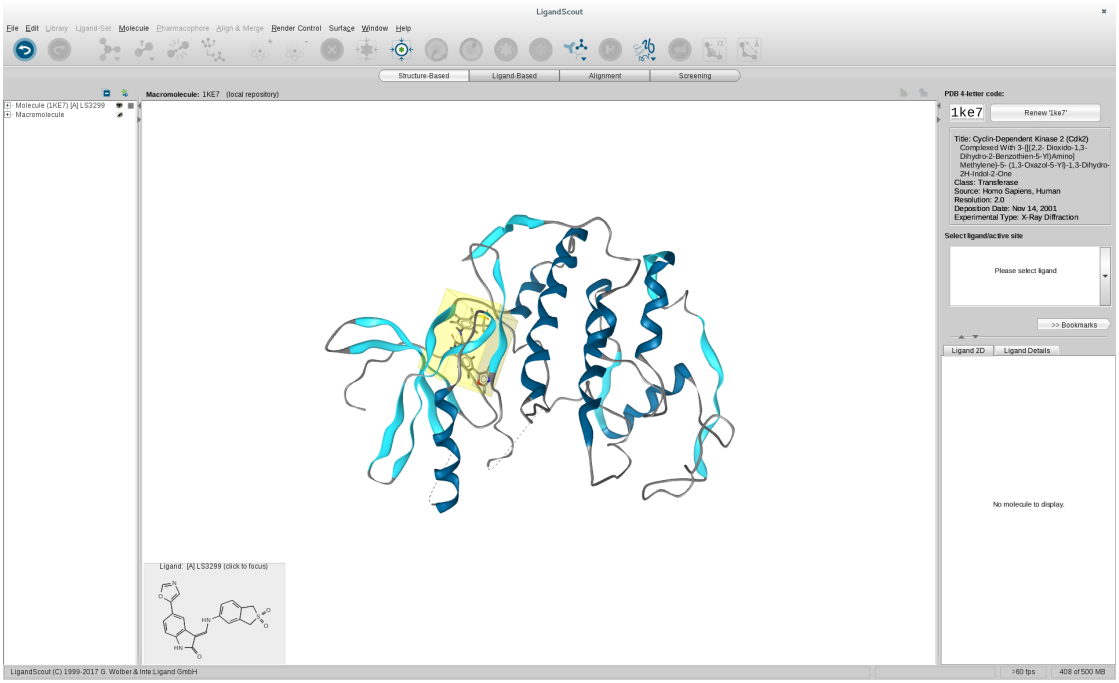


Figure 2.10: The LigandScout GUI

2.3.3 LigandScout

LigandScout is a commercial pharmacophore modelling and screening suite developed by Inte:Ligand GmbH.[52] It is known for its approachable Graphical User Interface (GUI) and user friendliness. It provides functionality for small and macromolecule visualisation, structure and ligand based pharmacophore generation, screening, and other functionalities.[53] This project will be in part built up from the LigandScout code base. This will include utilities from the LigandScout KNIME node package as well as the command line framework.

LigandScout provides an array of high quality KNIME nodes in addition to its standalone implementation. These offer the core LigandScout functionality in node format, as well as some utility functionality, such as Input/Output (IO) nodes. An advantage of having this functionality as KNIME nodes, is the possibility of bulk processing and automation, as well as a seamless interface between external functionalities.

LigandScout also provides command line tools for some discrete processes, such as, conformer generation, virtual screening and others. These tools are built from a common framework that interfaces Java functionality and the shell, making it usable over command line.

2.3.4 JNI-InChI

JNI-InChI provides a Java wrapper around the InChI C library developed by IUPAC.[54] JNI stands for Java Native Interface and describes the libraries function of providing Java function calls to generate InChIs and InChI keys. This allows for easy conversion of various Java molecule objects into an InChI.

2.4 Patent Searching Systems

This section will introduce some of the systems dedicated to chemical patent searching.

2.4.1 History

Searching for chemical structures in patents is not a new development. Since the early 1960s several systems exist to serve this purpose. The first of these systems were based on fragment codes, where alphanumeric characters would encode specific chemical fragments, such as functional groups. One such system is the Derwent Central Patent Index Chemical Code (CPI).[55] Originally designed for 80-column punch cards it attempts to encode Markush structures using a set of predefined chemical fragments. The CPI is still in use in the Derwent World Patents Index, which was up until recently a branch of Thomson Reuters Intellectual Property & Science and is now independent under the name Clarivate Analytics. This variant of a fragment code is described as a closed system, seeing as it only allows the use of predefined fragments. One example of an open system, assigning codes based on fixed rules instead of fragments, is the Genealogical Retrieval by Magnetic Tape Storage (GREMAS) system implemented by the Internationale Dokumentationsgesellschaft für Chemie GmbH (IDC).[56]

These fragment based systems were superceded by topological systems that began to appear in the 1980s. In contrast to a purely fragment based approach, systems like the MARPAT or Markush DARC[57] use combinations of fingerprint based searches and abstracted molecular graphs. This allows for speed and the ability to specify connectivity. It avoids the problem of two substances containing the same fragments, connected differently, having the same fragment code and being regarded as the same molecule.

Automated Extraction Systems

A fundamental development in patent molecule association, is the possibility of automated extraction of patent data from publications. A source of data for automated systems are, for example, CWUs in patent attachments. One system that takes advantage of these CWUs is SCRIPDB[14]. SCRIPDB, however, does not limit itself to patent molecule association but also describes chemical reactions and synthesis. Its main limitation is that only patents after 2001 contain CWU. This means only patents published after 2001 can be considered.

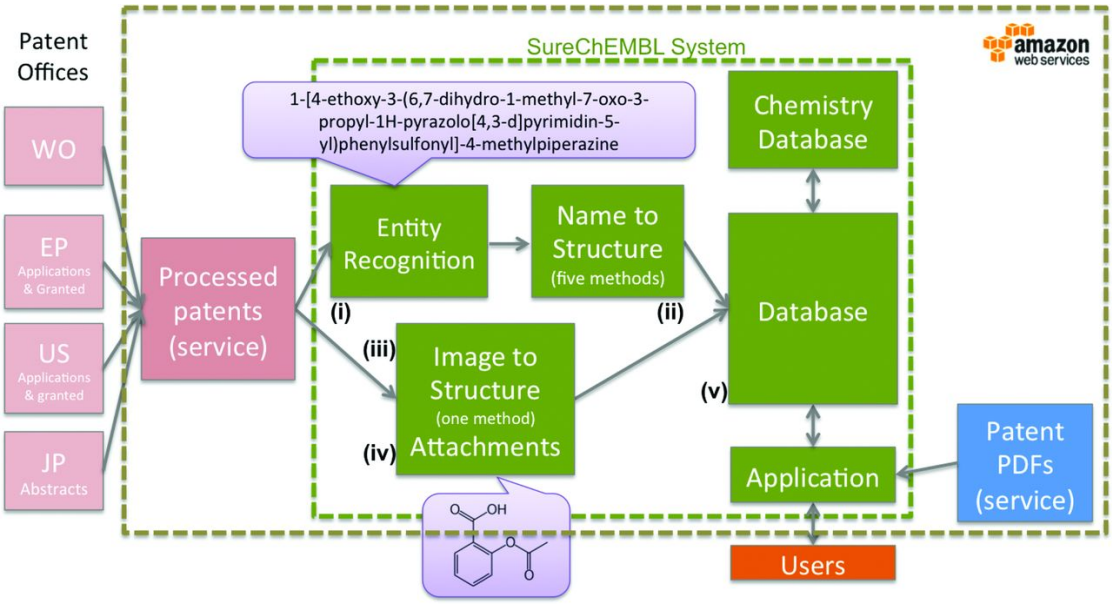


Figure 2.11: SureChEMBL Data Extraction Pipeline[8]

Circumventing the need for CWU requires very sophisticated semantic analysis techniques. One attempt was made by IBM in connection with several life sciences organizations. Data was extracted using the IBM Strategic IP Insight Platform[58] for the period of 1976 to 2010. Starting from 2001 CWUs were included in the extraction process. The data set was then provided to the US National Institutes of Health.[59]

2.4.2 Patent Searching as a Feature

Current commercial Systems offering patent searching in addition to their main functionality are, for example, Reaxys[60] and SciFinder[61]. Reaxys patent data is manually curated and therefore somewhat limited in scope. The number of patents published is too large to process by hand, therefore compromises have to be made to still remain up to date. Reaxys prioritizes patents by perceived importance, ignoring those not immediately relevant. While manual curation of patents may improve the accuracy of a search it also severely limits the throughput of data.

SciFinder relies on its PatentPak solution to find full text annotated patents for molecules. The scope of SciFinder is much broader than that of Reaxys, however, the mechanism of its patent search is proprietary and is therefore not disclosed.

2.4.3 SureChEMBL

Another new player in patent molecule associations is SureChEMBL.[8] Initially developed as a commercial entity under the name SureChem by Digital Science Ltd., it was donated to the European Bioinformatics Institute (EMBL-EBI) in 2013. Most public systems with

similar functionality are usually limited in their historical data coverage and not updated weekly with the publications of new patents. The SureChEMBL project attempts to achieve both of those tasks with a sophisticated data mining platform.

Chemical data is extracted from patents via three pathways shown in figure 2.11:

- name to structure conversion
- image to structure conversion
- attachment CWU parsing

As of 2017, the initial feed of data comes from the four major patent authorities WIPO, USPTO, EPO, and the Japanese Patent Office (JPO). This data is digitized and converted into XML by a patent content vendor named IFI Claims.

The entity recognition algorithm developed by SureChem scans the full text of the patent, extracts the contained chemistry, and also makes note of the section where a chemical entity was found in. The extracted names are then passed to several name to structure tools, with iterative contingencies in place to correct spelling or optical recognition mistakes. In a parallel branch, images are passed to the image to structure converter Keymodule CLiDE[62]. The chemical files attached as CWUs are parsed as well.

All successfully recovered structures are standardized and input through the ChemAxon JChem interface into a relational database. Unique identification in this system is achieved by canonical SMILES, generated using the ChemAxon Marvin toolkit. Molecules are supplemented with a variety of standard properties and additional representations, for example a standard InChI.

One of the main advantages SureChEMBL has over its competitors, is its dynamic nature. The fully automated data pipeline allows weekly updating of patent data at a rate of 80000 compounds or 50000 patents a month. The stated average latency between publication and searchability of a patent and its associated compounds within SureChEMBL is an impressive 1-4 days.[8]

The most visually prominent feature SureChEMBL boasts, is its GUI (figure 2.12), the main entry point for queries. It allows searching by keywords in full text patents supported by a Lucene index backend, chemical structures and substructures, as well as any combination of both. While a little daunting at first, it allows the creation of very efficient and specific queries. The search results are presented in an intuitive way and have some built in export functionality. However, SureChEMBL lacks one important feature, which is bulk data querying.

2.5 Summary

Patents in and of themselves are complex entities. While the core principles of patent legislation are similar internationally, important details can differ. Patent documents

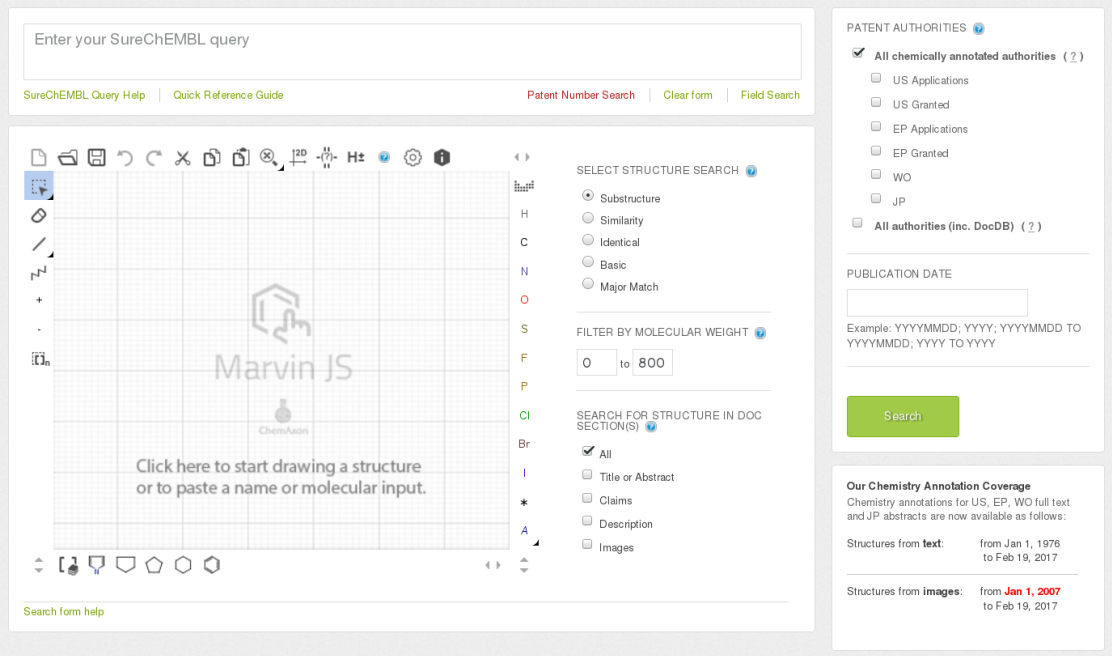


Figure 2.12: SureChEMBL GUI[9]

describing the nature, scope, and subject of the patent have a certain amount of standardization associated with them. However, particularly the subject of the patent can be expressed ambiguously. Despite this, patents play an integral role in the chemical and especially the pharmaceutical industry.

Chemical and pharmaceutical patents compound the complexity further because of their specialized subject matter. Representing chemical entities is not very straightforward, specifically in an informatics context. Furthermore, because of the inherent ambiguity of a patents subject, an automated association of chemical entities with relevant patents presents a challenge.

There are several frameworks that allow this project to meet these challenges. Starting with representation of patent molecule association in relational databases that allow intelligent and fast queries. Frameworks for handling and converting molecules are also a core part of this project. Furthermore, a way to communicate with the user makes the functionality available to different groups.

The need for patent searching is not a recently discovered one. Thus, this project is built on the experience and work of others. Open patent data has enabled much innovation in the field of patent searching, leading to smarter and more comprehensive systems. Automated extraction of chemistry from patents allows these systems to be consistently up-to-date. In the light of automated extraction, automated querying and bulk processing is the next logical step.

CHAPTER 3

Implementation

3.1 Intended Functionality

In the process of the project's implementation several decisions had to be made about the functionality to be developed. These will be briefly discussed before going into the specifics of the implementation. The driving force behind developing these tools was a need for bulk processing of patent molecule association. The implementation bulk processing requires overcoming a few challenges detailed below.

Import of Molecule Libraries

The first step to bulk processing is the import of the data sets that will be processed. In this case this means functional importing of compound libraries. Depending on the framework, IO must be handled differently.

Standardization of Molecules

Molecules come in very different formats. Consequently it is absolutely necessary to convert the input molecules into a format that is both unique and does not severely impact the performance. The standard InChI was chosen for its open source nature, its hierarchical structure, and its explicit design for use in database querying and interoperability. The InChI key was considered but would have prohibited a stereochemically unselective search.

Patent Molecule Association

The most integral step, of course, is the actual patent molecule association. A data set or a data search mechanism had to be found to describe the many-to-many relationships patents have with molecules. Initial attempts using Representational State Transfer

(REST) services to achieve this ran into several issues. In the end a relational database solution was built up using the data provided by SureChEMBL.

Classification

Classification of molecules into those with and those without associated patents is the most basic function the project aimed to achieve. A user should be able to easily split an entered data set by this criterion and continue working with each subset.

Relevance Sorting

Sorting patents by relevance is integral in any advanced usage of the patent search. Excluding all molecules with associated patents may be the simplest course of action. Nevertheless, a molecule that was patented before may have fallen out of patent protection. Such molecules could be extremely valuable in repurposing workflows. It would therefore be very helpful to also have filtering mechanisms that could retain these molecules.

Processable Output

Because of the complexity of patent data, human interpretation is still very important when processing results. Results must be presented in an interpretable way as well as a functional one. The interpretable option should allow users to make decisions about refining their search, while the functional one should integrate well with further processing of molecules as chemical entities.

User Friendliness

While the implementations described will vary in their interpretation of user friendliness, special emphasis was placed on making this functionality easily usable by average and advanced users. Patent searching is a very complex endeavour, so a deeper understanding of the subject will definitely enhance results, even so, this should not be a prerequisite.

3.2 Database

It quickly became clear that the SureChEMBL data set was the most up-to-date resource, yet, accessing it became problematic. The biggest problem was that SureChEMBL itself does not provide a REST Application Programming Interface (API).[8] REST service are a way to allow communication between a querying computer and a server connected to the internet for example using Hypertext Transfer Protocol (HTTP). Several REST services access the SureChEMBL data set, such as UniChem[63] and the Open PHACTS project[64], however these come with problems of their own.

The access of UniChem to the SureChEMBL data set is limited to its compound library. Querying whether a compound is in the SureChEMBL database would be a way to

classify whether a molecule has been in a patent, but would not make any attempt at qualifying that statement with validity, type, or relevance of that patent.

Even though the Open PHACTS project has access to the patent data contained in SureChEMBL, this access is initially limited by its static nature, seeing as only data up until 2012 has been integrated. In addition, several problems occurred in the accession of this data.

The entry point to the Open PHACTS system is the conversion of structural compound identifiers into a Unique Resource Identifier (URI).[65] Surprisingly many compounds queried this way returned puzzling internal server errors. While separate queries to the Open PHACTS data sources revealed that these compounds were, in fact, included in these databases, the Open PHACTS API itself was unable to find them. Furthermore, preliminary analysis of the compounds that were found and queried with patent lookup calls showed that there was a clear discrepancy between what the Open PHACTS API delivered and the data included in the SureChEMBL database, beyond the one caused by the limited integrated data set.

A decision was made to generate a modified version of the SureChEMBL database locally. The SureChEMBL project conveniently provides a collection of scripts to generate a mirror of the main database.[66] The main problem encountered in this process was database size, and so a few decisions were made to reduce it. As such it is about 45 gigabytes in size at time of writing.

3.2.1 Database Management System

PostgreSQL was chosen as the RDBMS for this project for its advanced functionality. The SureChEMBL data client, in its unmodified form, states that it supports Oracle and MySQL database systems. There is a clear bias towards Oracle as the support for MySQL only extends as far as a database agnostic schema. The bundle of scripts only contains specialized functionality for Oracle. However, Oracle and PostgreSQL are reasonably congruent in basic functionality and the SureChEMBL data client is written in a way to be easily extendable. Therefore using PostgreSQL only required some minimal modification of the scripts.

3.2.2 Database Schema

The SureChEMBL data client comes bundled with a database schema written for an Oracle or a MySQL database. It was modified in a few places to meet specifications. In general, the schema had few compatibility issues when it came to converting it for PostgreSQL. An overview of the modified database is given the appendix.

The database schema can be divided into three formal units:

1. documents
2. chemistry
3. interface

The document domain contains all information associated with the documents. This includes the titles, classifications, and a separate table for the issuing offices. The table of patent offices was created for performance reasons and is updated dynamically with every update of the database itself.

The chemistry domain contains all information associated with the chemical entities. This is split into a chemical property table and a chemical structure table. The chemical structure table contains a SMILES string, a standard InChI and a standard InChI key.

The last domain is found in between the aforementioned ones. This domain only consists of the "schembl_document_chemistry" table and acts as the interface between documents and chemistry. This table describes the many-to-many relationship the contained molecules have with the contained patents and maps them to each other using their respective primary keys.

Modifications

The main functional difference between the original schema and this modified one is in the "schembl_document_chemistry" table. In the original database schema this table consisted of the two foreign key columns, an integer column denoting the section of the patent, and an integer column denoting the frequency of occurrence in that section. This table was reformatted to have a column for the 6 patent sections described containing the frequency of occurrence. This reduces the number of rows and allows for easier querying.

Another major difference is the omission of the column "life_sci_relevant" from the table "schembl_document" which originally signified the life-science relevance of the patent. Non-life-science relevant patents where omitted in building this database in the interest of manageable size.

The "schembl_class_system" table was created to improve clarity when referencing the classification system of a classification under which a patent was filed. The original schema described the classification as integers and referenced them in the source code of one of the surechembl-data-client scripts. This is inconvenient and makes it hard to use the database without the surechembl-data-client.[66] The "schembl_class_system" table uses the integers in the "schembl_document_class" table as primary keys, connecting the integers to the name of the classification system.

3.2.3 Populating the Database

SureChEMBL provides two ways to download their data: as a map file and over a private File Transfer Protocol (FTP) server. The map file contains molecules mapped to patents in a line by line format where each line represents one mapping. This is a very verbose way to describe the many-to-many relationship otherwise contained in the database. It is also somewhat problematic for use as a primary data source to populate the database, mainly because of the redundancy of data necessary to display a many-to-many relationship in this format.

Fortunately the SureChEMBL provides FTP access to a far less redundant data set. This is the data set the surechembl-data-client is configured to use in order to replicate the SureChEMBL database.[66] This FTP data set is split into two parts: the backfiles and the frontfiles. The backfiles contain historical data extracted once and left static. These reach back as far as 1954. The frontfiles are updated daily and contain the results of the automated SureChEMBL data extraction pipeline.

3.3 Application Programming Interface

The main interface between end user applications and the database is the Patent Database Handler. The Patent Database Handler was implemented in the Java programming language. The Patent Database Handler provides the functionality to query patents for a compound.

3.3.1 Connection

The first step to a query is establishing the connection. A connection to a hosted database requires a few pieces of information. It requires a host name to identify where the database is hosted, a database name, a user name of an account with sufficient privileges for querying on the specified database, and a password if the account requires it. For added security the connection to the database is a Secure Sockets Layer (SSL) connection by default and supports common certificate authorities.

3.3.2 Multithreading

Both implementations discussed here use multiple threads and multiple connections to query the database. By default programs are executed along one thread, one step at a time, the steps being successive lines of code. Multithreading splits the program into multiple threads that can execute their own functions independently of each other. This also allows a program to distribute the processing load onto multiple cores.

Many different ways of implementing multithreading exist. The API itself does not contain code for multithreading to allow the application to decide on the framework it wishes to use. This way both implementations use a multithreading system native to their framework.

In this case multithreading was not implemented to distribute processing load. Using multiple threads allows the program to query multiple molecules at once. While parallel querying increases the absolute time one query takes to finish, experience has shown that the relative time for one molecule can be drastically reduced this way. This will be discussed further in section 4.2.

3.3.3 Molecule Conversion

Molecule conversion is handled outside the Patent Database Handler and is left to the application. Molecule conversion, in both implementations that will be discussed, begins with a LigandScout molecule representation. This, being natively Java, allows us to interface with the JNI-InChI library.[54] The JNI-InChI library converts the molecule into an InChI using the InChI C library distributed by the InChI trust.[67]

3.3.4 Configuration

Before a query call can be made, the Patent Database Handler requires some configuration. The main requirement is, of course, the identifier as a standard InChI. Two options, namely the number of patent offices and languages, are database dependent and are extracted at runtime to allow proper limiting of the results. The offices and languages contained in the database may increase, therefore, infrastructure has been put in place to compensate for this.

3.3.5 Filtering

The Patent Database Handler also allows a user to add some filters to the Query. Four main filters are implemented: a patent classification filter, a date filter, a field occurrence filter, and a patent office filter.

Classification Filter

The classification filter will remove all patents that are not filed under the specified classification. The hierarchical nature of patent classifications allows the user to define how selective the filter is supposed to be as the parts not specified remain open.

Patents are usually classified under a number of different classifications. For example, a patent for a series of small molecules in the treatment of a specific condition may under IPC be classified under the section "A" for human necessities as well as section "C" for chemistry. This should be kept in mind when filtering this way. Supported classifications are IPC, IPCR, and CPC.

Date Filter

A date filter is provided to filter out clearly outdated patents or to examine a certain time period. Patent documents more likely to still be valid (in other words, documents

not older than 25 years) are implicitly sorted to the top and are therefore more likely to be retrieved. However, in cases where molecules do not have many patents associated with them, the date filter allows the complete removal of outdated documents. In a more explorative analysis the date filter allows the user to define a particular time period.

Field Filter

The field filter allows the user to specify the frequency of occurrence for the queried molecule in a specific section or field a patent document requires to be retrieved. In other words, a field filter requiring 1 occurrence in the abstract will only retrieve patents for a molecule if the patent contains a reference for that molecule in the abstract. If a patent does not contain a reference to the molecule in the abstract it will be ignored. This allows the elimination of noise caused by patent documents mentioning various substances as examples or illustration. Known or off-patent drugs are particularly noisy and, for example, requiring an entry in the claims section will occasionally drastically reduce the number of patent documents applicable.

Patent Office Filter

The patent office filter excludes patent documents based on the patent office that issued them. At time of writing patents from five patent offices are included in the database corresponding to the ones contained in the SureChEMBL database. These will be dynamically expanded as soon as more are added to the SureChEMBL database. Issuing patents is still the prerogative of national patent offices, therefore it makes sense to be able to filter by them. Patent offices are represented by their two letter code laid down by the WIPO.[21] It is important to keep in mind that although patent documents issued by the WIPO exist, these are purely bureaucratic and do not represent legally binding patents.

3.3.6 The Query

The query itself is split into two parts for performance reasons. The first part is directed at the chemistry domain of the database. A stereounselective search of the given identifier is performed to find associated compounds in the database.

The decision to search for molecules disregarding stereochemistry is rooted in the fact that most patents are issued for racemic compounds. A specific stereoisomer may be patented separately if its properties are different enough from the racemic composition.[16] This is, however, a specialized strategy. A stereoselective search may mean missing information that would have been retrieved otherwise. Therefore a decision was made to err on the side of caution.

The second part of the query retrieves the patents associated with the previously retrieved compounds. This is done through the mapping table. The first filters are applied in the document domain of the database, filtering by date and occurrences in fields. The result is then passed on to post-processing.

3.3.7 Processing Results

The first step in the post-processing is still done in the SQL body of the query. The results of the main query are sorted according to several rules. The first rule aims to approximate patent expiration by checking if it was issued within the last 25 years.

The second rule begins the sorting by fields. Occurrences in the title are not sorted by their frequency, but by whether they are above 0. Although the data set contains entries where occurrences in title appear to be higher than one, in reality this is usually a reading mistake. Sorting by frequency of occurrence in title, tends to sort German language patents to the top for unknown reasons. In general, it makes little sense to demand an occurrence above one in the title. The sorting continues in a hierarchy corresponding to the legal and practical importance of the patent section or field.

The most important section of the patent is the claims section, as it is legally binding. The reason results are sorted by claims, only after having been sorted by title is that the title is far shorter and will therefore contain less noise. After that comes the abstract. This is prioritized assuming that a document's summary would preferably contain important elements.

After the abstract come the two fields that actually contain structural representations of the molecule, in the order: attachments and images.[32] The attachments are prioritized above the images because of their deterministic nature. Usually an attachment refers to a CWU. An attached molfile will usually only contain one molecule, while optical recognition of an image of a Markush structure may lead to very many molecules, only some of which the inventor has actually synthesized and tested. Furthermore, optical recognition will tend to have more mistakes than parsing a molecule file.

The end of this sorting cascade is the description, it being the most noisy of the fields. The description will, of course, include molecules used as examples to illustrate a point. Therefore it is deemed the most irrelevant.

Before returning results, the database performs one last step: it limits the results to minimize IO performance issues. On the client side the results are then subjected to the patent office filter and a language filter that prioritizes English language patents.

3.4 KNIME Implementation

The first implementation to be discussed is the KNIME node extension. KNIME itself is built on the Eclipse Integrated Development Environment (IDE), one of the most important IDEs in Java development.[68] KNIME uses the Eclipse plugin capability to allow developers to extend KNIME functionality at so called extension points. To this end they provide the KNIME Software Development Kit (SDK), a developer version of KNIME. This comes bundled with the extension wizard, which generates the required files for the developer.

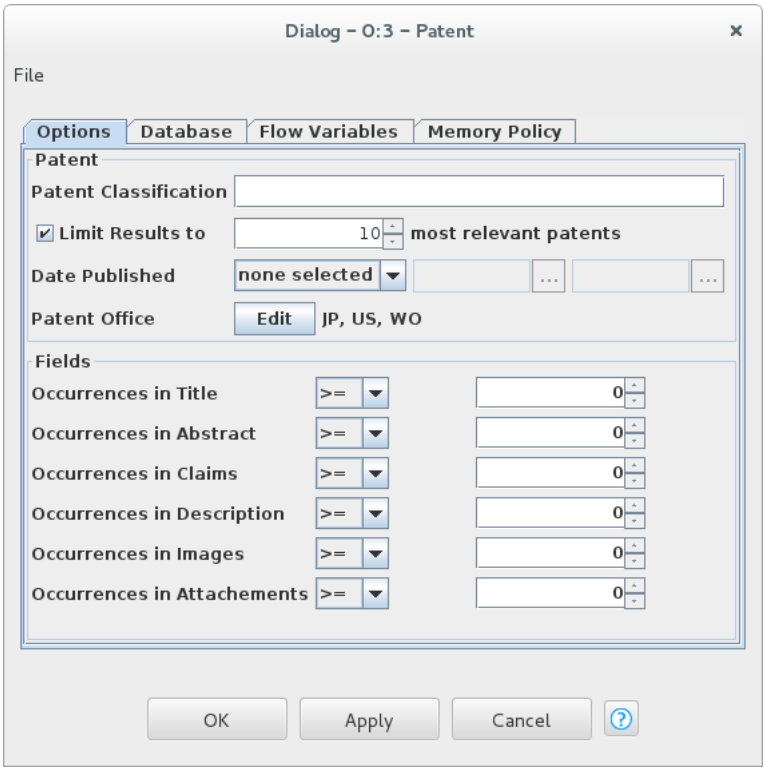


Figure 3.1: Filtering tab of the Patent KNIME node extension

3.4.1 Configuration

The KNIME Noding Guidelines require a node to configure its input and output even before the user has applied any settings. This allows a long chain of nodes to be configured at once and to be executed at once. In practice this means that the moment one node is connected to another, they begin to interact.

If user input is required, as is the case in this implementation, the user interacts with the node using a dialog. The KNIME Noding Guidelines lay down a few recommendations for GUI design within KNIME to allow for consistency. The dialog can be split into different tabs dealing with different subjects. In this case 2 tabs were implemented.

The first and most important is the filtering GUI (figure 3.1). It contains control options for all implemented filters the API can handle to allow for full control of the query. The second is the connectivity GUI (figure 3.2) allowing a user to control the necessary connection information, such as host name or user name. The node will perform some minimal input validation, for example to check if publication date ranges are possible.

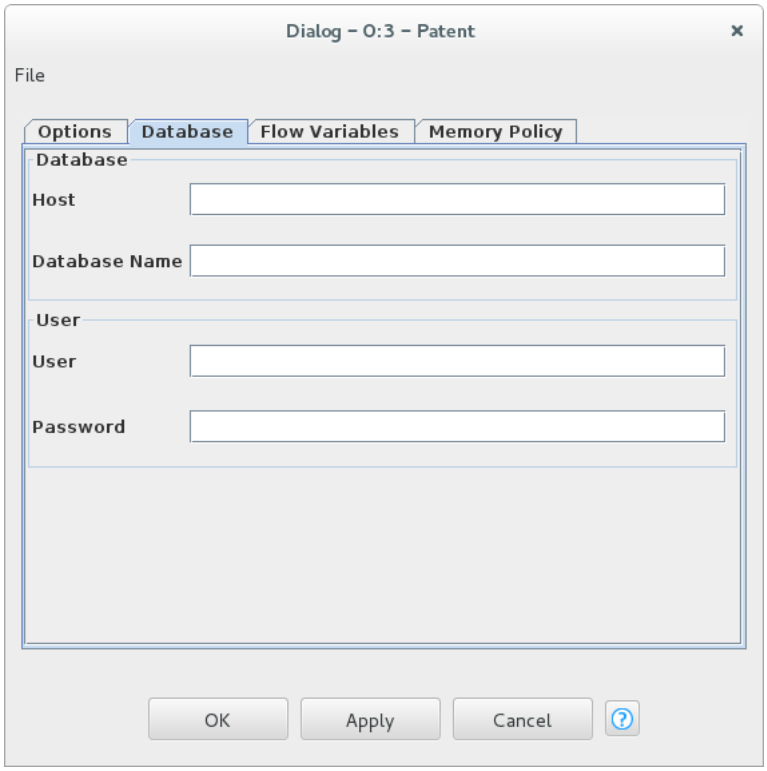


Figure 3.2: Connectivity tab of the Patent KNIME node extension

3.4.2 Input/Output

In general, the KNIME node is supposed to take a column of molecules as input and append patent information. Input molecules can be in any format LigandScout supports, as the extraction and representation of the molecule is done with LigandScout utilities.

The molecules are then output into two tables dividing molecules with associated patents and those without. The table with patents is built in such a way that depending on the number of patents found for the molecule, the row identifying the molecule is duplicated and information about the retrieved patents appended. This is done to express the one-to-many relationship the queried molecule will have with the patents. The table without patents contains a column that explicitly states that no patents were found for this molecule.

3.5 Command Line Implementation

Command line access to the patent search was added as an alternative way to use the functionality outside of KNIME. Not all users, especially the advanced ones, will want to be constrained to using KNIME and so this implementation presents a way to circumvent that.

3.5.1 Configuration

The framework used in this implementation is the LigandScout Command Line framework that provides some utilities for user friendly command line tools. Interaction between the user and the program on the command line is done in the form of command line options that are appended after the program execution statement. This is where the filters and the connectivity data are specified.

3.5.2 Input/Output

IO is also mostly handled within LigandScout functionalities. The user can decide whether to split the files containing molecules with patents and those without. The user can also specify the output file format. Supported formats are mostly chemical file formats but ".csv" and ".tsv" formats are also supported to allow an easy export into, for example, a spreadsheet program. This is done to facilitate an overview, especially of the found patents. Conventional chemical file formats will not be able to hold all the information extracted from patents, so the delimited file formats are given as an alternative.

3.6 Summary

Before the process of implementation could begin, the intended functionality was defined. The most basic function to be implemented was the classification into patented molecules and not patented molecules. This unitary function would then have to be embedded into a framework of bulk processing and the necessary infrastructure that handling data in bulk requires.

The first step in the implementation was finding a data source. After initial difficulty in finding a data source, one was generated based on the SureChEMBL database. SureChEMBL provides a way to set up a mirror of their database. This system was adapted to meet the needs of this implementation.

The intermediate step was the creation of an API to facilitate application development. This was achieved with the Patent Database Handler that is the connection point to the patent database backend. It also provides the filtering mechanisms necessary for more advanced queries.

The last step of the development cascade was the creation of two user facing applications in different frameworks. One was the KNIME node extension. This implementation allows the user to interact with the functionality using a GUI in a user friendly environment. The command line implementation was created to allow advanced users to access the functionality, in a way that allows convenient integration into scripting workflows.

CHAPTER 4

Discussion

The discussion section will present several experiments conducted with both implementations and analyze whether the aims of the project were achieved. The performed experiments will focus on the classification functionality, the performance in terms of speed, and on exemplifying real-world applications of the implemented functionality.

4.1 Classification Experiment

The aim of the classification experiment is to determine if the implementation can distinguish between molecules with and without associated patents. Seeing that validated and standardized test sets are not readily available, one had to be created over the course of the experiment.[12]

4.1.1 Data Set Creation

The data set was created from two sources attempting to represent either patented or not patented molecules. Patented molecules were chosen at random from the Prestwick Chemical Library. The Prestwick Chemical Library contains 1280 unique small molecules of approved drugs with sophisticated annotation. Every molecule has been associated with their initial patent, derived from a variety of sources such as the USPTO, the EPO, and Thomson Reuters Integrity.[69]

Proving that a molecule is not patented is significantly more complex and therefore the following method was used. 100 unique drug like molecules were randomly chosen from the ZINC database[70]. These were then checked for patents in the standard commercial solutions Reaxys[60] and SciFinder[61]. Reaxys found no patents for any of the given compounds. ScifFinder found patents for 36 of the initial molecules. Any mention in a patent document eliminated a molecule.

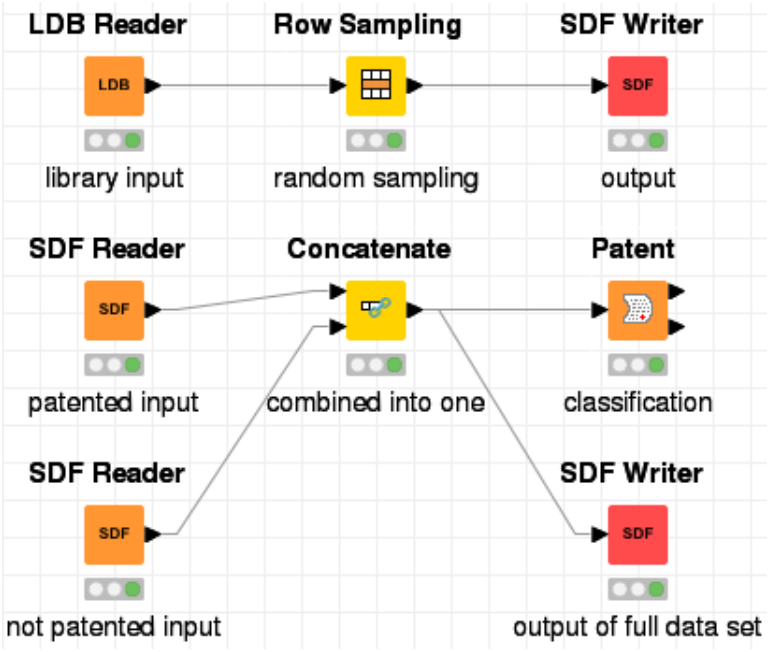


Figure 4.1: The KNIME workflow used to perform the classification experiment

To allow for a balanced data set, 64 random molecules from the Prestwick Chemical Library were added to the 64 molecules found to have no patents in Reaxys or SciFinder. All molecules were combined into an SD file.

4.1.2 Classification

The actual classification was done in the KNIME implementation. Figure 4.1 shows the workflow used to perform the experiment. The top part of the workflow is run for the Prestwick Chemical Library as well as a drug like ZINC database fragment. Both samples were exported. The sample from the ZINC database then underwent manual processing to confirm no patents were found for the chosen molecules. The remaining molecules were then concatenated and classified by the patent node. The full test set was exported for further use. The results of the classification can be found in figure 4.2. Two of the patented molecules could not be classified correctly. All of the not patented molecules were classified correctly.

4.1.3 Analysis

The two molecules that were not found ran into different issues. The first issue was one of molecular representation. The implementation as it is now only supports exact matching. This is usually not a problem due the efficient standardization that the InChI provides. However, exact matching has its limitations. In this specific case the SureChEMBL data set had extracted the molecule in question from patent documents only as a salt and

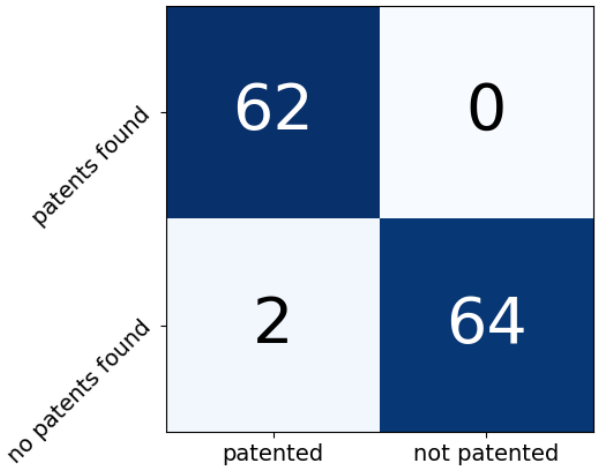


Figure 4.2: Confusion matrix of patent classification

stored it as such. The query molecule was not provided as a salt and thus a different standard InChI was generated for it. The standard InChI does not separate salts into a different layer, which makes it hard to account for different salts of the same molecule in an exact match search.

This highlights the need for a more permissive search functionality discussed in section 4.4.2. It should also show that storing a molecule as a specific salt in a database may be highly problematic. Integrating a salt stripper into the query workflow is easily done. Reinterpreting a data set after extraction to permit searching for generic molecules as well as specific salts may be quite resource intensive.

The other molecule was not found because it was missing from the data set imported from SureChEMBL. Considering this is an approved drug, oversight of this molecule is a glaring omission. The SureChEMBL project has released patches for its data before and hopefully this molecule will be included in one as well.

Another problem shown very early in the experiment is the lack of a validated test set for patent retrieval mechanisms. In this case the Prestwick Chemical Library, the ZINC database, Reaxys, and Scifinder had to be used to create a functional test set. The Prestwick Chemical Library and Reaxys patent information had to be manually curated beforehand and both Reaxys and SciFinder did not support querying a list of molecules.

The created data set has a shortcoming insofar as both patented and not patented molecules are very clearly patented or not patented. In other words, the patented molecules usually have a lot of patents associated with them. Senger et al. were able to correlate the number of patents a molecule had with the likelihood of its retrieval.[12] The molecules used for this data set were approved drugs that are usually associated

with very many patents. This results in a very unambiguous data set. On the other hand, creating a more ambiguous test set is hardly deterministic and would open up the results to interpretation. This once again exemplifies the native complexity associated with patent searching and the need for a properly validated test set for patent molecule association.

Furthermore the data set is comparatively small. A 128 molecule data set is not a representative sample of the chemical space. The size of the data set was mainly limited by the time needed to create it. Querying molecules for patents is very time intensive because every molecule has to be queried individually. This experiment was mainly set up to prove that the functionality can perform its basic function and it clearly demonstrated as much.

4.2 Performance Experiments

When processing large amounts of data, performance becomes a very relevant topic. Performance testing was done in two parts. Section 4.2.1 describes the query speed in relation to the number of threads used to query simultaneously. After this, section 4.2.2 describes the speed of queries by number of molecules.

4.2.1 Performance by Threads Used

As explained previously, multithreading the applications is not done to speed up processing, it is done to perform multiple queries at once. Processing on the client side is almost negligible. Only IO requires a relevant amount of processing.

Testing for the optimal number of threads was done with the command line implementation, seeing as this version supports explicitly setting the number of cores used, and by extension how many threads are run simultaneously. The data set used was the data set previously created for the classification experiment. This data set provided a balance of patented and not patented molecules. The program was run 3 times for every configuration to account for possible variance. The default setting of both implementations is creating a number of threads corresponding to the number of available processor cores. The computer that was used for this experiment had 8 processors.

Figure 4.3 shows the results of the experiment. As additional threads were used the query time dropped sharply, reaching its lowest point at 7 active concurrent threads. Additional threads stay within comparable levels. The minimal linear increase of query time for number of threads being above 7 may be due to overhead on the client side or the queries of one user on the database impacting each other.

The more threads are added, the more the absolute query time for each thread increases. The relative query time for all molecules, however, drops sharply. The variance encountered in measuring the query time was negligible and is therefore not included in figure 4.3.

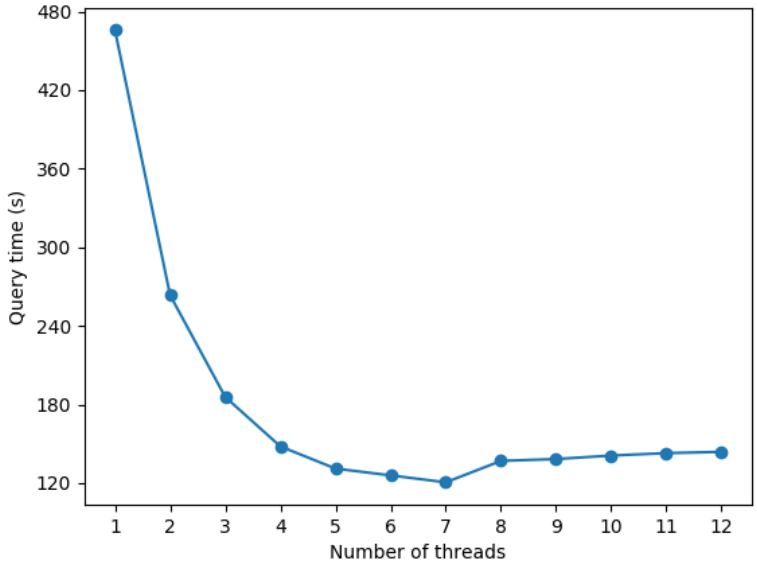


Figure 4.3: Overall query time by number of threads used for the data set created in section 4.1

Table 4.1: Query time by number of molecules

Number of molecules	Total query time	Query time per molecule
1	2.977 s	2977 ms
10	9.007 s	900 ms
100	1.171 min	702 ms
1000	11.41 min	684 ms
10,000	1.901 h	684 ms
100,000	19.08 h	686 ms

4.2.2 Query Time by Number of Molecules

The aim of this experiment was mainly to investigate how the functionality performs when given more and more data to process. To this end a data set of the generated molecules described in section 4.3.2 was split into incrementing log units and queried using the command line implementation. The resulting query times are described in table 4.1 and scaling of query time is visualized in figure 4.4.

The time per molecule queried clearly decreases depending on the size of the data set queried. This is an expected byproduct of the overhead generated by the command line tool, which remains the same for any data size. Therefore the impact of the overhead is reduced relative to the size of the data set.

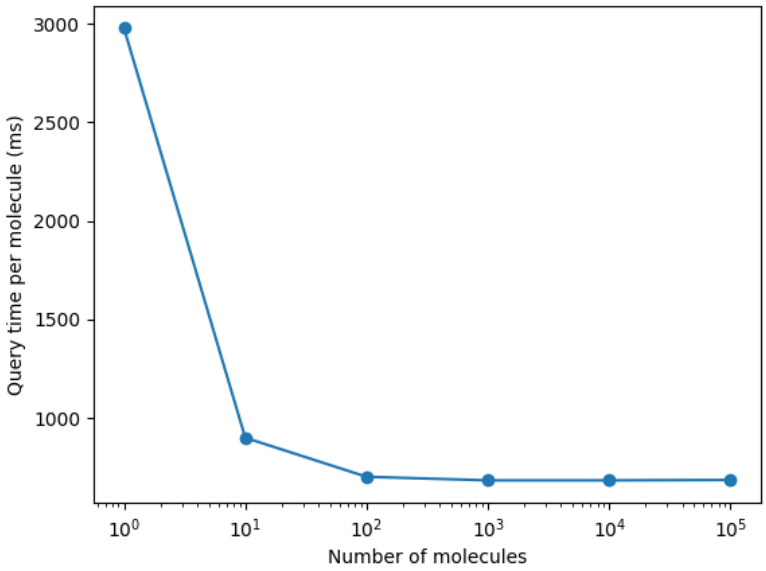


Figure 4.4: Scaling of query time per molecule

Another factor influencing the query time per molecule is the number of patents retrieved. The data set used for this experiment was largely not patented. The most patents were in fact contained in the fraction of 100,000 compounds, which could be an explanation of the slightly longer query time per molecule.

An example of the effect heavily patented molecules can have on the query time is shown in figure 4.5. The data set used was once again the data set from section 4.1. The patented molecules, as approved drugs, have a lot of patents associated with them, which results in an effect on the performance.

4.3 Use Case Demonstrations

This section is intended to demonstrate a use case for each implementation.

4.3.1 Integration into a Screening Workflow

This use case will demonstrate how the KNIME implementation can be integrated into a screening workflow. LigandScout provides KNIME nodes to screen within KNIME directly and chemical files can be imported through a variety of IO nodes. Thus for many basic use cases there is no need to leave the KNIME environment.

In this case the molecules were screened externally and exported into an SD file. The data can be imported into KNIME via, for example, an SDF reader node. The 20 molecules

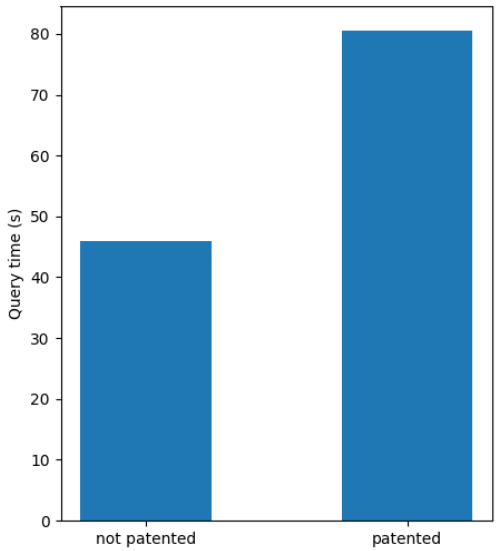


Figure 4.5: Query time of the patented and not patented parts of the data set created in 4.1

in this data set were screened using and modelled after a pharmacophore derived from Itraconazole. Itraconazole was also present in the data set.

After having selected these hits from the screening they were then checked for patents using the Patent KNIME node. The node is designed in such a way that for most applications it can be run without configuring. The Patent KNIME node found patents for Itraconazole and for a compound that was assumed to be novel. These patents were output in the patented data table shown in figure 4.7. The added bibliographic and frequency data is intended to give the user an overview of the patents found. The user may then, for example, choose to search again but only include results that mention the query molecule in the claims section.

The molecules without associated patents were output in a separate table. Splitting the table allows the user to continue working with the not patented molecules. A column that explicitly states that no patents were found for this molecule is appended.

4.3.2 Patent Searching for Generated Molecules

Another use case is demonstrated using a set of generated molecules. To get an overview of patents surrounding three scaffolds, 5 residues were permutated for each scaffold using an array of different substituents. The number of molecules came up to a total of 263,568 unique molecules. To avoid having to check these compounds manually, they were queried

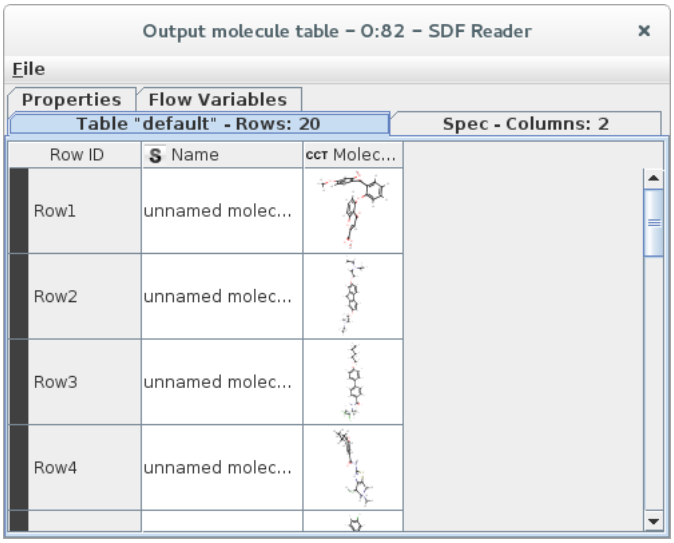


Figure 4.6: Hits imported into KNIME from an SDF

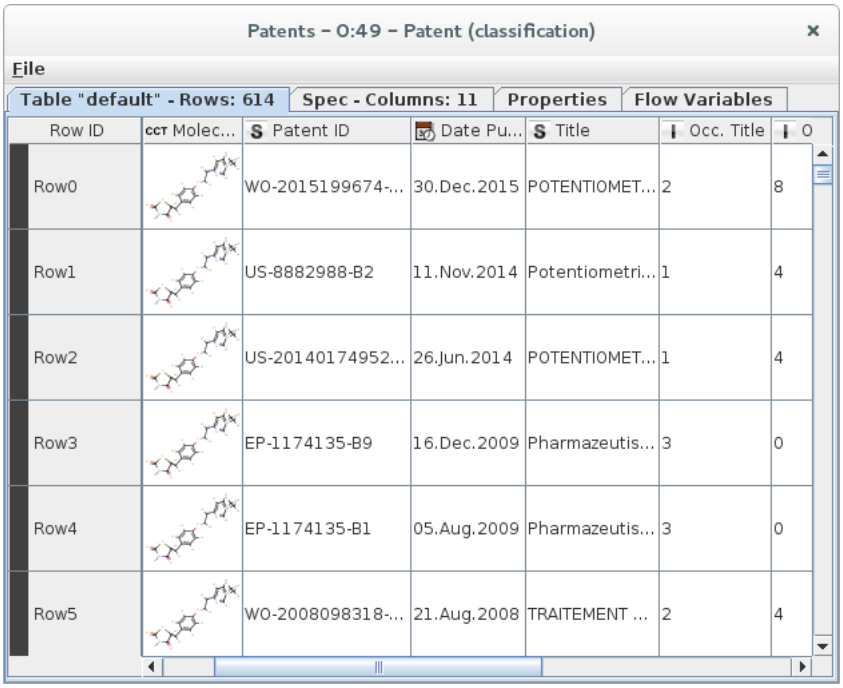


Figure 4.7: Molecule with its associated patents

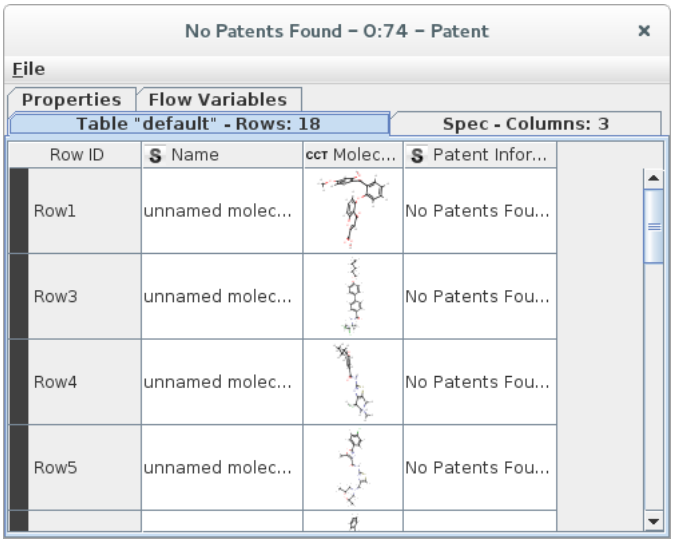


Figure 4.8: Table containing molecules without associated patents

Table 4.2: Results of querying generated molecules

Scaffold	total molecules	not patented	patented
1	13872	13559	313
2	124848	123862	986
3	124848	123950	898
total	263568	261371	2197

using the command line implementation of the patent searching algorithm. Results are shown in table 4.2

These results are meant to demonstrate 2 things: that even generated molecules may be patented and that the functionality can comfortably handle large data sets.

The fact that the majority of generated molecules are unpatented notwithstanding, the molecules that are patented will heavily influence patentability. The Markush structure and associated residues for a proposed patent of these scaffolds, would have to be revised to take this new information into account. Any overlap between a Markush structure contained in an older patent and a new one is considered prior art. If a patent claims molecules already present in an older patent it may not be granted. The worst case scenario would be that a patent would be granted if no examining party identifies the overlap, and then invalidated during litigation by a competitor. In this case less than 1% of the molecules were patented but this one percent will restrict the residues that may be claimed.

While the total number of molecules was 263,568, these were separated into 3 files according to their scaffold. In this case each scaffold query was started manually, however,

using the command line implementation allows for simple integration into, for example, a BASH script or any other scripting language a user prefers. This way a quarter of a million compounds can be queried automatically without user input beyond a simple for-loop. The query time this takes is discussed in section 4.2.2 using the same data set.

4.4 Further Development

The basic functionality of this implementation already provides unique application possibilities, nevertheless, it can definitely be expanded upon. Two possible expansions will be discussed in this section.

4.4.1 Query Customization

The current filtering options provide a simple way to create specific queries. Even so, advanced users accustomed to querying databases may want to build more complex and more specific SQL statements. Although in bulk processing the query does have to be general enough to be applicable to all molecules queried.

An example for a more complex query could be a combination of several patent classifications using Boolean operators, such as "AND", "NOT", and "OR". This way only patents corresponding simultaneously to several patent classification; or patents corresponding to one but not the other, could be retrieved. Another example would be the combination of field occurrences. That way only patents containing the specified molecule in the abstract and the claims could be retrieved.

4.4.2 Similarity Searching

A very helpful enhancement to the current exact match searching method would be a similarity searching mechanism. Exact match searching often fails to compensate for minute differences a skilled chemist would quickly recognize as irrelevant. The standardization of molecules using the InChI does already compensate for non canonicalized molecules, tautomers, and stereochemistry. Nonetheless, the classification experiment in section 4.1 showed that this does have its limits, for example, when it comes to salts.

Patent searching is rarely a very exact thing and a more permissive search, describing the patent space around a molecule, would reflect that. A very appropriate method of achieving this would be a mechanism based on Tanimoto similarity. An extensive substructure searching mechanism processing molecules in bulk would significantly impact query time, but a fingerprint comparison would not. Thus a fingerprint comparison is a high priority feature to implement in the future.

4.5 Summary and Outlook

Evaluation of the functionality created in this project exemplified aspects of its usage. The classification experiment showed that it performs its most basic task well. The over-

whelming majority of compounds were classified correctly. Limitations of the classification mostly derive from the exact match algorithm the functionality uses.

Performance is a vital aspect of an automated system. In general, the user can expect an average query time of approximately 1 second per compound. Initial overhead is overcome quickly and becomes negligible in larger data sets.

The use cases demonstrated practical applications of both implementations. Integration into a screening workflow is as simple as integrating any other filtering mechanism and a full virtual screening involving a patent search can be performed without leaving the KNIME environment.

The command line implementation was used to screen three scaffolds, and their proposed residues, for patents. This influenced the drawing up of Markush structures for these three scaffolds in pharmaceutical patents. The screening was achieved with minimal user input beyond the initial set up.

Development does not end when a system becomes functional, but is a continuous process of tweaking and improvement. The main features that would be an improvement to this functionality are custom querying and similarity searching. Custom querying would allow an advanced user to improve results by creating very specific queries. Similarity searching would also improve the system considerably, as it would compensate for some of the problematic parts of an exact matching algorithm and help describe the patent space surrounding molecules, as well as the patents directly associated with them.

The aim of this project was to create a patent searching functionality that can classify molecules into groups of patented and not patented molecules in an automated way, as well as make this functionality available to the end user. Both implementations achieved this and performed their function in real-life use cases, guiding research decisions. The most noticeable difference between this system and its predecessors is the automation and speed. Data sets that would be prohibitively large to query manually, can be queried automatically with minimal user input. This saves a lot of time even for smaller data sets. This way a basic patent searching element can become a part of a standard development workflow.

Acronyms

API Application Programming Interface. 26, 27, 29, 33, 35

CPC Cooperative Patent Classification. 9, 30

CPI Derwent Central Patent Index Chemical Code. 20

CWU Complex Working Unit. 8, 20–22, 32

EMBL-EBI European Bioinformatics Institute. 21

EPO European Patent Office. 6, 9, 22, 37

FTP File Transfer Protocol. 29

GREMAS Genealogical Retrieval by Magnetic Tape Storage. 20

GUI Graphical User Interface. 19, 22, 33, 35

HTTP Hypertext Transfer Protocol. 26

IDC Internationale Dokumentationsgesellschaft für Chemie GmbH. 20

IDE Integrated Development Environment. 32

InChI IUPAC International Chemical Identifier. 13, 14, 20, 22, 25, 28, 30, 38, 39, 46

IO Input/Output. 19, 25, 32, 35, 40, 42

IPC International Patent Classification. 8, 9, 30

IUPAC International Union of Pure and Applied Chemistry. 13, 20

JPO Japanese Patent Office. 22

RDBMS Relational Database Management System. 16, 17, 27

REST Representational State Transfer. 25, 26

SDF structure data file. 11, 42

SDK Software Development Kit. 32

SMILES Simplified Molecular Input Line Entry Specification. 12, 13, 28

SQL Structured Query Language. 17, 32, 46

SSL Secure Sockets Layer. 29

URI Unique Resource Identifier. 27

USPTO United States Patent and Trademark Office. 5, 6, 8, 9, 15, 22, 37

WIPO World Intellectual Property Organization. 5, 6, 8, 22, 31

Bibliography

[1] Roth, B. D., and Arbor, A. TRANS-6-[2-<3-OR 4-CARBOXAMIDO-SUBSTITUTED PYRROL-l- YL)ALKYL]-4-HYDROXYPY-RAN-2-0NE INHIBITORS OF CHOLESTEROL SYNTHESIS. 1987.

[2] WIPO, Guide to the International Patent Classification. 2016; `www.wipo.int/export/sites/www/classifications/ipc/en/guide/guide_ipc.pdf`.

[3] Song, C. H., and Han, J.-W. (2016) Patent cliff and strategic switch: exploring strategic design possibilities in the pharmaceutical industry. *SpringerPlus 5*, 692.

[4] Wikipedia, Simplified molecular-input line-entry system. 2017; `www.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system`.

[5] InChI Trust, Technical FAQ. `www.inchi-trust.org/technical-faq`.

[6] Jewett, T. Database Design - Many-to-many. 2002; `www.tomjewett.com/dbdesign/dbdesign.php?page=manymany.php`.

[7] Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. (2009) KNIME - The Konstanz Information Miner. *SIGKDD Explorations 11*, 26–31.

[8] Papadatos, G., Davies, M., Dedman, N., Chambers, J., Gaulton, A., Siddle, J., Koks, R., Irvine, S. A., Pettersson, J., Goncharoff, N., Hersey, A., and Overington, J. P. (2016) SureChEMBL: A large-scale, chemically annotated patent document database. *Nucleic Acids Research 44*, D1220–D1228.

[9] EMBL-EBI, SureChEMBL. `www.surechembl.org/search`.

[10] Brecher, J. (1999) Name=Struct: A Practical Approach to the Sorry State of Real-Life Chemical Nomenclature. *Journal of Chemical Information and Computer Sciences 39*, 943–950.

[11] Downs, G. M., and Barnard, J. M. (2011) Chemical patent information systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science 1*, 727–741.

[12] Senger, S., Bartek, L., Papadatos, G., and Gaulton, A. (2015) Managing expectations: assessment of chemistry databases generated by automated extraction of chemical structures from patents. *Journal of Cheminformatics 7*, 49.

[13] Bregonje, M. (2005) Patents: A unique source for scientific technical information in chemistry related industry? *World Patent Information 27*, 309–315.

[14] Heifets, A., and Jurisica, I. (2012) SCRIPDB: a portal for easy access to syntheses, chemicals and reactions in patents. *Nucleic Acids Research 40*, D428.

[15] Hemphill, C. S., and Sampat, B. N. (2012) Evergreening, patent challenges, and effective market life in pharmaceuticals. *Journal of Health Economics 31*, 327–339.

[16] Tucker, G. T. (2000) New drug classes: Chiral switches. *The Lancet 355*, 1085–1087.

[17] Burdon, M., and Sloper, K. (2003) The art of using secondary patents to improve protection. *Journal of Medical Marketing 3*, 226–238.

[18] WIPO, Patents. `www.wipo.int/patents/en`.

[19] WIPO, WIPO - World Intellectual Property Organization. `www.wipo.int`.

[20] WIPO, ST.1 - Recommendation concerning the minimum data elements required to uniquely identify a patent document. 2001; `www.wipo.int/export/sites/www/standards/en/pdf/03-01-01.pdf`.

[21] WIPO, ST.3 - Recommended standard on two-letter codes for the representation of states, other entities and intergovernmental organizations. 2011; `www.wipo.int/export/sites/www/standards/en/pdf/03-03-01.pdf`.

[22] WIPO, ST. 18 - Recommendation concerning patent gazettes and other patent announcement journals. 1997; `www.wipo.int/export/sites/www/standards/en/pdf/03-18-01.pdf`.

[23] Danzon, P. M., and Furukawa, M. F. (2006) MarketWatch: Prices and availability of biopharmaceuticals: An international comparison. *Health Affairs 25*, 1353–1362.

[24] USPTO, Manual of Patent Examining Procedure. 2017; `www.uspto.gov/web/offices/pac/mpep`.

[25] Arrangement of application elements, 37 C.F.R. 1.77. 2013.

[26] Application data sheet, 37 C.F.R. 1.76. 2012.

[27] Content of information disclosure statement, 37 C.F.R. 1.98. 2004.

[28] Title and abstract, 37 C.F.R. 1.72. 2013.

[29] Specification, 35 U.S.C. 112. 2015.

[30] Conditions for patentability; novelty, 35 U.S.C. 102. 2015.

[31] Drawings, 35 U.S.C. 113. 1999.

[32] USPTO, Complex Work Unit Pilot Program. `www.uspto.gov/patent/initiatives/complex-work-unit-pilot-program`.

[33] WIPO, IPC Publication. 2017; `www.wipo.int/classifications/ipc/ipcpub`.

[34] USPTO„ and EPO, Cooperative Patent Classification. 2017; `www.cooperativepatentclassification.org`.

[35] Druss, B. G., Marcus, S. C., Olfson, M., and Pincus, H. A. (2004) Listening to generic prozac: Winners, losers, and sideliners. *Health Affairs 23*, 210–216.

[36] DiMasi, J. A., Grabowski, H. G., and Hansen, R. W. (2016) Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics 47*, 20–33.

[37] Akhondi, S., Kors, J., and Muresan, S. (2012) Consistency of systematic chemical identifiers within and between small-molecule databases. *Journal of Cheminformatics 4*, 35.

[38] BIOVIA, *CTFile Formats*; 2005.

[39] Shoichet, B. K. (2004) Virtual screening of chemical libraries. *Nature 432*, 862–865.

[40] Weininger, D. (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci. 28*, 31–36.

[41] O'Boyle, N. M. (2012) Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics 4*, 22.

[42] Heller, S., McNaught, A., Pletnev, I., Stein, S., and Tchekhovskoi, D. (2015) InChI, the IUPAC International Chemical Identifier. *J Cheminf 7*.

[43] Kettle, J. G., Ward, R. A., and Griffen, E. (2010) Data-mining patent literature for novel chemical reagents for use in medicinal chemistry design. *Med. Chem. Commun. 1*, 331–338.

[44] Bone, R. G. A., and Kendall, J. T. (2008) Markush under threat : US PTO considers alternatives. *Industrial Biotechnology 4*, 246–252.

[45] Eller, G. A. (2006) Improving the quality of published chemical names with nomenclature software. *Molecules 11*, 915–928.

[46] Open PHACTS Foundation, Open PHACTS Support Portal: SureChEMBL. 2016; http://support.openphacts.org/support/solutions/articles/4000079208-surechembl.

[47] Hughes, J. P., Rees, S. S., Kalindjian, S. B., and Philpott, K. L. (2011) Principles of early drug discovery. *British Journal of Pharmacology 162*, 1239–1249.

[48] DB-Engines Ranking. 2017; www.db-engines.com/en/ranking.

[49] ISO, ISO/IEC 9075-1:2016. 2016; www.iso.org/standard/63555.html.

[50] Warr, W. A. (2012) Scientific workflow systems: Pipeline Pilot and KNIME. *Journal of Computer-Aided Molecular Design 26*, 801–804.

[51] Knime.com AG, KNIME Noding guidelines. www.tech.knime.org/files/development/noding_guidelines.pdf.

[52] Inte:Ligand GmbH, Ligandscout 4.0. www.inteligand.com.

[53] Wolber, G., and Langer, T. (2005) LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters. *Journal of Chemical Information and Modeling 45*, 160–169.

[54] Adams, S. JNI-InChI. 2010; www.jni-inchi.sourceforge.net.

[55] Simmons, E. S. (1984) Central Patents Index Chemical Code: a user's viewpoint. *Journal of Chemical Information and Computer Sciences 24*, 10–15.

[56] Rössler, S., and Kolb, A. (1970) The GREMAS System, an Intergral Part of the IDC System for Chemical Documentation. *Journal of Chemical Documentation 10*, 128–134.

[57] Schmuff, N. R. (1991) A comparison of the MARPAT and Markush DARC software. *Journal of Chemical Information and Computer Sciences 31*, 53–59.

[58] IBM, IBM BAO strategic IP insight platform (SIIP). www.ibm.com/services/us/gbs/bao/siip.

[59] IBM, IBM Contributes Data to the National Institutes of Health to Speed Drug Discovery and Cancer Research Innovation. 2011; www.ibm.com/press/us/en/pressrelease/36180.wss.

[60] Elsevier, Reaxys. www.reaxys.com.

[61] Chemical Abstracts Service, SciFinder. http://scifinder.cas.org.

[62] Valko, A. T., and Johnson, A. P. (2009) CLiDE Pro: The Latest Generation of CLiDE, a Tool for Optical Chemical Structure Recognition. *Journal of Chemical Information and Modeling 49*, 780–787.

[63] Chambers, J., Davies, M., Gaulton, A., Hersey, A., Velankar, S., Petryszak, R., Hastings, J., Bellis, L., McGlinchey, S., and Overington, J. P. (2013) UniChem: A unified chemical structure cross-referencing and identifier tracking system. *Journal of Cheminformatics 5*.

[64] Williams, A. J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E. L., Evelo, C. T., Blomberg, N., Ecker, G., Goble, C., and Mons, B. (2012) Open PHACTS: Semantic interoperability for drug discovery. *Drug Discovery Today 17*, 1188–1198.

[65] Open PHACTS Foundation, Documentation Main Page. `http://dev.openphacts.org/docs`.

[66] Siddle, J., Zavarin, Y., and Papadatos, G. surechembl-data-client. 2016; `www.github.com/chembl/surechembl-data-client`.

[67] InChI Trust, InChI Trust. `www.inchi-trust.org`.

[68] The Eclipse Foundation, Eclipse. `www.eclipse.org`.

[69] Thomson Reuters, Thomson Reuters Integrity. `http://integrity.thomson-pharma.com`.

[70] University of California San Francisco, ZINC Database. `http://zinc.docking.org`.