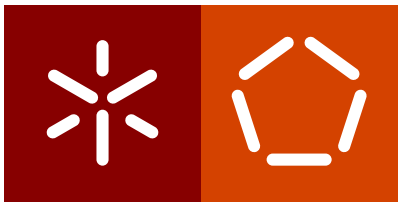
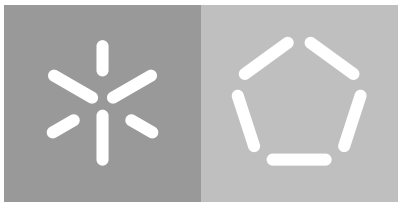


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Sofia Maria Alves Faria

Development of Algorithms for the Analysis and Data Mining of Chemical Compound Prices

May 2020



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Sofia Maria Alves Faria

Development of Algorithms for the Analysis and Data Mining of Chemical Compound Prices

Master dissertation
Master Degree in Bioinformatics

Dissertation supervised by
Prof. Miguel Francisco de Almeida Pereira da Rocha
PhD Paulo Ricardo Carvalho Vilaça

May 2020

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

ACKNOWLEDGEMENTS

Em primeiro lugar gostaria de agradecer ao Professor Dr. Miguel Rocha pela oportunidade e disponibilidade que me proporcionou. Obrigada por todo o seu auxílio no decorrer desta tese.

Ao Paulo por todo o esforço, dedicação e paciência que demonstrou neste trabalho para me ajudar a compreender todas as novas competências e obstáculos que me foram impostos no mundo da informática. Um enorme obrigado e espero que continue a ter muito sucesso em todo o seu caminho.

Agradeço também ao Hugo, que embora não tenha estado presente durante toda a realização da tese, ainda assim apoiou-me bastante no início e mostrou grande disponibilidade. O meu sincero obrigado.

À SilicoLife um obrigado pela oportunidade de realizar esta dissertação num novo ambiente, mostrando-me o mundo do trabalho.

A todos os meus amigos, em especial à Inês (de Castro) e ao Alexandre, por me aturarem nos meus stresses e tornarem este tempo mais fácil. Por todas as ajudas neste projeto, quando os problemas apertavam mais. E por todo o tempo que tiraram para me ouvir a desabafar e me ajudar a superar os maus dias. Sem vocês, teria estado muito mais sozinha neste momento stressante, por isso um enorme obrigado!

À Catarina Costa, Maria Inês e Catarina Carvalho por me desculparem e compreenderem todas as vezes que tive de dizer que não aos passeios e convites. Agradeço a vossa amizade de anos que me faz acreditar que certas pessoas são para sempre, sem vocês não seria quem sou hoje. Obrigada!

A toda a minha família, pelo o apoio que me deram, à minha irmã e aos meus pais por me ajudarem a esquecer os stresses passados neste ano com a vossa companhia e por me ajudarem pessoalmente e financeiramente a superar os obstáculos encontrados no caminho. Por todos os conselhos ao longo desta vida toda, um enorme obrigado!

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

Nowadays, the products deriving from the biotechnology industry have become quite valuable in the world market. Hence, it is highly advantageous to find out how the prices of the different chemical compounds needed for biotechnological processes behave in the bioeconomy. The SISBI project was developed to allow the retrieval and collection of different prices associated with certain chemical compounds through different available sources and databases. With access to this information, some behaviours and patterns can be detected in the price variations, indicating other relevant knowledge, such as the biotechnological interest of this compound in the field. However, it is necessary to take into account that SISBI data, although relevant, have inconsistencies that do not support an efficient analysis of these data, which is the case for the existence of duplicates, different units and problems in the price integration. As a result, this study developed algorithms to identify and solve these problems and to analyze the prices of compounds through time series. To effectively evaluate these data, a new database, *bioanalysis*, was built based on the data from the SISBI project. Then, several preprocessing methods were applied, including the elimination of duplicates, conversion of units, removal of defective and inconsistent prices, which led to the solution of the various complications encountered. Consequently, once the data was prepared for analysis, the prices pertaining to two specific metabolites, 4-aminopyridine and methane, were examined. Thus, different price variations over time were compared between different configurations (quantity + unit) of the same metabolite and between different metabolites. These variations were divided by the different price providers to identify any specific relationship or pattern depending on where the data originate. However, in this study, no particularly cheap provider was detected between 4-aminopyridine configurations or between the two metabolites. The only association found occurred only between certain methane configurations. In addition, the price variations analyzed are mostly constant, and when they are not, they do not show any pattern or seasonality. These results revealed that, using only the prices available to date, no correlation was determined by identifying the providers associated with low prices when comparing different metabolites or configurations.

Key-words: Biotechnology, Chemical Compounds, Time Series, Algorithms, Preprocessing

RESUMO

Atualmente, os produtos resultantes da indústria biotecnológica têm-se tornado bastante importantes no mercado mundial. Desta forma, é altamente vantajoso descobrir como se comportam os preços dos diferentes compostos químicos necessários para os processos biotecnológicos na bioeconomia. O projeto SISBI foi desenvolvido de modo a permitir a recolha e coleção de diferentes preços associados a determinados compostos químicos através de diversas fontes e bases de dados disponíveis. Com o acesso a esta informação, alguns comportamentos e padrões podem ser detetados na variação dos preços, indicando outras informações relevantes como o interesse biotecnológico desse composto na área. No entanto, é necessário ter em conta que os dados da SISBI, embora relevantes, apresentam inconsistências que não permitem analisar de forma eficaz estes dados, como é o caso da existência de duplicados, diferentes unidades e problemas na integração dos preços. Por esta razão, este estudo comprometeu-se a desenvolver algoritmos para identificar e resolver estes problemas, e para analisar os preços dos compostos através de séries temporais. De modo a avaliar eficazmente estes dados, uma nova base de dados, *bioanalysis*, foi construída com base nos dados do projeto SISBI. De seguida, diversos métodos de pré-processamento foram realizados, incluindo a eliminação de duplicados, conversão de unidades, remoção de preços defeituosos e não consistentes, que levaram à resolução das várias complicações encontradas. Por consequência, uma vez os dados prontos para a análise, os preços pertencentes a dois metabolitos específicos, 4-aminopiridina e metano, foram examinados. Assim, diferentes variações de preços ao longo do tempo foram comparadas entre diferentes configurações (quantidade + unidade) do mesmo metabolito e entre diferentes metabolitos. Estas variações foram agrupadas pelas diferentes fontes de preços de modo a identificar alguma relação ou padrão específico dependente ao local de onde os dados provenieram. Contudo, neste estudo, não se detetou nenhuma fonte em particular consistentemente barata entre configurações do 4-aminopiridina ou entre os dois metabolitos. A única associação descoberta ocorreu apenas entre determinadas configurações do metano. Para além disso, as variações dos preços analisados são maioritariamente constantes, e quando não são, não demonstram nenhuma tendência ou sazonalidade. Estes resultados revelaram que, utilizando apenas os preços disponíveis até à data, nenhuma correlação foi determinada ao identificar as fontes associadas a preços baixos quando comparando diferentes metabolitos ou configurações.

Palavras-chave: Biotecnologia, Compostos Químicos, Séries Temporais, Algoritmos, Pré-processamento

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

A

ACF Autocorrelation Function.

AO Additive Outlier.

API Application Programming Interface.

APSDS Auto Power Spectral Densities.

AR Autoregressive.

ARIMA Autoregressive Integrated Moving-Average.

ARMA Autoregressive Moving-Average.

ART Adaptive Resonance Theory.

C

CAS Chemical Abstracts Service.

D

DI Department of Informatics.

DNA Deoxyribonucleic Acid.

E

EC European Comission.

EU European Union.

F

FS Feature Selection.

I

ID Identifier.

IDE Integrated Development Environment.

INCHI International Chemical Identifier.

IO Innovational Outlier.

IQR Interquartile Range.

IS Instance Selection.

IUPAC International Union of Pure and Applied Chemistry.

J

JPA Java Persistence API.

JSON JavaScript Object Notation.

L

LOWESS Locally Weighted Regression.

M

MA Moving-Average.

MBIOINF Master in Bioinformatics.

MLP Multi-Layer Perceptrons.

O

OECD Organization for Economic and Cooperative Development.

P

PACF Partial Autocorrelation Function.

R

REST Representational State Transfer.

RNA Ribonucleic Acid.

S

SARIMA Seasonal Autoregressive Integrated Moving-Average.

SISBI Intelligent Decision Support System for Industrial Biotechnology.

SMILES Simplified Molecular-Input Line-Entry System.

SPRT Sequential Probability Ratio Test.

U

UM University of Minho.

URL Uniform Resource Locator.

V

VAR Vector Autoregressive.

VARMA Vector Autoregressive Moving-Average.

1

INTRODUCTION

This dissertation describes the Master's thesis work developed in the context of the *Master in Bioinformatics (MBIOINF)* assured by SilicoLife and the *Department of Informatics (DI)* at *University of Minho (UM)*.

1.1 CONTEXT AND MOTIVATION

Biotechnology companies are looking for solutions to develop competitive bioprocesses able to address the production of compounds of interest in the world market. Over the last few years, SilicoLife and the UM have been developing an *Intelligent Decision Support System for Industrial Biotechnology (SISBI)*. This system allows to boost the metabolic engineering platforms at the company, promoting the development of tools for the extraction and integration of knowledge, to provide an adequate support to reach decisions in the context of Industrial Biotechnology, as it is the case of the selection of products with high added value, of adequate host microorganisms or of advantageous raw materials to be used in target bioprocesses.

This project led to the creation of a knowledge hub including a set of knowledge extraction and integration modules, which took into account diverse sources of available technical and economic data. Among the data collected in this context, the availability of prices of compounds and its evolution along time is of foremost importance, since it allows to identify compounds that show promising trends, and supports calculations for the economic viability of biotechnological processes towards their production.

Nowadays, Biotechnology is a strong field worldwide. The necessity to improve it and create advantages is something important to take into account. Thus, the identification of compounds with biotechnological interest and the assessment of how the prices of these compounds are evolving throughout the time is an important topic of study. With the compilation of all these prices it is possible to determine which are the trends in a specific period of time and what is the viability in terms of the financing for the corresponding biotechnological processes for their production.

With this aim in mind, the SISBI project has allowed the development of tools to collect price information from distinct sources, spanning a few thousand compounds over the last years. These can be visualized in the web-based platforms developed within the same project.

However, the analysis of these data has shown that there seem to be many inconsistencies in the prices collected from different sources, which may be due to issues in data collection, problems with different sample sizes/units, among others. The manual analysis of these possible inconsistencies is a very labour intensive task, and the development of automated algorithms would be highly desirable.

Also, the available data, as the number of time points in the series of prices increases, has a large potential for data analysis and data mining algorithms seeking to extract relevant patterns for the assessment of the compounds' relevance in future biotechnological processes.

1.2 OBJECTIVES

In the context provided above, the main aim of this work will be to allow the development of algorithms able to analyse time series of compound prices from different sources, identifying possible inconsistencies or outliers and providing tools for the analysis of these data and for data mining of relevant patterns.

In more detail, the technological/scientific objectives are the following:

- Review the state of the art in the analysis of chemical compounds prices and data analysis/mining for time series data;
- Study in detail the available data sets and computational tools built within the SISBI project;
- Design and implement algorithms to detect inconsistent sources of prices for a single compound and for sets of compounds;
- Design and implement algorithms for the analysis/data mining over time series of compound prices;
- Write the thesis with the results of the previous topics.

1.3 THESIS OUTLINE

This dissertation is divided into 6 chapters: Chapter 1 "Introduction", Chapter 2 "State of the Art", Chapter 3 "SISBI and Bioeconomics Platform", Chapter 4 "Development", Chapter 5 "Case Studies" and Chapter 6 "Conclusion".

In the first chapter, a part of the background of this project, the motivation, and this subject interest is discussed, as well as the objectives aimed to achieve until the completion of this dissertation.

Subsequently, the second chapter, not only discusses the world of the Biotechnology industry, and its influence on Europe and the economy, but also what are time series, the different algorithms already conceived as solutions to other problems and the disclosure of some price sources of chemical compounds essential for this study, which have already been used in prior tasks or may be implemented in the future.

The third chapter contextualizes the previous work done by SilicoLife. Afterwards, this work's development is explained in the fourth chapter, including steps such as, the acquisition of the data in question, the conception of a data base in *Java*, the problems encountered in the data and its preprocessing and solution in *Python*.

The fifth chapter reviews the outcome of the previous preprocessing methods in specific case studies, as well as, the price variation analysis, namely the cheapest providers identified in these particular situations.

Finally, the chapter six, sums up the issues detected throughout the dissertation and the reasons why in the previous chapter some results occurred. Also, other ideas for future work are presented.

2

STATE OF THE ART

In this work, several compound prices are gathered by a significant number of relevant sources. Consequently, it is necessary to understand those price sources and how their evolution and update can be influenced or influence the biotechnological world — recording compound prices with a time order, which results in the creation of time series. However, there are many price inconsistencies throughout the ordered data. The assessment of algorithms that can automatically evaluate possible outliers in these time series, and check their values consistency is crucial, as well as the acknowledgement and understanding of the prior algorithms already created for this purpose. This section will review the state-of-the-art in these aspects.

2.1 BIOTECHNOLOGY INDUSTRY

2.1.1 *The Biotechnology Industry in the World*

Looking at several articles, it is possible to conclude that there is no consensus in the definition of biotechnology. Since Portugal and a large number of European countries make part of the *Organization for Economic and Cooperative Development (OECD)*, we will use the definition of biotechnology given by this organization. According to Van Beuzekom and Arundel (2009), Biotechnology is "the application of science and technology to living organisms, as well as parts, products and models thereof, to alter living or non-living materials for the production of knowledge, goods and services".

There are a number of different types of biotechnology techniques. These are divided into those handling *Deoxyribonucleic Acid (DNA)*/*Ribonucleic Acid (RNA)* (the coding), proteins and other molecules (the functional blocks), cell and tissue culture and engineering, process biotechnology techniques, gene and RNA vectors, bioinformatics and nanobiotechnology (Van Beuzekom and Arundel, 2009).

In consequence of this definition, biotechnology companies are defined as those involved in biotechnology by adopting one of the techniques described above, with a purpose to provide services or goods, that are useful to the society, and/or to perform biotechnology

research and development (Van Beuzekom and Arundel, 2009). Biotechnology can therefore be seen as the science that utilizes living things to produce goods and services, if this involves manipulating and modifying organisms to develop applications that are new and useful in a number of sectors, like primary production, health and industry (Van Beuzekom and Arundel, 2009; Dahms, 2004; McCormick and Kautto, 2013).

With the past progress in genomics and the subsequent advances made in relation to other omics families, not forgetting the improvement in the knowledge about the humans' development and aging, the biotechnology field will rapidly revolutionize our society. As reported by Dahms (2004), this field is in the beginning of a "technology curve" that presents limitless potential.

The development of a sustainable industry can be achieved with biotechnology, since it presents unique opportunities for this purpose (Lokko et al., 2018). The Biotechnology Industry has a significant impact on already existing technologies, and consequently these improved technologies have an impact on other industries (Dahms, 2004). One of the greatly impacted industries is the chemical one (Lokko et al., 2018; Gavrilescu and Chisti, 2005; Dahms, 2004; McCormick and Kautto, 2013).

The way people live now was greatly allowed by the chemical industry (Gavrilescu and Chisti, 2005). Nevertheless, this industry relies heavily on nonrenewable energy and resources, like fossil fuels, which affects the safety and health of the future generations (Lokko et al., 2018; Gavrilescu and Chisti, 2005). As the human population and the world evolves, many challenges emerge, being at an economic level, social level and also at an environmental level (Lokko et al., 2018). The transition to biotechnology can solve the dependence on expensive fossil resources, since it is capable of handling renewable raw materials and energy, designing a sustainable production (Lokko et al., 2018; Hermann et al., 2007; Dornburg et al., 2008). Besides, most of the products developed by this industry have greater advantages compared to other industries (Lokko et al., 2018; Gavrilescu and Chisti, 2005; Dornburg et al., 2008; Dahms, 2004). According to Hermann et al. (2007), the chemicals based on industrial biotechnology have higher advantages at the economic level than their petrochemical counterparts, improving the economics of production.

To the economy where the basics for materials, chemicals and energy are obtained from renewable biological resources, such as plants and animals sources, it is given the name of bioeconomy (McCormick and Kautto, 2013). As mentioned in McCormick and Kautto (2013), biotechnology is an essential component of the bioeconomy, the latter being involved in a great number of sectors, for instance agriculture, forestry or fishery, among others. The benefits of the bioeconomy and biotechnology are immense, including the increasing of the provision, security and sustainability of the production of food and food ingredients, the increasing on quality of the water and the improvement on the health of people and animals. All of this is possible while also reducing the greenhouse gases emissions and the supply of

renewable energy, since there is a decrease of the fossil resources dependence and a smarter management of these renewable resources (McCormick and Kautto, 2013).

Because of all of these advantages, the biotechnology industry has become one of the most impactful and growing industry fields in the world (Dornburg et al., 2008). Substantial advances can be expected in this field at regional, national and international scales, generating a rise on the urban and rural employment (McCormick and Kautto, 2013; Dornburg et al., 2008). According to McCormick and Kautto (2013), it is necessary to take into account that one of the biggest challenges is increasing the activities, such as biomass production, in a way that the major goals of sustainability continue to be met.

Furthermore, there is a lot of competition when talking about the biotechnology industry. This area is unique, since it is a revolutionary technology (Quintana-Garcia and Benavides-Velasco, 2004; Powell and Brantley, 1992). According to Quintana-Garcia and Benavides-Velasco (2004), the continuous innovation alone in these firms can maintain a competitive advantage, as it develops patentable products that can turn current products into outdated products in a short period of time.

2.1.2 *Bioeconomy and Biotechnology Industry in Europe*

According to McCormick and Kautto (2013), Europe is the global leader and pioneer of a significant number of fields related to biosciences and its technologies. However, other countries, such as USA and China, are starting to invest heavily in these fields. Hence, the *European Comission (EC)* dreads that the long term competitiveness of Europe is at risk (McCormick and Kautto, 2013).

There is interest in the development of the bioeconomy in an increasing number of countries. This interest occurs due to the necessity of a sustainable supply of food, fuel and raw materials considering there will be a high demand for these supplies in the next decades to satisfy the exponential growth of the population and also due to the need to successfully face enormous challenges, like climate changes and energy security (McCormick and Kautto, 2013). The *European Union (EU)* and the OECD underline the inherent value that the biological material has (McCormick and Kautto, 2013). The EC, in 2002, stated that probably one of the most "promising frontier technologies" are the life sciences and biotechnology (European Commission, 2002). According to the EC, the bioeconomy is one of the vital elements for a safer and greener growth. In 2011, they led a public consultation, where most of the people had a positive perspective about the bioeconomy in Europe, expecting that most of its possible benefits would be achieved by 2020 or 2030 (European Commission, 2011; McCormick and Kautto, 2013).

However, as stated in European Commission (2011), most of the people also believed that there was a great number of risks linked with this topic, as well as possible extreme

exploitation of natural resources and repercussions on food security. Thus, in 2012, the EC established a strategy that had the purpose of improving the basic knowledge about bioeconomy, of emboldening the innovation, so that the natural resources production would increase sustainably, and of supporting the development of production systems that prevent climate changes. (European Commission, 2012; McCormick and Kautto, 2013).

As believed by the EC, the bioeconomy will boost the request for a: sustainable supply of food, raw materials, and fuel; the competitiveness and productivity of Europe; and the life quality of the European citizens (McCormick and Kautto, 2013). Consequently, this progress will possibly obligate considerable changes in technology and market development, as well as in the industrial processes, since it may affect the patterns of consumption and production in Europe. The EU is also concerned that this growth in the bioeconomy and biotechnology may have undermining effects in the erosion of the Earth, in the loss of biodiversity, and in the shortage of food, since this can implicate a massive amount of used biomass (McCormick and Kautto, 2013).

2.2 CHEMICAL COMPOUNDS PRICES

The prices of several chemical compounds and their global market are critical factors for the generation of innovative bioprocesses, as they constrain the selection of the new possible chemical compounds to produce (Pollard and Woodley, 2007; Woodley, 2017). Indeed, the conjugation of the prices, and their predicted evolution, with the predicted production costs is the basis for the evaluation of the economic viability of biotechnological processes. These processes greatly influence the development of the biotechnology industry. So a viable bioeconomy can help prevent some of the negative consequences already discussed, such as the possibility of using massive amounts of biomass, ruining the goal of a sustainable industry (McCormick and Kautto, 2013).

Hence, the biotechnology industry players need to track the prices of the chemical compounds they produce and also the ones they use in their processes to create and update business plans and make informed strategic decisions. There are numerous databases and online platforms that compile these prices. Some of those are listed below:

- *MolPort* (https://www.molport.com/), a chemical marketplace that collects data from all the major providers;
- *eMolecules* (https://www.emolecules.com/), a chemical database with pricing and availability information from chemical suppliers;
- *ChemExper* (http://www.chemexper.com/), a chemical database with supplier information. However, it does not include the prices, so for more details it is necessary to search on the company website;

- *Mcule* (https://mcule.com/), a high quality compound database, that provides purchasable compound information;
- *ChemSpace* (https://chem-space.com/), the database of chemical building blocks, with the goal to create a "one-stop-shop" for searching and ordering building blocks online;
- *ChemSpider* (http://www.chemspider.com/), a free chemical structure database that provides fast access to structures, properties, and additional information, such as suppliers;
- *Fisher Scientific* (https://www.fishersci.com/us/en/home.html), a laboratory supplier that provides information on chemicals, supplies, and services used in scientific research;
- *iChemical* (http://www.ichemical.com/), an online sourcing and trading service platform that presents accurate price forecast and indicates the suppliers that are suitable for the sourcing budget;
- *LabNetwork* (https://www.labnetwork.com/), a global eCommerce platform with the aim of connecting suppliers and buyers for research products;
- *TCI Chemicals* (https://www.tcichemicals.com/en/pt/), a chemical manufacturer that supplies high-quality organic reagents.

These online science marketplaces allow an exchange of prices and chemical products' information between the associated consumers and sellers (Standing et al., 2010). Thus, they should also facilitate the identification of possible trading partners and the transaction execution (Standing et al., 2010).

Consequently, the online science marketplaces described above are essential to this work and a vital part of making the prices of the chemical compounds more accessible. The prices can be found resorting to different chemical compounds identifiers, such as *Chemical Abstracts Service (CAS)* number, chemical name, *International Union of Pure and Applied Chemistry (IUPAC)* name, *International Chemical Identifier (InChI)* keys, *Simplified Molecular-Input Line-Entry System (SMILES)*, chemical structure, MFCD numbers (MDL's unique *Identifier (ID)* number), catalog number, among others (some of these identifiers will be explained in further detail below). In addition, a few of these marketplaces, such as *MolPort*, *eMolecules*, and *ChemSpace*, have advanced filter options on the search engine, making this process effortless for potential clients (MolPort; eMolecules; ChemSpace).

However, the search for the compounds prices is arduous to be made manually, since it is a very time-consuming task. Moreover, not all online platforms and databases available are reliable, as discussed above. Some of these marketplaces have different operations. The *ChemSpace* marketplace functions as an open platform, allowing any supplier to join

without a cost and sell its compounds without having to pay a service fee. However, its role is to provide a comprehensive database of compounds that can be accessed in the e-commerce of the supplier, not being generally involved in managing operational activities on the side of buyers or suppliers (ChemSpace; BiopharmaTrend.com). In contrast, the *eMolecules* marketplace does get involved in operational activities, offering capabilities for building workflows for approval and validation of the purchases between buyers and vendors (eMolecules; BiopharmaTrend.com). On the other hand, *MolPort* makes sure to ensure that the prices for the listed compounds and the prices of the original suppliers are kept at the same level, not varying too much. Also, it takes care of some operations and logistics to help buyers (MolPort; BiopharmaTrend.com).

As recognized in this section, there is a wide variety of marketplaces. This leads to numerous prices fetched by these platforms and databases, and consequently, to a demand for the fundamental analysis of these prices, so that some useful information can be possibly retrieved. In addition, considering the price values presented in these marketplaces, take into account that the higher the amount of the compound sold, the less expensive it is, as guaranteed in the law of supply and demand by Gale (1955).

2.3 TIME SERIES

While studying the usual behaviour of a system, the data observed is mostly collected taking into account the order in time of the data being generated. For a time-ordered data sequence, there is the denomination of *time series* (Dasgupta and Forrest, 1996; Wei, 2013). A time series is the outcome of a stochastic process, a family of random variables indexed by time (Jazwinski, 2007), that can be continuous if the observations are taken continuously throughout the time, or can be discrete if the observations are collected in distinct time points (Brockwell et al., 2002; Wei, 2013; Chatfield, 2003). Hence, since the prices of the chemical compounds are collected throughout the time and can change depending on the date of their collection, they are characterized as a time series.

Considering a time series as a family of variables indexed by time, these data are typically correlated. On that account, statistical methods that depend on random samples cannot be used, even though almost all the statistical methods developed are random sample oriented (Wei, 2013). Thus, for *time series analysis*, different statistical methods are required (Wei, 2013; Chatfield, 2003).

Time series analysis encloses methods that analyze data, so it becomes possible to extract valuable statistics and other attributes (Wei, 2013; Brockwell et al., 2002). As reported by Chatfield (2003), the goals of analysing a time series may be divided into description, explanation, prediction, and control.

While attempting to describe a time series, the first and more important step to take is to plot a time series graph (Chatfield, 2003). In this type of graph the x-axis contains the time points and the respective time intervals, and the y-axis contains the observed values (Figure 1). The difference between a time series plot and a standard x-y graph is that in the first plot mentioned the x-axis can only have one independent variable, the time, while in the other graphs, other independent variables are allowed (Chatfield, 2003; Chandler and Scott, 2011).

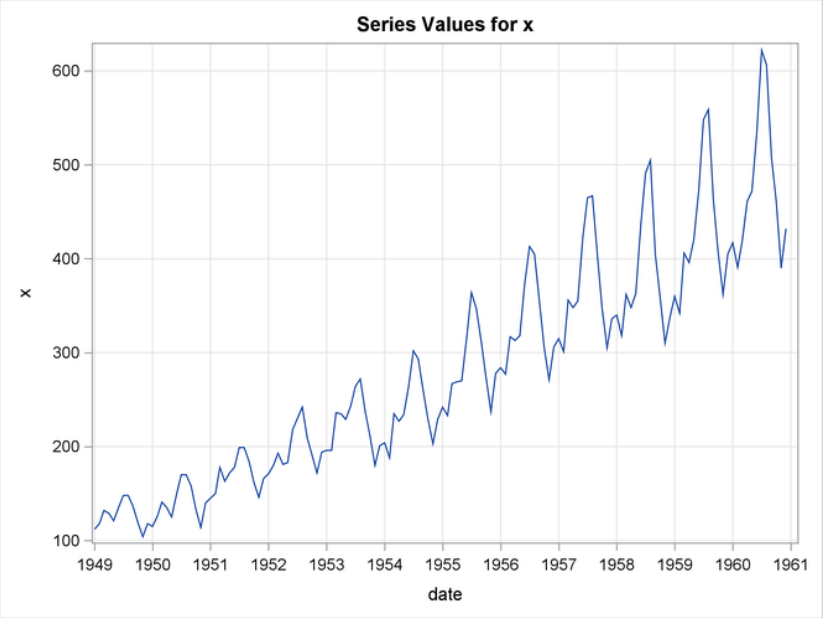


Figure 1: Representation of an example of a time series plot from Airline Passenger Series. The x-axis depicts the time in which the data was generated/collected and the y-axis represents the possible values of x. Retrieved from SAS Institute Inc. (2019).

The advantage of the plot's creation is the straightforward observation of various fluctuations and points of interest (Brockwell et al., 2002). That way the plot will help to easily observe simple measures that might describe the variations in the time series, such as seasonal effects, trends, and turning points (Chatfield, 2003). Furthermore, a time series plot can also disclose apparent outliers, but more on that in the next subsection.

Moreover, while creating the time series plot, it might be helpful to transform the data. As mentioned by Chatfield (2003), the data transformation might be performed due to some particular reasons: to stabilize the variance, to turn the seasonal effect additive (make the seasonal effect constant from time period to time period) and to give a normal distribution to the data. However, if the transformation is realized to stabilize the data, it should only occur if there is a trend displayed. If not, the transformation will be useless. These transformations can be accomplished by, for instance, performing square roots or logarithms in the data (Chatfield, 2003).

In addition, depending on the time series, more complex models and techniques might be needed to analyze and explain the data (Chatfield, 2003; Wei, 2013). The summary statistics first performed, such as the measure of the mean and standard deviation, can be significantly deceptive if there are systematic components on the time series in question (Chatfield, 2003). Some of the systematic components mentioned above are trends and seasonal variation.

In more detail, seasonality occurs when the data displays cyclic variation in a certain period of time, that is, when distinct patterns repeat within any fixed period, such as a season or a year (Metcalfe and Cowpertwait, 2009; Chatfield, 2003). However, take into account that some oscillations/cyclic variations on the data from the time series might not only happen due to seasonality, but also due to some other physical explanations, such as temperature, which compared to the seasonal variation, these cases can also be predicted to some degree (Chatfield, 2003). Regarding the trend, it is often defined as changes in the mean level of the time series in any consistent direction (Chatfield, 2003; Chandler and Scott, 2011; Kendall and Ord, 1990). Consider that, if in a specific case the trend was not found, it may not mean that there is no trend, just that the data was not plentiful to clarify a trend that actually occurs (Chandler and Scott, 2011).

Moreover, while analysing the data after discarding the cyclic variations and trends, it may still be possible to identify if some other cyclic variations were not observed before, or if some seemingly irregular variations might be explained. For that to happen, it is crucial to use probability models, similarly to *Autoregressive (AR)* or *Moving-Average (MA)* models (Chatfield, 2003).

Besides the time series plot, other graphical techniques might fetch more essential details for exploratory analysis, such as observation of outliers that might been lost in long series. One of these alternative graphical techniques is the boxplot (Figure 2) (Chandler and Scott, 2011). The boxplots consist in comparing the distributions of the data observed in distinct groups, for instance, the comparison of the distribution of data in each particular month of the year (Chandler and Scott, 2011; Tukey, 1977).

In each group, there is a rectangle where its top is the third quartile of the data, and its bottom is the first quartile of the data (Chandler and Scott, 2011; Benjamini, 1988). As specified in Chandler and Scott (2011), the height between the top and the bottom of the rectangle is denominated as the *Interquartile Range (IQR)*. Furthermore, in each rectangle there is a horizontal line that represents the median, which gives, in just a glance at the boxplot, a potential location parameter for the data distribution and, consequently, more knowledge about the location of the data (Chandler and Scott, 2011; Benjamini, 1988). Also, there are two vertical lines - the 'whiskers' - that prolong from the top and from the bottom of the rectangle to the largest and smallest observation, respectively, those observations being within 1.5 times the IQR (Chandler and Scott, 2011; Benjamini, 1988). If there are other observations farther from these vertical lines, these are represented individually and may be

possible outliers (Benjamini, 1988; Chandler and Scott, 2011). As mentioned in Chandler and Scott (2011), the concept of the boxplot's whiskers is to represent each data distribution's main body.

Since the data's information about location, skewness, longtailedness and spread can be observed by just taking a glimpse at the boxplots, it is possible to use this approach as a simple alternative to visually compare distributions of many groups of data (Chandler and Scott, 2011; Benjamini, 1988). A way to compare entire distributions with this type of graph is to present each distribution's boxplot side by side, as shown in Figure 2 (Benjamini, 1988).

Hence, boxplots can be of great benefit to the time series plots, helping with the exploratory analysis (Chandler and Scott, 2011). This type of graphical method can, as already mentioned above, determine changes in the distributions' variability and shape, and also identify probable outliers (Chandler and Scott, 2011). Nevertheless, boxplots also have some inefficiency when it comes to interpreting data that has trends or has many lacking observations (Chandler and Scott, 2011).

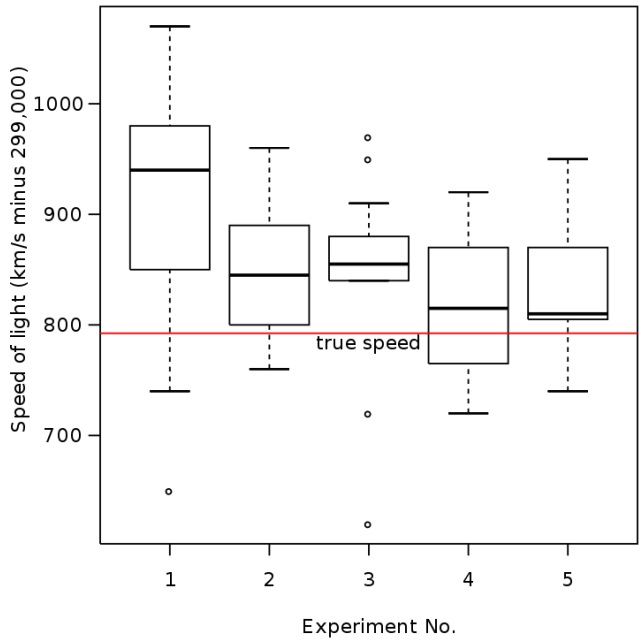


Figure 2: Example of a boxplot with Michelson's data on the speed of light. The x-axis represents each of the five experiments that were performed from which the data was collected, and the y-axis represents the values for the speed of light. This boxplot is from the Michelson–Morley experiment (Kempson, 1988).

2.3.1 *Data Preprocessing Methods*

Nowadays, people are surrounded by immense raw data and this enormous data growth makes their organization and understanding arduous (García et al., 2016). Because of the rapid growth of the amount of data collected, beyond the addition of time dimensions in these vast databases and the appearance of more high-dimensions data sets, it has become increasingly difficult to find solutions to the various problems encountered (Wei, 2013; Brockwell et al., 2002). Ergo, the data most certainly will consist of redundancies and inconsistencies, not suitable to a data mining process - process of data analysis with the goal to solve the problems in question (García et al., 2016, 2015). To adequately analyse long time series it is necessary for the data to undergo a series of preprocessing stages, so that useful datasets can be attained for additional data mining methods (García et al., 2016, 2015).

The major tasks of data preprocessing are data preparation and data reduction (García et al., 2016). The data preparation consists in data cleaning, data transformation, data integration and data normalization, making this step crucial since it modifies raw data into suitable input to data mining (Figure 3) (García et al., 2016, 2015). Whilst, the data reduction includes *Feature Selection (FS)* and *Instance Selection (IS)*, which help to decrease the data complexity (Figure 4) (García et al., 2016, 2015). All of these preprocessing methods are capable of adapting the data and making sure it meets the requirements of the data mining process, turning data that was once unfeasible to data that can now be processed, and optimizing the quality of the future applied models, since the complexity of the data is reduced (García et al., 2016, 2015).

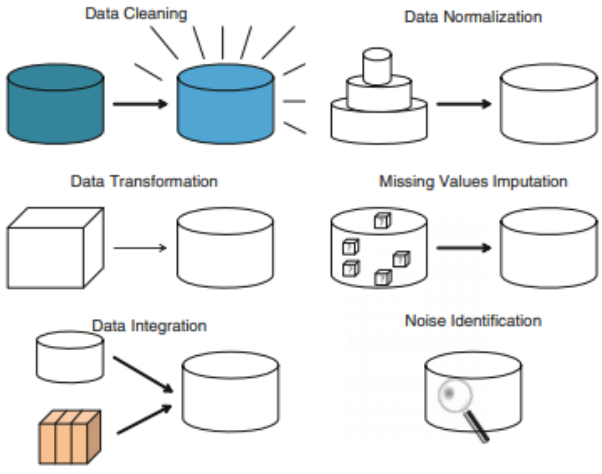


Figure 3: Representation of the various methods included in data preparation. The data preparation includes methods that turn the raw data into fitting data, applying steps such as data cleaning, data normalization, data integration, data transformation, among others. Retrieved from García et al. (2015).

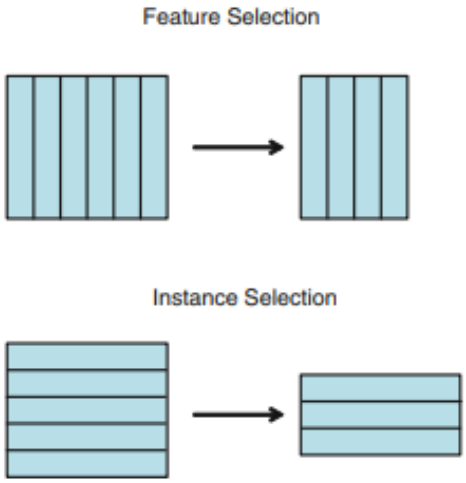


Figure 4: Representation of the various methods included in data reduction. The data reduction applies methods like feature selection and instance selection, so that data complexity can decrease. Adapted from García et al. (2015)

Firstly, the data cleaning technique comprises a series of procedures that adjust the original data, filtering out from the data set some inaccurate data and downsizing irrelevant details (García et al., 2015). Also, in this technique there are tasks to recognize discrepancies and data segments that do not correspond to the rest of the original data (García et al., 2015). Furthermore, techniques such as missing data imputation and noise identification are a part of the data cleaning process (García et al., 2016, 2015).

Regarding the problematic of the missing values, there are a few options to resolve this situation: discarding the instances that might include missing values (García et al., 2016). Nevertheless, take into account that this process might result in a biased learning process, once the elimination of important information might be a possibility (García et al., 2016, 2015; RJa and Rubin, 1987); filling the missing values with approximate ones, provided by models with probability functions, sampled by maximum likelihood procedures (García et al., 2016).

On the other hand, the noise is commonly treated by data polishing methods, that rectify them, or by noise filters, that identify and eliminate the noisy occurrences that exist in the data (García et al., 2016, 2015; Zhu and Wu, 2004).

Next off, there is the data transformation, which is a data preprocessing family, where a number of techniques are included (García et al., 2015). Some of those techniques are data's summarization, normalization, generalization, smoothing and feature construction (García et al., 2015). Due to the data transformation, the data mining outcome might be more effective, in consequence of the conversion or consolidation that the data suffers in this process (García et al., 2016, 2015).

Regarding the data integration process, this technique consists in the combination between data from numerous data supplies (García et al., 2015). Some frequent operations that are performed in this process are the variables' identification and its consolidation, the recognition of disagreements that might happen in values from distinct data sources, and the analysis of complementary relationships between the data attributes (García et al., 2015). Consider that this technique might result in inconsistencies and redundancies in the data set if not executed cautiously (García et al., 2015, 2016). With this in mind, Elmagarmid et al. (2006) refers to duplication as one of the most usual causes for these inconsistencies.

Another crucial technique is data normalization. Since for the data analysis it is necessary to handle data with equivalent measurement units and a similar range or scale, normalizing the data is fundamental (García et al., 2015). As mentioned in García et al. (2015), the data normalization is specifically advantageous while using methods of statistical learning.

Some common types of transformations the distributions of the original data go through are **min-max normalization**, **z-score normalization** and **decimal scaling normalization** (García et al., 2015). The former has the intention of scaling all the data in a distinct range as its purpose (García et al., 2015). When the method of the min-max normalization cannot be used, it is time to apply the z-score normalization, which converts the data so the mean and the standard deviation become, respectively, zero and one (García et al., 2015).

Ultimately, due to the raw data not being free of missing values or/and outliers, there is a necessity for the assessment of the data quality and its probable transformation, as already explained above (Chatfield, 2003).

After evaluating the data's arrangement, it is essential to implement various techniques, like the ones mentioned above, that possess important roles to clean the data, such as adjusting outliers, correcting missing values and recognizing as well as fixing errors that are apparently displayed in the data (Chatfield, 2003). Nonetheless, sometimes it is not enough to use simple methods to carry out this data cleaning process, and more complex methods might be required (Chatfield, 2003). As a result of this, it is fundamental a profound knowledge of time series models (Chatfield, 2003).

The next stage for this analysis is to verify if trends or seasonal variations are exhibited in the data and how to proceed from that - how to model, measure or remove it (Chatfield, 2003). It is necessary to develop new statistic methods taking into account these high-dimensional variables, data sets, and highly correlated data, such as the one presented in time series. This necessity is heightened, considering that most of the statistical methods created so far are too traditional and cannot be used in these types of data. (Wei, 2013).

2.3.2 *Existing Probability Models for Time Series*

In general, as already defined in the section above, time series are a result of stochastic processes. Hence, the time series' probability models are also labeled as 'stochastic processes', being under its properties (Chatfield, 2003). A simple approach of characterizing a stochastic process is to provide the mean function, the autocovariance function and the variance function as "moments" of this process (Chatfield, 2003; Karlin and Taylor, 1975).

Among the existent stochastic processes, some can be identified as stationary (Chatfield, 2003; Karlin and Taylor, 1975). This definition occurs when, in a stochastic process, the mean and the variance are constant independently of the time, and when the covariance between two periods of time depends entirely on the lag - the time difference - among the two time periods, not on the exact time value (Chatfield, 2003; Karlin and Taylor, 1975).

As reported by Wei (2013), the *time series analysis* can be divided into the time domain approach and the frequency domain approach. The first domain approach mentioned resorts to time functions in order to describe the characteristics of the process and delineate its evolution (Wei, 2013). Some of those time functions include the *Autocorrelation Function (ACF)* and the *Partial Autocorrelation Function (PACF)*.

The ACF is the autocovariance function standardized, simplifying the interpretation and description of a time series (Chatfield, 2003; Wei, 2013). This autocorrelation helps to find similarities among observations that are apart by a time lag, its analysis aiming to the discovery of repeating motifs, like finding signals that happen systematically and which are not clear due to the noise that is present (Chatfield, 2003; Wei, 2013; Chandler and Scott, 2011).

One of the tools, reported by Kozma et al. (1994), used in time-domain studies is the *Sequential Probability Ratio Test (SPRT)*, which is a sequential hypothesis test that differentiates between two hypotheses and in this context can be used as a method to encounter particular occurrences of different behavior in the past (Wald, 1945; Spiegelhalter et al., 2003).

On the other hand, the frequency-domain approach uses a spectral function to study how the variation of this process can be explained by the combination of sines and cosines at various frequencies (Wei, 2013). An advantage of this domain analysis is that it acquires additional information on the anomaly's spectral distribution, that then can be used to characterize an identified anomaly (Kozma et al., 1994). Nevertheless, this domain has a distinct disadvantage in comparison with the time-domain, since it needs to calculate an average of the time series' *Auto Power Spectral Densities (APSDs)* over specified time intervals to enhance the statistical accuracy of these studies. This necessity makes it a lot slower to respond, while the time-domain methods perform sample-by-sample evaluations, which are faster (Kozma et al., 1994).

In the analysis of time series there are various stochastic processes that might be suitable when creating a time series model. Some of the processes mentioned in Chatfield (2003) and in Box et al. (2015), are:

- MA processes;
- AR processes;
- *Autoregressive Moving-Average (ARMA)* models;
- *Autoregressive Integrated Moving-Average (ARIMA)* models.

All the MA models are stationary (Chatfield, 2003; Chandler and Scott, 2011). These type of processes, as well as the AR processes, are different kinds of processes that adopt the ACF (Chatfield, 2003; Wei, 2013; Chandler and Scott, 2011).

The MA model establishes that the result is determined by the linear combination of various random variables of the time series (Wei, 2013; Chatfield, 2003). On the other hand, the AR model portrays a randomly generated process (Wei, 2013). This process has the intention of regressing to its own prior values, since its expression is a linear collection of the process's former values, alongside with a random shock (Chatfield, 2003; Wei, 2013; Box et al., 2015). Opposed to the MA model, the AR may not be stationary (Chatfield, 2003).

The combination between MA models and AR models leads to models with a more complex stochastic structure, the ARMA models and the ARIMA models, introduced by Box et al. (1970) (Chatfield, 2003; Wei, 2013; Whittle et al., 1963; Pandit et al., 1983). Both are suitable to interpret actual time series or to forecast its eventual values (Chatfield, 2003; Whittle et al., 1963; Pandit et al., 1983). However, take into account that the ARMA models are only suitable to describe stationary time series (Box et al., 2015; Wei, 2013). While, the ARIMA models arise from ARMA models, being more appropriate to analyse non-stationary time series, as well as, stationary ones (Box et al., 2015; Wei, 2013). There is also the *Seasonal Autoregressive Integrated Moving-Average (SARIMA)* process that might be achieved in a time series that displays a seasonal variation by an adjusted ARIMA model with seasonal parameters (**?**).

According to Agnieszka and Magdalena, in 2018, one of the most applied analytical tools, based on the analysis of the obtained data, was the different existing models, mentioned above. These models help in the visualization, clarification, and prediction of the gathered data. However, this gathering process is indeed impactful, since the model's estimation and its results can be significantly influenced simply by an incorrect entry data.

2.3.3 *Existing Algorithms for Determination of Outliers*

The information of the existence of outliers is crucial for the gathering and the exploratory analysis of the data. An outlier is an observation from a stochastic process that is far off from the great mass of the remaining observations (Raña et al., 2015), which, eventually, can be recognized as a legitimate data point or as noise (Agnieszka and Magdalena, 2018).

An important task in system's monitoring and diagnostic is the detection of these unusual observations. The discovery of these outliers in the time series data might assist the classical statistical techniques, since they can seriously influence the results of these analysis (Raña et al., 2015). Removing their consequences from the observations will help to better comprehend the fundamental structure of the time series (Chang, 1982; Chang et al., 1988).

In time series, since the outliers can appear step-by-step, they might remain unseen at the initial stages of their development, being only recognized later, depending on the method sensitivity (Kozma et al., 1994). Constant monitoring to encounter discrepancies of the systems' usual pattern can be performed using control graphics, expert systems, pattern recognition, models based methods, cluster analysis and neuronal networks (Jones et al., 2014; Dasgupta and Forrest, 1996).

Since the outliers detection is a valuable research problem, researchers are trying to find universal approaches for this problem (Agnieszka and Magdalena, 2018). However, each type of data has their own features, which makes this objective hard to achieve. As reported in Raña et al. (2015), the statistical literature suggested two approaches to handle this problem: developing robust methods, which are designed to be insensitive to outliers; and detecting the outliers. While using the approach of detecting outliers, these should only be removed from the initial sample if they derive from an error, before further analysis of the data. However, if an outlier does not originate from an error, robust methods should be applied to estimate parameters that are associated to the process (Raña et al., 2015; Chatfield, 2003).

Regarding to the detection of outliers, there are some graphical methods that can be used. According to Raña et al. (2015), the boxplot (Figure 2) (Tukey, 1977) is the most common graphical method for the discovery of outliers.

As mentioned in the section 2.3, the observations that happen farther then 1.5 times the IQR are considered outliers (Chandler and Scott, 2011; Benjamini, 1988; Sun and Genton, 2011). Consequently, the outliers can be detected using the following equations retrieved from the knowledge of the boxplots,

$$LW = Q1 - 1.5 \times IQR \tag{1}$$

or as,

$$UW = Q3 + 1.5 \times IQR \tag{2}$$

where the Q1 and Q3 are, respectively, the first and third quartiles of the data, and the IQR is given by $IQR = Q3 - Q1$, as mentioned above. In equation 1, the result defines the lower whisker (LW) of the boxplot, whether in equation 2, the result defines the upper whisker (UW) (Chandler and Scott, 2011). The observations that are higher or lower than, respectively, the value of UW and LW are considered outliers of the data.

Also, another approach to detect outliers in time series is cluster analysis. Since these kind of analysis organizes the data according to the main patterns encountered on the data set, the observations that are significantly differentiated from these patterns will be interpreted as outliers (García et al., 2015).

Besides these two approaches, there are also some models that are used to encounter unusual observations on the time series. First, in Box et al. (2015), it is mentioned an additive and innovational outliers' model. In the first publication about outliers in time series, conducted by Fox (1972), he described two types of outliers, *Additive Outlier (AO)* and *Innovational Outlier (IO)*. The first type of outlier only influences a particular observation at a distinct time value. The second type corresponds to cases where a unique "innovation" influences, not only that specific observation in question, but also all the succeeding measurements (Box et al., 2015; Chatfield, 2003; Fox, 1972). The impact of these IO is more permanent in non-stationary time series, once the stationary time series's effect diminishes after a relatively short period of time (Box et al., 2015).

Due to the possibility of several AO and IO being detected in a single time series in various time values, multiple outlier models might be thought-out (Box et al., 2015). However, an issue related to these models is the possible lack of information about the time values where the outliers occur and which type of outliers are present (Box et al., 2015).

In Dasgupta and Forrest (1996), another anomaly detection model was mentioned, the *Multi-Layer Perceptrons (MLP)*. This model is capable to answer problems with a stochastic approach, calculating almost accurate results for these complex problems, using supervised learning (Rosenblatt, 1961; Rumelhart et al., 1985; Dasgupta and Forrest, 1996).

Another outlier detection approach is the iterative procedure, suggested by Chang et al. (1988) and Chen and Liu (1993). First off, the ARIMA model is performed for the time series in question, as if there were no outliers. Take into account that, in a time series each time value has one residual - the difference between the predicted value of a time series' data point and the actual value noticed in the time series data. The IO or AO's information is enclosed in a certain residual with a distinct time value or scattered through various residuals throughout the time points, respectively (Box et al., 2015). Therefore, when estimating the ARIMA model, residuals can be attained and possible outliers can be detected. Subsequently, after removing the effect of the detected outliers from these residuals, it is possible to calculate the ARIMA model again, this time using the new modified residuals. If

more outliers are detected, these procedures can be repeated in the expectation that at some point there are no outliers left to identify (Box et al., 2015).

With this iterative procedure, at the end it is possible to detect all the outliers and remove them, maybe creating an altered time series where the outliers' impact is not presented (Box et al., 2015).

Nevertheless, removing outliers and their impact from the time series can be deceptive, if in reality the outliers are not an error, but an authentic value of the data (Chatfield, 2003). On the other hand, if an outlier that is originated by an error remains on the data, it can miscalculate the time series analysis. Hence, it is crucial to make a thorough decision, based on the situation's context (Chatfield, 2003; Box et al., 2015).

For that reason, it is beneficial to find the outliers' origin, in order to better comprehend their effects on the time series and to prevent feeble decisions. However, when the time series has various outliers it becomes more troublesome to discover the cause behind these unusual observations (Box et al., 2015).

An alternative approach for this issue is the use of robust models, as a substitute for removing outliers (Chatfield, 2003; Box et al., 2015). As mentioned above, when using robust models, even though there are most likely various outliers in the time series, these models are not influenced by them. Hence, it is possible to analyse the time series data without worrying about misleading values and effects that might be present (Chatfield, 2003; Box et al., 2015). Some of these robust methods include running median smoothers from Hoaglin and Velleman (1981) and *Locally Weighted Regression (LOWESS)* from Cleveland (1993), both aiming at smoothing a time series and obtaining smooth estimations (Chatfield, 2003).

Another existing robust model is the *Adaptive Resonance Theory (ART)*. This model focus on problems like pattern recognition to detect outliers, and uses supervised and unsupervised learning methods, being a neural network model (Caudell and Newman, 1993; Carpenter and Grossberg, 2016; Grossberg, 2013). Besides, it does not require prior work, to be calculated. Bear in mind that a powerful model should detect significant changes that were not yet seen, such as the example of the ART model, rather than searching for particular unusual activity patterns already seen (Dasgupta and Forrest, 1996).

2.3.4 *Time Series Forecasting*

With the data mining of a time series, analysts aim to extract some meaningful information from the time series data. A very interesting information retrieved from this, is the prediction of future observations within a time series. However this forecasting task is yet a complex problem for most of the existing computers (Esling and Agon, 2012).

The *time series forecasting* task is advantageous for many of real-life problems, being adopted in various types of planning and decision making processes (Esling and Agon, 2012;

Montgomery et al., 2008). Areas, including economics, marketing, production planning, finance and sales forecasting, risk management, stock control and medical surveillance, can improve as a result of these forecasting processes, which attempt to find a valuable answers for the related significant problems (Box et al., 2015; Chatfield, 2003; Montgomery et al., 2008; Esling and Agon, 2012). Also, most of the problems related to control are associated with forecasting, since it is possible to fix a situation that is leading to an unusual observation in the data, if it is possible to predict that same situation (Chatfield, 2003).

Besides these comprehensible applications, the forecasting task can also evaluate the appropriateness of the time series models, since it is possible to use fitted time series models to forecast (Chatfield, 2000).

Nevertheless, bear in mind that not all the time series data has the same features. Because of this reason, the performance of the models utilized to predict the subsequent values of the data in question, might be different depending on the time series analysed (Agnieszka and Magdalena, 2018).

As mentioned in Chatfield (2003), there may even exist time series that are surprisingly constant throughout the time and thus not really predictable since the subsequent observations may not vary from the prior observations in the future. However, other time series vary throughout the time. For that reason, plotting a time series is very beneficial to understand what kind of data is being analysed (Chatfield, 2003).

In general, the forecasting models predict values within a period of time, this period is denominated as lead time and varies depending on the problem the forecasting models is being applied to (Box et al., 2015; Chatfield, 2003). Take special attention to the fact that the lead time and the time where the forecasting model is calculated are different time points (Chatfield, 2003).

These forecasting models exist in a broad range of variation, where there is no distinct model that might be applied to all the forecasting problems. Therefore, there is a necessity to choose among these different forecasting models, the one that best suits the conditions in question (Chatfield, 2003).

According to Chatfield (2003) there are three types of forecasting methods: subjective, univariate and multivariate. The subjective methods use information obtained from subjective circumstances, such as intuitions and judgements, to predict the future (Chatfield, 2003, 2000). Even though the subjective approach is used in most of the statistical approaches, such as choosing the suitable model, most of these models are not really used alone, because there is a higher interest in objective forecasts (Chatfield, 2003).

Regarding to the univariate forecasting methods, these predict the future with present and past observations as the only support, since only one variable varies throughout the time. Whereas the multivariate methods are more complex and predict the following values

of a variable, taking into account other observations from other time series variables, that influences the time series data in question (Chatfield, 2003, 2000).

With these 3 methods in mind, consider that most of the forecasting models are a mixture of these methods (Chatfield, 2000). Furthermore, in some situations, there are the need to use either automatic methods or non-automatic methods. In the automatic methods humans do not intervene, while in the non-automatic methods its require their intervention (Chatfield, 2003, 2000). These type of distinction is necessary, since in some cases there is the demand to have a careful and thorough analysis, where the non-automatic methods are more appropriate, and in some cases there is the need to have a more broad and simpler analysis, using automatic methods (Chatfield, 2000).

In general, the subjective methods are executed with non-automatic methods, as well as, much of the multivariate methods, and the univariate methods can be adapted with each of the two methods (Chatfield, 2003).

Since a high number of forecasting methods uses time series models, some of the models mentioned on the subsection 2.3.2 are used as forecasting models. Examples of univariate forecasting models are the ARMA and ARIMA models, as well as, the SARIMA models (Chatfield, 2003, 2000). In addition, while using ARIMA and SARIMA models in the forecasting procedure there is the advantage that the forecast can be conducted in non-stationary time series and time series with seasonality (Chatfield, 2000; Box et al., 2015). The general procedure to forecast with ARIMA models consists in devising a reasonable model, fitting it to the time series data, making sure its suitable, and if not, adjusting the model to become suitable for this forecast problem (Chatfield, 2000).

Regarding the multivariate forecasting methods, these are applied in multivariate data, where it is necessary to model multiple variables that differ throughout the time. In this case, some examples of multivariate forecasting models are multiple regression, and AR and ARMA models that are adapted to multivariate data, such as *Vector Autoregressive (VAR)* and *Vector Autoregressive Moving-Average (VARMA)* models (Chatfield, 2003, 2000).

Comparing to the MA approaches, the AR approaches are much more easier to utilize in a multivariate time series data. As reported by Box et al. (1976), AR models were applied during a long period of time in the prediction of values. Take into account, that an AR model predicts the values of the next observation calculating the correlation between the measurements of the prior observations (Mills and Mills, 1991; Pandit et al., 1983; Chatfield, 2000). Nevertheless, regarding to the VAR models, these do not take into consideration the seasonal variation and the trend that might be present in the time series data in question (Chatfield, 2003, 2000).

In the end, it is necessary to fully comprehend the data, so that the selection of the forecasting models is the most appropriate for the problems in hand.

3

SISBI AND BIOECONOMICS PLATFORM

3.1 SISBI

Industrial Biotechnology presents numerous challenges. Some of these challenges can be solved resorting to the field of Metabolic Engineering since it enables the production of compounds with great importance in the economy of the world.

SISBI, in cooperation with the Center of Biological Engineering from UM, is a project managed by SilicoLife with the purpose of developing an intelligent decision support system that can be used in the Industrial Biotechnology field. This system supports the improvement of innovative tools that can extract and integrate knowledge, supporting the decision-making in the context of Industrial Biotechnology, such as selecting suitable host microorganisms, compounds with additional value or favorable raw materials to utilize in target bioprocesses.

3.2 BIOECONOMICS PLATFORM

In the SISBI context, a platform that provides continuous monitoring of metabolites and raw materials prices was created - the **bioeconomics platform**. The back-end of this platform was built in *Java* over the *Spring Boot* framework, which supports and facilitates the creation of *Java* applications (Walls, 2016; Webb et al.; Gutierrez, 2014). Moreover, the *Java Spring Boot* framework is supported by *MongoDB*, a document-oriented database (Pivotal Software; MongoDB).

In fact, to adapt the platform to manage web requests and responses in *Java*, Spring Data *Representational State Transfer (REST)* was employed, being crucial when building systems that have a web extension (Webb et al.). This platform also utilizes Spring Data *Java Persistence API (JPA)* [1] with the purpose that MySQL Database connectivity can be integrated in Spring Boot. The Spring Data JPA helps to store data in a relational database using JPA repositories, which is necessary when conceiving a database (Walls, 2016).

1 *Application Programming Interface (API)*

In order to reach its goal, the platform includes various processes that are illustrated in the figure below (Figure 5).

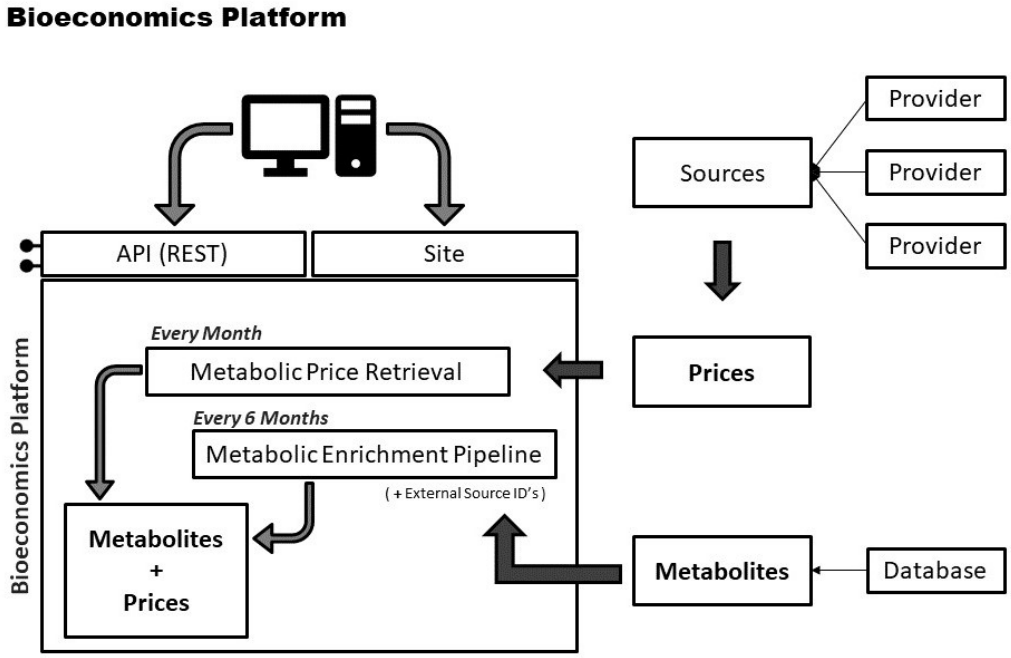


Figure 5: Diagram of the Bioeconomics Platform and its general process. As shown in this figure, after collecting the metabolites from the existent databases, a process designated as *Metabolic Enrichment Pipeline* occurs in the bioeconomics platform. Then the enriched metabolites go through another process named *Metabolic Price Retrieval* which gathers the prices related to the metabolite in question from distinct sources. Following both these processes, the metabolites and the prices become associated and are maintained in a database.

To truly understand this platform, the first fundamental step is to comprehend the related concepts and data objects. At the beginning, various metabolites are gathered so they are included in the bioeconomics platform (Figure 5). Hence, so that these metabolites can be managed in the platform, they are transformed into *Java* objects, since this is an object-oriented programming language.

The object representation of one of these metabolites is named as **Bioeconomics Metabolite**, where its description is a set of external database identifiers, conjugated with a preferred name and a list of synonyms (Table 1). Also, other attributes included in this object are an InChI, an InChIKey, a hashed counterpart of the full InChI (Heller et al., 2015), and a SMILES description, all with the purpose to better identify the respective metabolite (Table 1).

IUPAC developed the InChI so it would function as a text label that designates a chemical substance. For this reason, the same InChI refers to a unique compound, and consequently, two different substances must have different InChI's. To create a shorter and more compact

version of this identifier, the InChIKey was generated, which is a compact fixed-length digital format of an InChI string. However, this format loses the uniqueness, since two distinct molecules may have the same InChIKey (Heller et al., 2015). About SMILES, this is a chemical notation language specifically defined for computer usage, that depicts the molecular structure of a compound into a series of characters (Weininger, 1988).

Regarding the external sources ID's, these are the identifications correspondent to other sources, namely databases besides the bioeconomics platform, such as *PubChem*, *MolPort*, among others (Table 1). With these identifications, the platform has more information to search for different prices in various databases. Consequently, these external sources ID's are considered the main properties to attain a considerable number of different prices available for the same metabolite. Therefore, the external database linkage becomes one of the most relevant information in this work, once it promotes the discovery of the corresponding prices for that bioeconomics metabolite in a broad set of search modules.

Take into account that this work introduces the CAS number as an external source ID.

Table 1: Representation of a bioeconomics metabolite. In this table, each of the properties are displayed with a given explanation and example. These six properties put together define a metabolite in this study.

Properties	**Explanation**	**Example**
Prefer Name	Most common name.	Carvone
InChI	Text label that designates a chemical substance (Heller et al., 2015).	InChI=1S/C10H14O/c1-7(2)9-5-4-8(3)10(11)6-9/h4,9H,1,5-6H2,2-3H3
InChIKey	A compact fixed-length digital format of an InChI string (Heller et al., 2015).	ULDHMXUKGWMISQ-UHFFFAOYSA-N
SMILES	A chemical notification language specifically defined for computer usage (Weininger, 1988).	CC1=CCC(CC1=O)C(=C)C
External Sources ID's	Identifications from external databases, such as *MolPort*, *PubChem*, among others.	MOLPORT: [MolPort-002-506-968, MolPort-001-769-671] OXCHEM: [AX8120147, AX8048317]
Other Names	List of the name synonyms for that particular metabolite.	2-Methyl-5-(1-methylethenyl)-2-cyclohexen-1-one; Oil of curled mint; Dill weed oil terpeneless

As clarified in figure 5, these metabolites from the bioeconomics platform will then take part in the *Metabolic Enrichment Pipeline*.

On the other hand, the object representation of a price in this platform is a **Bioeconomics Metabolite Price**. This object includes various attributes, which are the price value, the amount, the currency, the unit, the provider, the source, and the date. Therefore, for a better understanding, table 2 presents all the properties of this object with some examples.

Table 2: Representation of a bioeconomics metabolite's price. In this table each property of a price is portrayed with a given explanation and example. These seven properties put together define a metabolite's price in this study.

Properties	Explanation	Example
Price	Price value that the metabolite has.	28.0
Amount	Quantity of the metabolite that is for sale.	2.0
Currency	Currency of the price value.	USD, EUR
Unit	Unit of the amount for sale.	mg, g, kg
Provider	Company that sells the metabolite.	AK Scientific, Inc.; Vitas-M Laboratory, Ltd.
Source	The place from where the price was fetched.	MolPort, CHEMSPACE
Date	Date from when the price result for the corresponding metabolite was found.	23/11/2018

As presented in figure 5, the prices are collected by the *Metabolic Price Retrieval*, that will be further explained in the next section.

First, the platform gathers these bioeconomics prices from various sources, which they, in turn, had already collected from numerous providers. The following diagram was created to envision how this price accumulation takes place (Figure 6). Also, as depicted from this figure, consider that different sources can acquire prices from the same provider. Nonetheless, this does not mean that the prices are inevitably the same.

In addition, bear in mind that the source property, represented in figure 6, is not the same as the external source property in table 1. Both belong to different objects of the bioeconomics platform, respectively, the bioeconomics metabolite price and the bioeconomics metabolite. Alternatively to the explanation given above for what the external source ID represents, the source property is the place where the bioeconomics platform retrieves the prices from

(Figure 6). For instance, some of these sources are the online marketplaces mentioned in section 2.2.

Besides this, while observing figure 6, the difference between these sources and the providers is also evident. Following the explanation above, the source is where the bioeconomics platform retrieves the prices from, and the provider is the company that sells the metabolite (Table 2).

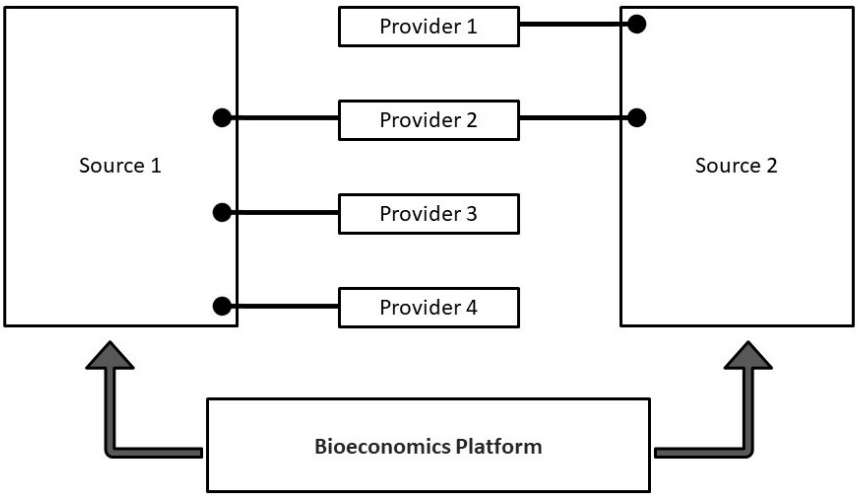


Figure 6: Representation of the collection of the different prices, from the providers to the bioeconomics platform. In this figure it is possible to see that one price from a provider is collected from a source, where from there the bioeconomics platform collects numerous prices.

Another detail to take into account in this step is the encounter of prices from various metabolites in each provider. In other words, each provider may comprise more than one price from one specific metabolite. This situation occurs because these prices are differentiated by the amount and unit of the metabolite in question. If perhaps they are identified with the same amount and unit, they can still be distinct to each other due to the different dates associated. Consequently, as sources bring together multiple prices from different providers, it might be possible to encounter different prices for the same metabolite, with the same amount and unit, and also at the same time, in the same source.

Furthermore, note that, frequently, different sources with prices from the same provider might identify the same provider with different names. These names might have slight changes between one another, such as commas and upper cases.

To sum up the representation in figure 6, the sources have the price information of the sale the provider is making, and the bioeconomics platform gathers the prices encountered in these sources by the Metabolic Price Retrieval.

Eventually, the bioeconomics platform will possess a list of prices associated with a specific metabolite, where each price will differ on the several characteristics that defined

it (Figure 7). One of these characteristics, the retrieval date, is vital since each price of a particular provider, source, amount, unit, and currency, contains different price values linked to different dates of the updates performed. That way, the evolution over time of that specific price can be examined.

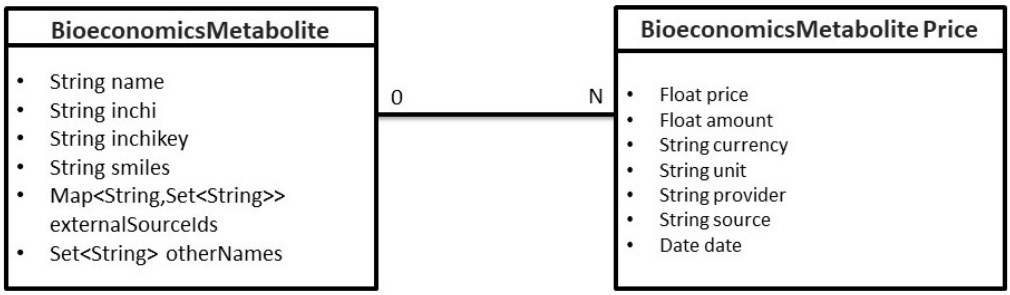


Figure 7: Representation of a bioeconomics metabolite and its prices in the bioeconomics platform. In this figure, the relation between a metabolite and a price is displayed, as well as the properties that define both objects and their data type. Notice that one metabolite is allowed to have different prices, whereas a price is always associated with only one metabolite.

Notice that, for simplification, every metabolite and every price further mentioned in this dissertation will have the definitions described above.

In short, the whole process of the bioeconomics platform initiates with a metabolite add/update request. At this point, the bioeconomics platform saves/updates the metabolite in question and attempts to enrich its information. This process is called the *Metabolic Enrichment Pipeline*, and it updates every six months since there is the possibility of a future change in the attributes of each metabolite.

After the enrichment of the metabolite, each metabolite is expected to possess a diverse number of external sources ID's. The higher the number of these identifiers, the higher the probability of discovering the available prices for this metabolite. Thus, the initial metabolite requested becomes updated so that it can proceed to the next stage, the *Metabolic Price Retrieval*.

In this stage, the platform picks up the prices for the respective metabolite. This process is performed monthly so that the prices can be updated, and consequently, the evolution of the price value throughout the time can be observed.

At last, the database of the bioeconomics platform is updated, so that every metabolite added contains different prices that can be consulted.

3.2.1 *Metabolic Enrichment Pipeline*

The main goal of the *Metabolic Enrichment Pipeline* is to retrieve additional information about the metabolite attributes, considering these are essential elements to search the prices accurately.

First, the metabolite requested goes through each of the enrichment modules so that it gets updated with the additional information found on each database (*PubChem*, *ChemSpider*, and *DrugBank*). For instance, the added metabolite may only have the InChI information. Therefore, when entering an enrichment module, for example, *PubChem*, it searches additional information that might be necessary for the price retrieval, employing the InChI. Afterward, the metabolite passes to a next enrichment module, for example, *ChemSpider*, applying the initial information and the additional information acquired in the previous step, where it undergoes another search for further information related to the requested metabolite.

In order to successfully enrich the data information through these modules, meeting some of the underlying properties is a necessary factor. In other words, to run a specific module is necessary to have at least one identifier from a list of needed properties, that might be different for each module. Table 3 presents the properties that are necessary for each module so that the enrichment can be concluded. To clarify, the enrichment in *PubChem* can only be performed using either the *PubChem* ID, the InChI, the InChIKey, or the SMILES related to the metabolite. Next, concerning the enrichment module *DrugBank*, besides the InChI, InChIKey, or the SMILES, two other properties are applicable in this module: the *Drugbank* identifier or some other external source identifiers. Finally, for the *ChemSpider*, only the *ChemSpider* ID and the InChI property are available to perform the enrichment.

Subsequently, the metabolite is enriched with hopefully useful information, so that it can move on to the next phase of the bioeconomics platform. As mentioned above, this enrichment process occurs every six months so that the information of each metabolite is always updated, increasing the chances of discovering more prices.

3.2.2 *Metabolic Price Retrieval*

Following the previous step, is time for the *Metabolic Price Retrieval*. Here, the enriched metabolite goes through a series of procedures, so that the several prices in different online marketplaces and databases are detected.

Table 3: Portrayal of the modules for the *Metabolic Enrichment Pipeline* and their necessary metabolite's properties. For each module, there is a set of properties that are essential for the discovery of the metabolite in question in other databases and its external ID's.

Module	Metabolite's property needed
PubChem	1) PubChem ID
	2) InChI
	3) InChIKey
	4) SMILES
DrugBank	1) DrugBank ID
	2) Other external ID's
	3) InChI
	4) InChIKey
	5) SMILES
ChemSpider	1) ChemSpider ID
	2) InChI

For now, this step utilizes the following sites to encounter prices:

- *DrugBank*, already described above, which is a database that is exclusive to bioinformatics and cheminformatics resources combining precise drug data with extensive information of the drug target (TMIC);
- *MolPort*, an advanced chemical marketplace that sources a considerable amount of compounds from various suppliers (MolPort);
- *ChemSpace* - the database of chemical building blocks that is the most extensive small molecules' database. Their primary purpose is to provide to their clients, high-quality drug discovery solutions (ChemSpace);
- *Oxchem*, a chemical supplier with the aim to reach the universities, pharmaceutical companies, and research institutions of the world (Oxchem).

Similarly to the enrichment pipeline, in the price retrieval, there are also fundamental metabolite properties so that the existent modules can be performed. Table 4 lists all these properties for each available source.

In the *ChemSpace*, the price retrieval only happens if there is an ID from this source, a CAS number, an InChI, an InChIKey, or a SMILES associated with the metabolite in question. Alternatively, in the *DrugBank*, the retrieval is performed by adopting an ID of the

metabolite from this database, other ID's from external sources, an InChI, an InChIKey, or a SMILES. Thirdly, for *MolPort* source, the retrieval happens only when the metabolite has in its information the *MolPort* identification. Lastly, for the *Oxchem*, the necessary information is solely the *Oxchem* identification or the CAS number of the metabolite (Table 4).

Table 4: Portrayal of the modules/sources for the *Metabolic Price Retrieval* and their necessary metabolite's properties. For each module, there is a set of properties that are essential for the discovery of the prices associated to the metabolite in question.

Module	**Metabolite's property needed**
ChemSpace	1) ChemSpace ID
	2) CAS number (as an External Source ID)
	3) InChI
	4) InChIKey
	5) SMILES
DrugBank	1) DrugBank ID
	2) Other external ID's
	3) InChI
	4) InChIKey
	5) SMILES
MolPort	1) MolPort ID
Oxchem	1) OxChem ID
	2) CAS number

After performing the price search by metabolite, each of them contains a price list, in which each price has a specific attribute, such as a different provider, currency, amount, among others. The definition of bioeconomics metabolite price is already displayed above, with all the different attributes that each price can have. Nevertheless, an example of this platform's output for the *Carvone* metabolite is presented below, where the list of prices for this metabolite is presented (Table 5).

As shown in table 5, the first column displays the price value. Nevertheless, when analysing a price, a currency needs to be related so the price can have meaning. Thus, the price currency is displayed in the third column. In this case, only one currency is observed within these listed prices.

Additionally, table 5 also displays the amount of metabolite related to each price in the second column. Moreover, this amount can be in different units (kg, g, mg, etc.), represented in the fourth column (Table 5). Both these two pieces of information, the amount plus the

Table 5: An example of the price list output from bioeconomics platform. In this figure, the first 20 prices withdrawn from the *Carvone* metabolite are presented.

Price	**Amount**	**Currency**	**Unit**	**Provider**	**Source**	**Date** (Day/Month/Year)
115.0	5.0	USD	g	UkrOrgSynthesis	MolPort	07/12/2018
10.0	250.0	USD	mg	Oxchem Corporation	MolPort	13/10/2017
157.68	10.0	USD	g	Fluorochem Limited	MolPort	23/10/2018
52.0	50.0	USD	g	Astatech Inc	CHEMSPACE	25/11/2017
97.0	100.0	USD	g	Accela ChemBio Inc.	CHEMSPACE	07/12/2018
383.0	1.0	USD	kg	Labseeker	MolPort	13/07/2018
115.0	5.0	USD	g	UkrOrgSynthesis	MolPort	13/07/2018
35.0	20.0	USD	mg	TargetMol	MolPort	13/10/2017
28.0	2.0	USD	mg	Vitas-M Laboratory, Ltd.	MolPort	23/11/2018
229.0	100.0	USD	g	Angene International Limited	CHEMSPACE	23/08/2018
14.6	5.0	USD	g	Fluorochem	MolPort	07/12/2018
10.0	250.0	USD	mg	OXCHEM CORPORATION	CHEMSPACE	13/05/2018
25.0	5.0	USD	g	AK Scientific, Inc.	MolPort	13/07/2018
171.0	10.0	USD	g	UkrOrgSynthesis	MolPort	23/11/2018
363.4	100.0	USD	g	Biosynth AG	CHEMSPACE	23/10/2018
157.0	250.0	USD	g	Astatech Inc	CHEMSPACE	25/11/2017
63.0	1.0	USD	g	UkrOrgSynthesis	MolPort	13/06/2018
157.68	10.0	USD	g	Fluorochem Limited	MolPort	13/05/2018
63.0	1.0	USD	g	UkrOrgSynthesis	MolPort	07/12/2018
115.4	500.0	USD	g	J&K SCIENTIFIC LTD.	MolPort	07/12/2018

unit, are vital and need to be taken into account when examining the price evolution. This reason stems from the fact that those properties deeply relate to how much of a tendency there is when buying the metabolite, contributing to better prices in some amounts and unit scales.

Finally, the last three columns display the provider, the source, and the date of the different prices, respectively (Table 5). As mentioned above, the provider is the company that sells the metabolite, and these costs are gathered in the sources, from where the bioeconomics platform retrieves its prices.

As acknowledged above, this price retrieval process occurs every month. Hence, there is a possibility to create an evolution of the price values throughout the time, keeping the information updated and developing a straightforward approach to forecast the future prices of the corresponding metabolites.

Overall, the price retrieval task has many challenges. Above were already mentioned some of the difficulties of discovering the chemical compound prices, including the unreliability

of several online platforms and databases available. Besides, another one of these challenges is the fact that this task is **time-consuming**. As a result, a considerable amount of time is necessary to disclose all of these prices. Furthermore, it may exist inaccuracies in the values of the amount or the units, that can make all the forecasting and data analysis useless. Also, the demise of a provider's existence or a change in its name can make the tracking of all the prices under the same provider much more difficult.

3.3 REST API OF THE BIOECONOMICS PLATFORM

The bioeconomics platform has two main approaches concerning the access by external users to the data contained. These users can connect with the platform in question by entering the associated website or by linking to the available REST points using different *Uniform Resource Locator (URL)*'s (Figure 5).

For this dissertation, the focus will be set on the REST approach. Since the bioeconomics platform is a web service with a REST style, the transaction of data between the client and the web server can be allowed employing an API. Due to this possibility, the use of a REST API can give a response to the client request (Masse, 2011). However, to interact with the REST API of the bioeconomics platform, first, the access to the existent REST API endpoints/keys is necessary (Masse, 2011; Walls, 2016; Webb et al.). Depicted by table 6 are some of the existing fundamental REST API endpoints, their function, and, if applicable, their path variables. In this table 6, the path variables of these URL's are distinguished by the curly brackets. Depending on the purpose of these variables, the user needs to assign them when executing the REST API endpoints, so that the expected information is retrieved. Some of these path variables are:

- Page index - all the pages are numbered, starting from zero;
- Page size - the number of objects per page (metabolites or prices);
- Order - order in which the information will appear ('ASC'/'DESC');
- Sort - the element that will be sorted in the data;
- Text Search - text written by the user that will be searched in the data;
- Some other attributes, such as the bioeconomics metabolite ID, price ID, InChI, *etc.*

With this in mind, take into account that the path variable of the order can only be "ASC" (ascending) or "DESC" (descending). Also, regardless of the option to choose "0" (zero) as page index, the path variable of the page size can never be "0" (zero), since it would not retrieve any information.

Moreover, each WEB URL has a distinct function that aids in the goal of the user (Table 6). Nevertheless, despite how the information will be managed later, take into account the output nature that each of these REST API points displays after being accessed.

As observed in figure 8, using as an example a REST API point, its output is presented in *JavaScript Object Notation (JSON)*, which is currently the norm for data delivery. Overall, the outputs for all the REST API points of this platform are displayed in JSON, which as a result, leads to the necessity of understanding how to transform these data so it can be manageable (further information in the next chapter).

To clarify this example, the URL used helps to exchange the information about the metabolite in the chosen page, between the user and the bioeconomics platform, including all the available properties of this object (Table 1) (Figure 8). As shown in this figure (8), all the information is contained inside an array. If a specific REST API point includes the data of more than one metabolite, the information related to each metabolite will be inside curly brackets, and all the metabolites and their information will be in an array. In this case, there is only one metabolite, and therefore this is represented with only one block of data, inside curly brackets, in the whole array.

In order to choose this specific page, the path variables, page index, page size, order and sort by, were assigned respectively, with 1, 1, descending (DESC), and ID (Table 6).


```
[{"id":9222710493671371933,"entry":"PUBCHEM:21881641","name":"Telithromycin","descr
iption":null,"inchi":"InChI=1S/C43H65N5O10/c1-12-33-43(8)37(48(41(53)58-43)19-14-
13-18-47-23-31(45-24-47)30-16-15-17-44-22-30)27(4)34(49)25(2)21-42(7,54-
11)38(28(5)35(50)29(6)39(52)56-33)57-40-36(51)32(46(9)10)20-26(3)55-40/h15-17,22-
29,32-33,36-38,40,51H,12-14,18-21H2,1-11H3","inchiKey":"LJVAJPDWBABPEJ-UHFFFAOYSA-
N","smiles":"CCC1C2(C(C(C(=O)C(CC(C(C(C(=O)C(C(=O)O1)C)C)OC3C(C(CC(O3)C)N(C)C)O)
(C)OC)C)C)N(C(=O)O2)CCCCN4C=C(N=C4)C5=CN=CC=C5)C","otherNames":["I06-2338","191114-
48-4","AKOS015896612","Telithromycin"],"mapSourceExternalIdSet":{"PUBCHEM":
["21881641"],"CHEMSPIDER":["10628581"],"AKOS":["AKOS015896612"],"SHANGHAI IS
CHEMICAL TECHNOLOGY":["I06-2338"],"ABI CHEMICALS":["AC2A05E1H"],"JALOR-CHEM":["I06-
2338"],"CHEMBL":["CHEMBL3184043"],"CAS":["191114-48-
4"]},"createDate":1507721169485,"updateDate":1559778692379,"lastPriceSearchDate":15
71031528673,"entranceSource":null}]
```


Figure 8: Output from accessing the REST API point https://mendel.bio.di.uminho.pt/bioeconomics/rest/metabolites/getMetabolitesPaginated/1/1/DESC/id. As shown in this figure, the data from this endpoint is displayed in JSON, where the information of one metabolite is preserved inside the curly brackets that are inside the square brackets.

Table 6: REST API Points. Display of some of the fundamental REST API endpoints that exist in the bioeconomics platform, as their description and present path variables. In this table, to make it more understandable, the WEB URL's initial part was not included since it is the same in all of them (https://mendel.bio.di.uminho.pt/bioeconomics).

WEB URL	Function	Path Variables
(...)/rest/metabolites/getAllMetabolites	Returns all the metabolites from bieoconomics platform.	N/A
(...)/rest/metabolites/getMetabolitesPaginated/{paginationIndex}/{pagesize}/{asc}/{sortBy}	Returns the selected page content about the metabolites.	Page index, number of metabolites per page, order (ascending and descending), sort method
(...)/rest/metabolites/getMetabolitesPaginatedSearchable/{paginationIndex}/{pagesize}/{asc}/{sortBy}/{textsearch}	Returns the selected page content about the metabolites while matching the text given.	Page index, number of metabolites per page, order (ascending and descending), sort method, text to search
(...)/rest/metabolites/getMetabolitesCount	Returns the number of metabolite entities available.	N/A
(...)/rest/price/search/byid/{id}	Returns the price searching by its ID.	ID of the price
(...)/rest/price/search/bymetaboliteid/{id}	Returns the prices related to the ID of the metabolite.	ID of the metabolite
(...)/rest/price/search/bymetaboliteidorderbyprice/{id}	Returns the descending ordered prices related to the ID of the metabolite.	ID of the metabolite
(...)/rest/price/search/units	Returns a list with all the units available.	N/A
(...)/rest/price/search/currency	Returns a list with all the currencies available.	N/A
(...)/rest/price/search/getMetabolitePricePaginated/{metaboliteid}/{paginationIndex}/{pagesize}/{asc}/{sortBy}	Returns the selected page content about the prices for the specific metabolite.	ID of the metabolite, Page index, number of metabolites per page, order (ascending and descending), sort method
(...)/rest/price/search/getMetabolitePricesCount/{metaboliteid}	Returns the number of prices available for a specific metabolite.	ID of the metabolite
(...)/rest/metabolites/search/byid/{id}	Returns the metabolite searching by its ID.	ID of the metabolite
(...)/rest/metabolites/search/byinchi/{inchi}	Returns the metabolite related to the chosen InChI.	InChI of the metabolite
(...)/rest/metabolites/search/byinchikey/{inchikey}	Returns the metabolite(s) related to the chosen InChIKey.	InChIKey of the metabolite
(...)/rest/metabolites/search/bysmiles/{smiles}	Returns the metabolite(s) related to the chosen SMILES.	SMILES of the metabolite
(...)/rest/metabolites/search/bysourceexternalid/{source}/{externalid}	Returns the metabolite(s) related to the chosen external ID.	External source and External source ID

4

DEVELOPMENT

Since the primary purpose of this thesis is to create algorithms for the analysis of the data contained in the bioeconomics platform, the establishment of an easy way to access this information and analyze it is necessary. Hence, amidst this work, a new database called **bioanalysis** was generated. Take into account that all the steps related to the analysis and manipulation of the information will be done in this new database so that the bioeconomics data does not end up being managed directly. The algorithms developed in this study are available in https://github.com/SofiaMM/bioanalysis.

4.1 OVERALL ARCHITECTURE

With this dissertation's goals in mind, a series of steps were taken so they could be achieved (Figure 9). As demonstrated in the *bioanalysis* pipeline represented in figure 9, first, a connector was generated to allow the exchange of bioeconomics data. The objects retrieved were then converted to suitable objects so that the new database could include them. Subsequently, with the bioanalysis already populated, the data went through a preprocessing and an analysis step, accomplished in *Python* using the computational environment, *Jupyter Notebook*. All of these phases will be further explained below.

First and foremost, a fitting database was determined. Consequently, the bioanalysis, has the following six tables: **bioanalysis_metabolite_price**, **bioanalysis_metabolite**, **metabolite_name**, **external_source**, **external_source_id**, and **provider**. The figure 10 shows the relational model of the bioanalysis database. With this in mind, each table is expected to gather specific sets of information that were fetched from the bioeconomics platform.

First, all the prices ever retrieved from this platform are stocked in the **bioanalysis_metabolite_price** table. Those prices have the same properties as the ones in the bioeconomics, already introduced in chapter 3, subsection 3.2 (Figure 7). However, besides these properties, three more were added - the log_id, the metabolite_id, and the provider_id. The first one corresponds to the ID in the bioeconomics platform, which is useful to track the metabolite back. Whereas, the other two were necessary so that the price could be linked to one metabolite and one provider, already contained in their respective tables. For this reason,

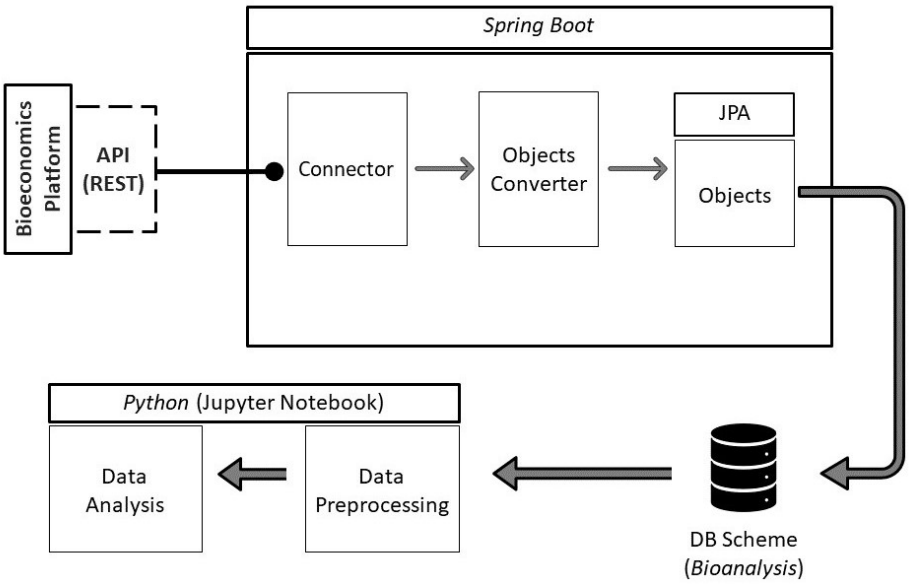


Figure 9: Bioanalysis Pipeline. Representation of the several steps included throughout this dissertation, for achieving a more efficient price analysis.

each price in the bioanalysis database can only be related to one metabolite and one provider at a time (Figure 10).

Next, in the **bioanalysis_metabolite** table, the retrieved metabolites are collected. Here, the new database only kept the InChI, the InChIKey, and the SMILES properties, already included in the bioeconomics platform metabolites (Figure 7). Also, the metabolite ID from the platform in question was saved as id_bioeco, for further distinction. Along with this new property, the common_name was also created to simplify the metabolite identification. In contrast to these properties, the map of external source ID's and the set of names were removed as metabolite properties. In fact, new tables in the database were created to accommodate these sets of information - the **external_source_id** table, the **external_source** table, and the **metabolite_name** table (Figure 10).

Subsequently, the **metabolite_name** table has only the name and a metabolite ID related to it, containing all the metabolite names, excluding the common ones. Additionally, both the **external_source_id** and the **external_source** tables include the information fetched from the map of external source ID's. To clarify, this map contains, for each metabolite, a list of external sources where it was found. Along with this list, each source contains a set of ID's that identify the metabolite in that source. Thus, in the **external_source_id** table there is a name, which represents the identification (ID) of the metabolite in question in the external source, a metabolite ID to relate that identification to just one metabolite, and an external source ID, that links that identification to just one external source that is stored in the **external_source** table with a corresponding database ID (Figure 10). Bear in mind that,

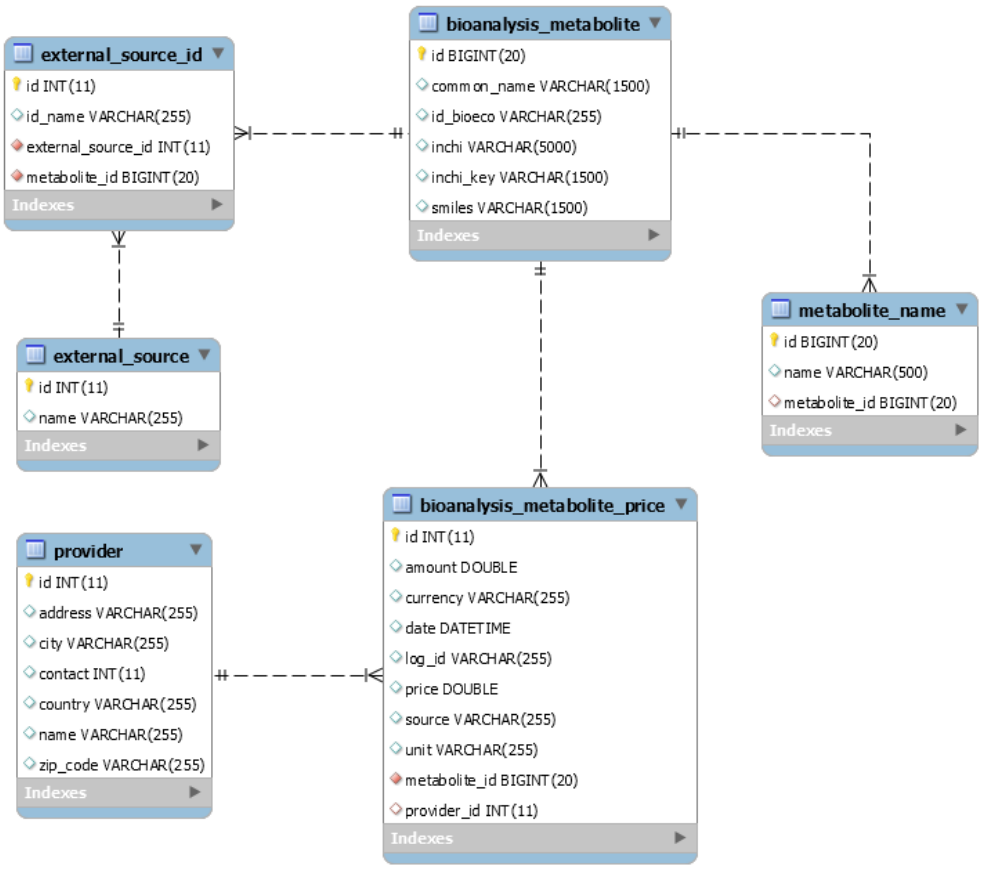


Figure 10: Relational model of *bioanalysis* database, exported from MySQL Workbench. In this model, six tables are displayed, as well as their relations. The bioanalysis_metabolite table saves the metabolite information and relates to the tables that gather the information of the external sources, other names and prices. The bioanalysis_metabolite_price table gathers the prices and relates to the tables that contain the metabolites' and providers' information.

in the **external_source_id** table, the name and the ID properties are not the same, the former corresponds to the ID of the metabolite given by the external source, and the latter to the ID given to that external source ID (represented by the name) by the bioanalysis database.

Besides, another distinction to be made is between the name in the **external_source_id** table and in the **external_source**, since the property in the second table corresponds to the actual name of the external source, and the other to the ID of the metabolite in that external source. Hence, a metabolite in bioanalysis can have many prices, many names associated, and many external sources, which in turn each can have numerous external source ID's (Figure 10).

Meanwhile, all the providers' information is contained in the **provider** table of the bioanalysis database. Admittedly, some information about the location and contact of the provider was thought to be valuable, not just the name. Therefore, this table has properties

such as the address, city, country, and zip code, as well as, the contact. Nonetheless, the data stored in this database was retrieved from the bioeconomics platform, which does not contain this information. Consequently, these properties are all null in the database, whereas the name of the provider is the only property for which this table has information (Figure 10).

4.2 DATA RETRIEVAL

To create the new database (Figure 10), it is then fundamental to have a connection to the bioeconomics platform. One approach that was considered in this development, was the construction of this database in *Java* programming language over the *Spring Boot* framework. This decision was based on what was explained in section 3.2, about the structure of the bioeconomics platform, so that the generation of the *bioanalysis* through the bioeconomics platform would be straightforward.

First, to design the *Java* application to run in the creation of the new database, including the API connector, the *Eclipse Integrated Development Environment (IDE)* was employed. In this IDE, a project named *bioanalysis* was developed, containing various class files with code for the different steps already mentioned above (Figure 9). In figure 11, an overview of the project in question can be visualized.

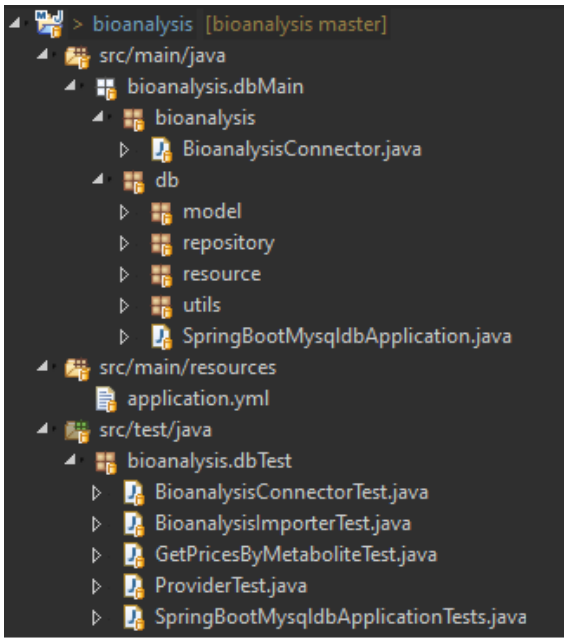


Figure 11: *Bioanalysis* project. Screenshot of the main packages and folders and their organization in this project from *Eclipse* IDE. The bioanalysis project contains two essential folders, named main and test. The former includes projects to connect and import the data, as well as create the database, whereas the latter consists of test files that check the code.

To best describe the fundamental class files, and packages developed in this project, table 7 was created. All the files shown here are necessary to create the bioanalysis database, and thus, are associated to the first three steps displayed in figure 9 (connector, object converter and database creation and population).

Table 7: Fundamental packages/files of the *Java* application. In this table are represented the class files and packages crucial to the development of the database and its population. The database was populated with data retrieved from bioeconomics.

Packages/Files	Function
bioanalysis/BioanalysisConnector.java	File that connects with the chosen REST API endpoints.
db/utils/BioanalysisImporter.java	File that accesses the data retrieved from the connector and converts it.
db/model	Package that contains a file for each new database's table, each creating the respective one.
db/resource	Package that contains files that create REST API endpoints for each table.
db/repository	Package that contains files that make the extension of the JPA repository possible, which helps to store the data in the new database.
BioanalysisImporterTest.java	File that tests the code in the BioanalysisImporter and BioanalysisConnector file.

Furthermore, while observing the figure 11, the files and packages represented in the first column of this table appear to be divided by folders. As noticed, the first five rows of the table include packages and files from the main folder, and the last row depicts a class file from the test folder. Although the last file is a test, it will be central to this phase.

Also, because of these class files and packages importance, they will be further explained in the subsections below.

4.2.1 *API Connector*

To generate the bioanalysis database from the *Java* application, a connector needs to be developed from the REST API points referred to in table 6, from section 3.3, so that the data from the bioeconomics platform can be collected (Figure 9).

As seen above, there is a class file with code for the creation of this API connector, the "BioanalysisConnector", located in the **bioanalysis** package (Figure 11, Table 7). Subsequently, the code in this class file connects with the chosen REST API endpoint, being able to access specific data from the bioeconomics platform that appears in JSON, as noted in section 3.3. In this case, the chosen REST API URL's need to comprise all the data related to the metabolites and prices. Namely, all the common names, InChI's, external source ID's and amounts, units, prices, among others, need to be included in these endpoints, so that the population of the database can take place.

Therefore, noticing table 6, in this work the "BioanalysisConnector" uses two of those available URL's. One contains the information about the metabolites through pages that could be selected by the user in the path variables (https://mendel.bio.di.uminho.pt/bioeconomics/

rest/metabolites/getMetabolitesPaginated/{paginationIndex}/{pagsize}/{order}/{sortBy}) (Table 6). The other REST API URL has all the information about the prices of each metabolite, these prices being searched by the different metabolite ID's fetched by the first URL (https://mendel.bio.di.uminho.pt/bioeconomics/rest/price/search/bymetaboliteid/{metaboliteid}) (Table 6). As mentioned in section 3.3, when using these REST URL's, the establishment of the path variables is still necessary. Hence, for this thesis, the path variables in the first URL related to the metabolites were set to 2 for the page size, "DESC" (descending) for the order, and ID for the sort. The other path variable, the pagination index, will not have a specific value (this will be explained later in this section). Likewise, the same was done in the REST URL that has the information about the prices. However, in this URL there is only a path variable, the metabolite ID, that, as already mentioned above, can be fetched with the first URL. Therefore, to populate the *bioanalysis*, the metabolite information from one REST API point can be used to find the price data from the other REST API point, since the metabolite ID is part of the info retrieved earlier.

Following this connection to the chosen REST API endpoints, the next thing that is needed is the adaptation of the JSON format to the *Java* language. This was done using the Jackson Binding, Maven dependency, that changes the data in JSON to the selected *Java* class (a class from the bioeconomics platform).

4.2.2 *Objects Converter*

To successfully create bioanalysis, other class files were conceived to complement the connector. These class files aid in fundamental steps that are necessary to reach the final database (Table 7). With this in mind, after connecting to the REST API endpoints and transforming the JSON data into manageable information, the next vital step is the conversion between the objects fetched from the bioeconomics platform to the new objects that will be implemented in the newly designed database for this dissertation (Figure 10). The reason why this conversion is necessary is that these objects are equivalent but not identical to one another, as depicted in figure 12. In other words, for each object related to each table from the bioanalysis database, the associated bioeconomics object had to undergo specific changes on its properties so it could be considered as a bioanalysis object and could be fitted in this database. Therefore, to make this conversion possible, the "BioanalysisImporter", located in the **utils** package, was handled (Figure 11, Table 7). In this file, the code imports the "BioanalysisConnetor" file, accessing the retrieved data and converting it.

To clarify, below are depicted the possible sets of changes encountered in the code from the "BioanalysisImporter" class file [1]:

1 Bear in mind, that the designations of the objects from the *Java* files are not exactly identical to the designations of the same objects on the MySQL Database. However, these designations are very similar.

- **Converting "BioeconomicsMetabolite" to "BioanalysisMetabolite"**: setting the InChI, the InChIKey, the SMILES associated to the "BioeconomicsMetabolite", using the name on the "BioeconomicsMetabolite" as the common name of the "BioanalysisMetabolite" and also setting the ID of the "BioeconomicsMetabolite" as the ID bioeco of the "BioanalysisMetabolite";
- **Converting "otherNames" from Bioeconomics to "MetaboliteName" from Bioanalysis**: save each of the metabolite names in the set "otherNames" as a name in "MetaboliteName" table with the corresponding metabolite;
- **Converting Hash Map of "SourceExternalId" from Bioeconomics to "ExternalSourceId" and "ExternalSource" from Bioanalysis**: save as "ExternalSource" the keys of the Hash Map from bioeconomics and as "ExternalSourceId" each ID that is in the list of a specific external source, not forgetting to set to the "ExternalSourceId", the "ExternalSource" related and the metabolite ID related;
- **Converting "BioeconomicsMetabolitePrice" to "BioanalysisMetabolitePrice"**: setting the price, the currency, the amount, the unit, the source [2] and the date associated to the "BioeconomicsMetabolitePrice" as a "BioanalysisMetabolitePrice", using the ID of the "BioeconomicsMetabolitePrice" as the log ID in *bioanalysis*;
- **Converting "Provider" from Bioeconomics to "Provider" from Bioanalysis**: set the "Provider" from bioeconomics as the name of the "Provider" from *bioanalysis*.

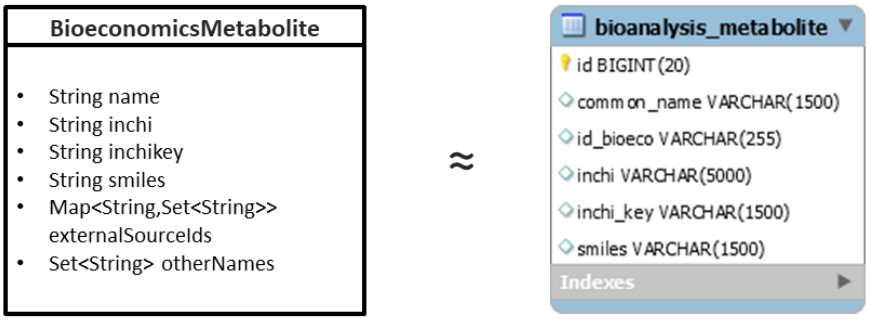


Figure 12: Comparison between the "BioeconomicsMetabolite" and the "BioanalysisMetabolite". As noticed, a new property was created, the id_bioeco, while the external source map and the other names set were removed from the properties. This is just one of the several objects that needed to be transformed so they could fit the bioanalysis database.

Furthermore, take into account that, to prevent errors in the bioanalysis database, such as duplication, it was assured that if for some reason the code fetched the same data

2 This source is not the same as the external source contained in the external_source table from the database. For more distinction read the section 3.2.

from bioeconomics more than once, the importer would only save the same object in the bioanalysis once.

4.2.3 *Database Conception*

Both the "BioanalysisImporter" and the "BioanalysisConnector" are crucial to retrieve the data from bioeconomics and place them in the bioanalysis database, so it can be managed later in *MySQL Workbench*. However, another step is necessary, the implementation of the database in the workbench. Therefore, in addition to all the class files already referred to in this chapter, this project also includes other files that contain the code needed to attain the six tables (Figure 10). These are located in the **model** package (Figure 11). Moreover, as mentioned above, to help store the data in the newly created database the use of Spring Data JPA is necessary, which, in turn, contributes to the development of six interfaces, each related to a class file of the **model** package. It is in these interfaces that the extension of the JPA repository occurs, thus being placed in a package named **repository** (Figure 11) (Walls, 2016; Webb et al.). Lastly, in the **resource** package, there are other class files that correspond to each of the tables created in the **model** (Figure 11). These resource files are crucial to create REST API points (Walls, 2016; Webb et al.).

As shown in the figure 11, both the **db** - that includes the packages considered above - and the **bioanalysis** packages are contained in the main source folder. Nevertheless, the test class files, that were created to check the developed code, are located in the test source folder. One of these files, the "BioanalysisImporterTest", assesses the code implemented in the "BioanalysisImporter" class file mentioned above, and consequently, the code in the "BioanalysisConnector". For this reason, the main function of this file in this project is to import all the data and to create the bioanalysis database, while being managed as a test.

As a result, considering the explanation above, the "BioanalysisImporterTest" was decided to be the file that would be running through this phase so that all the data from the bioeconomics platform could be imported to the generated bioanalysis database. Ergo, to keep the file continuously running, a *while* function in *Java* was established in order for the code to go through all the pages from the first REST URL mentioned (https://mendel.bio.di.uminho.pt/bioeconomics/rest/metabolites/getMetabolitesPaginated/{paginationIndex}/2/DESC/id). In this function, the path variable not specified before, the pagination index, is set to zero and its value is increased consecutively by one, after the two metabolites presented in the page and all their prices are imported and arranged in the proper database's table.

Consequently, this test was thought to run throughout all the pages attained from the bioeconomics platform, importing all the information available about the metabolites and the respective prices. Nonetheless, bear in mind that, in the context of this thesis, because of

the considerable amount of time it takes to run the code, the "BioanalysisImporterTest" only reached the page index 2151, not going through all the data of the bioeconomics platform.

4.3 THE NEW DATABASE - *bioanalysis*

Following the creation and population of the database, a basic analysis was accomplished in *MySQL*. Even though the new database does not comprise the totality of the bioeconomics platform, from the 2151 pages imported, it resulted **4304 metabolites**, 262 external sources, 131 providers and **1038707 prices**, which appears to be sufficient for the context of this work.

On top of this, other elements were noticed in the database prices. Displayed in table 8 are the values of three of the prices' properties. Regarding these properties, it is noted that values do not vary much in bioanalysis in comparison to the remaining ones.

Table 8: Distinct values for the currency, unit and source properties of the bioanalysis_metabolite_price table from the bioanalysis database.

Properties	**Values**
Currency	USD
Unit	g, mg, kg, mg/ml, $mg/5ml$, μmol, ml and l
Source	CHEMSPACE, DRUGBANK, MolPort, OxChem

As uncovered by this table, it is noted that these prices only have four sources associated, CHEMSPACE, DRUGBANK, MolPort and OxChem. This corroborates to what was noted in the Metabolic Price Retrieval, explained in section 3.2.2. This step utilizes these sources to encounter prices and, thus, all prices from bioanalysis were gathered from these specific sources.

In addition, all prices available in this database solely have the USD currency. This makes it easier to compare and analyse them, since there is no need to convert this property. On the other hand, the prices have 8 different units (g, mg, kg, mg/ml, $mg/5ml$, μmol, ml and l) from different types - g, mg, kg are mass units, l and ml are volume units, mg/ml, $mg/5ml$ are density units and μmol is an amount of substance unit - that can not be converged into a single one unit, creating the need to analyse the prices in separate sets. Consequently, since analysing various sets of prices is not ideal, a solution for this conversion problem is essential (this matter will be further explained in the section 4.3.2).

4.3.1 *Data Issues*

First, every table of this new database was examined and, in some of them, issues were detected. As mentioned in section 2.3.1, the existence of various problems in the data is

common. Consequently, a way to solve these complications is fundamental, so that the price analysis can be legitimate. These initial problems were easily classified into two major groups: missing data and duplicate data. They usually happen while retrieving the information, since it may exist some faults in the methods or algorithms employed.

Furthermore, some other situations, misrepresenting the information gathered, were identified in more recent analysis performed in this dissertation. These recent issues seem to happen in prior steps of the data retrieval, for instance while integrating prices in the bioeconomics platform or in farther steps that do not happen in the platform. In this work, all these situations will fall into a single category, the integration problem.

In sum, there were a set of errors intrinsically found in the data from the database, where in some situations the data assimilation from the bioeconomics was creating more objects than it should, and in others it was creating less objects than it should. Nonetheless, these problems complicate the price analysis and, hence, they are explained in the sections below.

Missing Data

In the bioanalysis database, the missing data problem is seen in the metabolites and the provider tables. As mentioned in section 2.3.1, data can often contain inaccuracies, which in some cases translate into missing values in crucial properties of the objects.

Regarding the metabolites table, while examining it and counting the number of lines that exist in each property, it is possible to observe that each metabolite has a related InChI, InChIKey and SMILES. However, some metabolites do not have a common name associated with them, since the number of names is lower compared to the number of metabolites (Figure 13 (**A**)). This means that some metabolites were fetched without a common name, having that property as "Null" (Figure 13 (**B**)). An example is seen in the figure below.

A

num_id	num_commonName	num_idBioeco	num_inchi	num_inchiKey	num_smiles
4304	4163	4304	4304	4304	4304

B

id	common_name	id_bioeco	inchi
8	NULL	9221877112145259159	InChI=1S/C61H86N13O13P.CN.Co/c1-29-21-3...
20	NULL	9218130609997550993	InChI=1S/C21H22N2O7.ClH/c1-23(2)14-13-16(...
22	NULL	9217712383863453606	InChI=1S/2C33H35N5O5.C4H6O6/c2*1-32(35-...
40	NULL	9208529877926441543	InChI=1S/C23H23N3O5.ClH/c1-4-23(30)16-8-1...
78	NULL	9195321569871133783	InChI=1S/C20H36O4/c1-5-9-11-15(7-3)13-17(...

Figure 13: *Printscreen* of output from bioanalysis_metabolite table in *MySQL Workbench*. (**A**) The query from this output calculates the number of lines for each column in the metabolite table. Consequently, as observed in the red box, the number of existing common names is lower than the number of existing metabolites, verifying the existence of a number of metabolites that do not have common names associated. (**B**) In this output some examples of the situation above are represented, showing five metabolites with different ID's and InChI's, and without a common name (red box).

Next, in the provider table, it was initially depicted that there are 131 providers in the database. To verify this number, a query similar to the one used in the metabolites' analysis was developed, which counted the number of lines presented in each column for the providers' table [3]. Nevertheless, both the query counting all the lines and the one counting the unique values, gave the same output represented in figure 14 (**A**). As displayed, there is minus one name in comparison to the number of ID's presented in the table. This indicates that one provider has a "Null" value in the name column. To confirm this affirmation, figure 14 (**B**) illustrates the provider in question. Ergo, there is evidence that a case of missing data is occurring in the providers' table.

Figure 14: *Printscreen* of output from provider table in *MySQL Workbench*. (**A**) This output represents both the query associated with the sum of all the lines in each column of the providers' table, as well as the query calculating only the unique values in each column. (**B**) Output of the provider that has a "Null" value in the name column.

Both these situations verify the missing data problem, turning it an issue that needs to be dealt with before any data analysis.

Duplicate Data

Concerning the duplicate data problem, some other shreds of evidence were also found. As explained below, both the metabolite table and the price table contain data that is duplicated. Here, the information includes some redundancies, including objects that appear repeated. To understand the difference between an object with some repeated properties and an entirely repeated object is necessary to take into consideration that some values are unique to the metabolite or price in question.

Subsequently, the examination of the bioanalysis_metabolite table showed that the counts of the different values in each property were not the expected results. If no errors were to be found, these counts should be equal to the results in figure 13 (**A**). As this was not the case, and since the missing data problem has already been interpreted, an examination of these results is required (Figure 15).

3 Be aware that, as already mentioned in this chapter, most of the columns in this table are null. Therefore, the queries developed for this analysis will only take into account the ID and the name column.

num_id	num_commonName	num_idBioeco	num_inchi	num_inchiKey	num_smiles
4304	4085	4304	4278	4278	3969

Figure 15: *Printscreen* of output from bioanalysis_metabolite table in *MySQL Workbench*. In this figure, the number of distinct values contained in each column of this table is displayed, demonstrating that the common name, InChI, InChIKey and SMILES properties have fewer values comparing to the number of existing metabolites.

In fact, multiple columns possess lower numbers in comparison to the values presented in the prior query (Figure 13 (**A**)), reinforcing the existence of some duplicated properties. However, as stated above, not all duplicate properties mean that the metabolite is also duplicated. As mentioned in chapter 3.2, the InChI is the only property unique to the metabolites. Consequently, there should be no different metabolites with the same InChI. However, when comparing the number of unique InChI's and individual metabolites (number of ID's), they do not match (Figure 15). This conclusion can confirm that certain metabolites can be found repeated in the database.

Alternatively, other properties may be duplicated, such as the common names (Table 9). Nevertheless, in this situation, none of them are unique to the object, since they do not identify a single metabolite, which makes the existence of duplicates in these properties more reasonable. As a consequence, these properties cannot confirm the duplication of the object in question.

Table 9: Representation of the count of metabolites for each common name. The column "count_id" shows the number of ID's that are associated with each common name, confirming that some names are presented in more than one metabolite.

common_name	**count_id**
Telithromycin	5
Temsirolimus	4
pimaricin	4
cholecalciferol	3
ATORVASTATIN CALCIUM	3

Additionally, whilst finding the reason for the disparity between the number of InChI's and number of ID's, the existence of duplicated metabolites saved in the bioanalysis database is confirmed. In table 10, a number of existing metabolites in this situation are represented. In this table, each metabolite is grouped by InChI, counting for each of them the number of ID's that are associated. Take into account that the group by did not include the common name, since, as already mentioned above, some metabolites are different but have the same name (Table 9). Subsequently, it is confirmed that there are some metabolites with more

than one ID per InChI in the database, meaning that the same metabolite is presented more than one time, and consequently is duplicated. Overall, in this database, using this query, 26 duplicates exist.

Table 10: Representation of the number of metabolites with the same related InChI. The column "count_id" shows the number of ID's that are associated with each InChI, showing that some InChI's are shared by different metabolites. This table displays the first ten metabolites with the duplicate situation, ordered by number of ID's.

common_name	inchi	count_id
Cyanokit	InChI=1S/C62H90N13O14P.Co.H2O/c1-29-20-...	3
D-cycloserine	InChI=1S/C3H6N2O2/c4-2-1-7-5-3(2)6/h2H,1,...	2
Betaxolol hydrochloride	InChI=1S/C18H29NO3.ClH/c1-14(2)19-11-17(...	2
Temsirolimus	InChI=1S/C56H87NO16/c1-33-17-13-12-14-18...	2
CLONIDINE HYDROCHLORIDE	InChI=1S/C9H9Cl2N3.ClH/c10-6-2-1-3-7(11)8(...	2
Glycine hydrochloride	InChI=1S/C2H5NO2.ClH/c3-1-2(4)5;/h1,3H2,(...	2
Creatine phosphate disodium salt	InChI=1S/C4H10N3O5P.Na.H2O/c1-7(2-3(8)9...	2
deltamethrin	InChI=1S/C22H19Br2NO3/c1-22(2)17(12-19(2...	2
L-cystine	InChI=1S/C6H12N2O4S2/c7-3(5(9)10)1-13-14...	2
Iothalamate sodium	InChI=1S/C11H9I3N2O4.Na/c1-3(17)16-9-7(1...	2

To comprehend this duplication issue, other queries were built in *MySQL Workbench*. Because the InChI - property that should be unique to each metabolite - repeats in the database, the duplicated metabolites were individually analysed, so that the source of this problem could be understood. As noted in table 11 the examination of the *Temsirolimus* confirmed that the same metabolite with the same InChI, has a different id_bioeco. This occurrence was also observed in the other duplicated metabolites.

Table 11: Example of duplicate metabolites (*Temsirolimus*). In this output, the same metabolite is observed with the same InChI, however it does not contain the same ID from bioeconomics platform, having different id_bioeco's. This confirms that there are two *Temsirolimus* metabolites in the bioanalysis database, as shown in table 10.

common_name	inchi	id_bioeco
Temsirolimus	InChI=1S/C56H87NO16/c1-33-17-13-12-14-18...	9162562803064309984
Temsirolimus	InChI=1S/C56H87NO16/c1-33-17-13-12-14-18...	7997890823186803415

Taking into consideration that, as stated above, the algorithms created for the database specifically prevent the integration of duplicate objects (Section 4.2.2), this duplication error did not happen when importing the data from bioeconomics to the bioanalysis database, but

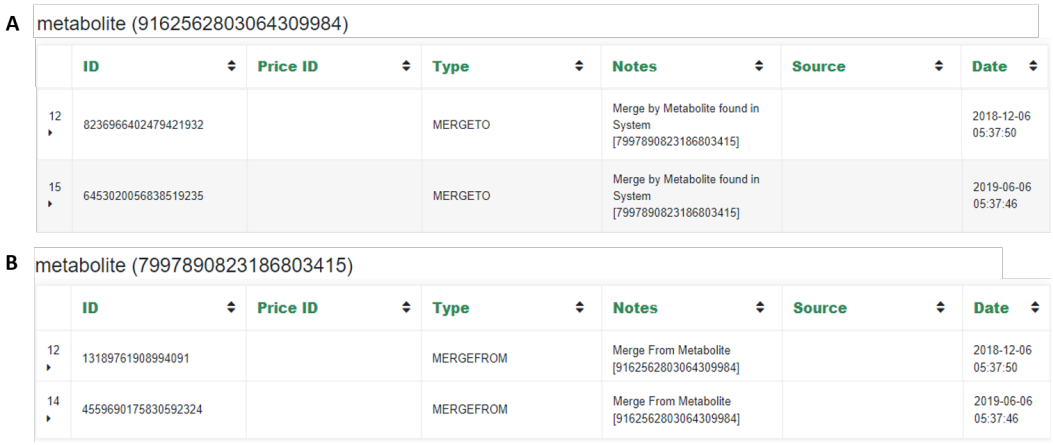
in a previous step. Due to the example from table 11, there is evidence that the metabolic enrichment pipeline is adding identical metabolites as different, which as a result contributes to distinct ID's in the platform. Consequently, because the metabolites were retrieved with different id_bioeco (Table 11), the bioanalysis algorithm considered them as different.

After an analysis via website of the bioeconomics platform, this situation was encountered, as anticipated. In this case, the *Temsirolimus* metabolite was found duplicated, as shown in figure 16. However, while trying to understand the reason behind this duplication, a merge command between the two metabolites was discovered, thus affirming that the bioeconomics had already identified both as being the same (Figure 17). Nonetheless, the merge was not successful considering that they are still presented independently in the platform. Bear in mind this condition was also seen in other examples besides the *Temsirolimus* metabolite.

Figure 16: *Printscreen* of the duplicated metabolite, *Temsirolimus*, in the bioeconomics platform's website. As shown in this figure both metabolites have the same properties, however the metabolite ID from this platform is different (red boxes). This *printscreen* was taken from the platform URL (https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolites).

Taking into account the bioanalysis_metabolite_price table, the situation observed above is also encountered. Nevertheless, in this table, no property can be considered exclusive to just one price (Table 2), in exception to the ID and log_id (ID from the bioeconomics). Hence, when comparing the number of unique values in each property with the number of lines each property has, it is expected that the former is lower in most columns (Figure 18). That is also why only those two properties (ID and log_id) have 1038707 unique values,

confirming that there is a different value per line [4]. In addition, the numbers corroborate with the total number of prices depicted before in the database (1038707 prices).



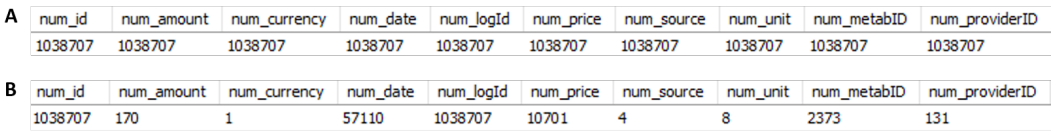
A metabolite (9162562803064309984)

	ID	Price ID	Type	Notes	Source	Date
12	8236966402479421932		MERGETO	Merge by Metabolite found in System [7997890823186803415]		2018-12-06 05:37:50
15	6453020056838519235		MERGETO	Merge by Metabolite found in System [7997890823186803415]		2019-06-06 05:37:46

B metabolite (7997890823186803415)

	ID	Price ID	Type	Notes	Source	Date
12	1318976190899409 1		MERGEFROM	Merge From Metabolite [9162562803064309984]		2018-12-06 05:37:50
14	4559690175830592324		MERGEFROM	Merge From Metabolite [9162562803064309984]		2019-06-06 05:37:46

Figure 17: *Printscreen* of the merge action from each *Temsirolimus* metabolite by the bioeconomics platform. (**A**) In the top *Temsirolimus* metabolite there was a commanded "MERGE TO" in December 2018, however that evidently did not occur, and a second merge was required by the platform in June 2019 (retrieved from https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/9162562803064309984). (**B**) In the bottom *Temsirolimus* metabolite the same has occurred, where a "MERGE FROM" command was given in the same dates, December 2018 and later in June 2019 (retrieved from https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/7997890823186803415).



A

num_id	num_amount	num_currency	num_date	num_logId	num_price	num_source	num_unit	num_metabID	num_providerID
1038707	1038707	1038707	1038707	1038707	1038707	1038707	1038707	1038707	1038707

B

num_id	num_amount	num_currency	num_date	num_logId	num_price	num_source	num_unit	num_metabID	num_providerID
1038707	170	1	57110	1038707	10701	4	8	2373	131

Figure 18: *Printscreen* of output from bioanalysis_metabolite_price table in *MySQL Workbench*. (**A**) The query from this output calculates the number of lines in each column for the metabolite prices' table. (**B**) In contrast, the query from this output calculates the number of the distinct values for each column in the metabolite prices' table. In other words, how many unique values exist in each price property.

Subsequently, the number of unique price objects in the database was calculated, keeping in mind the condition noticed above. To calculate them, a join between the amount, the unit, the price, the currency, the date, the source, the provider and the metabolite was necessary to find how many of these combinations appear only one time in bioanalysis so they can be consider a unique price. Hence, a query for this task was developed, where the output is shown in the column "num_price" of the figure 19. When comparing the result with the

4 Consider that in the amount, price and date properties there is a high number of distinct values, since they correspond to all the available dates, metabolite quantities and price values in this whole database. Despite this, is probable that some these values are duplicated.

number of ID's in the price table, there was a confirmation that the number of real prices is lower to the one initially assumed, since the ID is always unique for each price line (Figure 19). For this reason, this result indicates a duplication of the prices, similar to the one found in the analysis of the metabolites (46923 duplicate prices).

num_id	num_price
1038707	991784

Figure 19: *Printscreen* of the number of prices from the bioanalysis_metabolite_price table in *MySQL Workbench*. The query from this output calculates the number of ID's in comparison with the number of distinct prices that are present in the metabolite prices' table from bioanalysis database.

Furthermore, in contrast to the number of metabolites duplicated, the number of duplicate prices is higher, considering that, in the bioanalysis_metabolite_price table, some values were saved more than 50 times (Figure 20).

id	amount	currency	date	log_id	price	source	unit	metabolite_id	provider_id	num_price
380060	20	USD	2017-09-22 17:55:02	882945209732730073	0.82	DRUGBANK	mg	1465	8	135
380053	40	USD	2017-09-22 17:55:02	7716852278305210358	0.82	DRUGBANK	mg	1465	8	135
267143	4	USD	2017-09-22 17:55:03	3362019816119351520	7.9	DRUGBANK	mg	1021	8	108
692862	4	USD	2017-09-22 17:55:03	6052307429296381730	7.9	DRUGBANK	mg	2794	8	108
267292	8	USD	2017-09-22 17:55:03	226365596665999672	12.06	DRUGBANK	mg	1021	8	108
693666	8	USD	2017-09-22 17:55:03	2030678443908049832	12.06	DRUGBANK	mg	2794	8	108
617734	10	USD	2017-09-22 17:55:02	4662041673623115219	1.17	DRUGBANK	mg	2476	8	90
890710	10	USD	2017-09-22 17:55:02	6464172842447987744	1.17	DRUGBANK	mg	3599	8	90
617435	20	USD	2017-09-22 17:55:02	6823329254637884939	1.45	DRUGBANK	mg	2476	8	90
890717	20	USD	2017-09-22 17:55:02	2223660693549015291	1.45	DRUGBANK	mg	3599	8	90

Figure 20: *Printscreen* of the first ten lines of the output from the bioanalysis_metabolite_price table in *MySQL Workbench*. The query from this output counts the number of prices (num_price column) presented in each group by amount, date, price, source, unit, metabolite and provider. These lines are displayed by descendant order of the num_price column.

For a more specific analysis of this duplication problem, another query was executed that displayed the comparison between the unique prices and the total prices, considering the metabolite associated (Table 12). In this table, the first 13 metabolites saved in the database are illustrated. Some of them have the same number in the last two columns, noting that these metabolites do not have duplicated prices. On the other hand, others have different prices, with lower prices in the num_dist_price column, showing the duplication situation.

As a result, a thorough examination of the metabolites that include duplicate prices was performed. In table 13, the *Telithromycin* is adopted as an example. To clarify, the prices have the same amount, date, price, source and unit, as well as, the same metabolite and provider associated. Thus, this proves that only the log_id value is different in all the prices (Table 13).

Table 12: Output of the number of prices and distinct prices grouped by metabolite ID. The difference between both values is that the latter only counts the values with distinct price, amount, date, source, unit and provider, while the former counts all prices without distinction, including the log_id from bioeconomics.

metabolite_id	common_name	num_price	num_dist_price
1	tobramycin	647	557
2	Telithromycin	30	3
4	3,5-DIMETHYLPHENOL	1274	1251
5	5-Oxooctanoic acid	260	260
7	3-Chloro-5-fluoroisonicotinic acid	387	387
9	2-(Hexyloxy)ethanol	361	361
10	Methyl jasmonate	18	18
11	DIPHENOXYLATE	30	14
12	DIETHYLCARBAMAZINE CITRATE	350	350
13	4'-Hydroxyacetophenone	2796	2769

Table 13: Example of duplicate prices (*Telithromycin*). In this output, the same price (6.12 USD) with the same date, source, unit, amount, metabolite and provider associated is displayed in the bioeconomics platform ten times, all with different log_id's. This confirms that there are duplicated prices associated to *Temsirolimus* in bioanalysis, as shown in table 12.

amount	currency	date	log_id	price	source	unit	metabolite_id	provider_id
400	USD	2017-09-22 17:55:03	82492135741291800 93	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	81649511729303279 49	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	45217354446153190 79	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	54491468565377792 09	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	75364784252466528 28	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	46282308334667223 22	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	21384867562359315 50	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	86813392683158620 83	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	422519945569557 06	6.12	DRUGBANK	mg	2	8
400	USD	2017-09-22 17:55:03	89766611071766841 52	6.12	DRUGBANK	mg	2	8

In short, as seen with the metabolites' table, the code to create the bioanalysis database assumed that these prices were different because they have a distinct ID from the bioeconomics platform (log_id). To corroborate this affirmation, these examples were discovered in the bioeconomics website, already in the duplication situation. For instance, this platform considers the prices of table 13 as different, since they have different ID's (Figure 21). Nonetheless, while observing this figure, is evident that all the prices have the same properties. Take into account that the same situation was also observed in other duplicate prices present in bioanalysis.

	ID	Amount	Units	Price	Currency	Provider	Source	Date
1	8249213574129180093	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
2	8164951172930327949	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
3	4521735444615319079	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
4	5449146856537779209	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
5	7536478425246652828	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
6	462833083466722322	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
7	2138486756235931550	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
8	8681339268315862083	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
9	42251994556955706	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02
10	8976661107176684152	400	mg	6.12	USD		DRUGBANK	2017-09-22 06:55:02

Figure 21: *Printscreen* of the duplicated prices from *Telithromycin* metabolite in the bioeconomics platform's website. As displayed in this figure these prices contain the same properties, however the price ID from this platform is distinct (ID column), confirming the duplication shown in table 13. This *printscreen* was taken from the platform URL (https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/9222710493671371933).

To sum up, since there are duplicate prices and metabolites in the database, these also need to be solved before the price analysis.

Integration Problem

Next, the integration problem, mentioned above, will be explained. This issue was found later in this work, when performing specific case studies and, since this problem arises before the creation of the new database, its resolution was challenging.

In this issue, the bioanalysis database has different prices for the same provider, source, unit, amount, and month, whilst analysing an individual metabolite. However, this situation is not expected, since the metabolic price retrieval occurs every month, retrieving only one price for these specific properties sets (Section 3.2.2). Thus, there should only be a price for these sets in each month of a metabolite. Consequently, in the metabolites where this

problem occurs, they contain different prices for these same properties because some of those values were faultily integrated, belonging in reality to another metabolite.

To introduce the information provided above about the issues' cause, and also find the location of these integration faults, the prices were previously examined in the bioeconomics platform. As a result, two possibilities for the origin of this problem were identified, deriving from two completely different steps of the price and metabolite combination in this platform. The issue that leads to different prices in the same provider, source, amount, unit and month was detected in the **metabolic enrichment step** from SISBI, as well as, in the **sources** themselves. Two examples of these circumstances are presented above.

In figure 22, an example of this problem is displayed, using the 4-aminopyridine metabolite. As shown, these two prices are identical, with the exception of the price value and the fact they were retrieved two seconds apart. Besides, while observing the notes of each price demonstrated in this figure (22), it is possible to identify the origin of these values. Both were retrieved from the CHEMSPACE source but with different ChemSpaceID's. Hence, a thorough search in the CHEMSPACE source pinpointed that the first ChemSpace ID presented in figure 22 (CSC000211974) represented the 4-aminopyridine metabolite, whereas the second (CSC000217147) did not represent the metabolite in question but the 3-(4-bromophenyl)propanoic acid metabolite (Figure 23).

Figure 22: *Printscreen* of the prices from the 4-aminopyridine metabolite (id_bioeco = 88071033288822559O3) and *ChemShuttle* provider in the bioeconomics platform. As observed in the notes for each price, the ChemSpaceID adopted to retrieve the value is different in the two prices (red boxes). The higher value (123 USD) is associated to the ChemSpaceID CSC000217147, whereas the lower price (40 USD) is associated to the ChemSpaceID CSC000211974. This information was retrieved from https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/8807103328882255903.

Similarly, in some other providers of this metabolite, such as the *ENAMINE Ltd.*, the same situation occurs. Although, in this case, the external source ID's that were merged belonged to the *MolPort* source. Furthermore, the added MolPort ID that does not represent the 4-aminopyridine (MolPort-000-152-357), represents the metabolite mentioned above, the 3-(4-bromophenyl)propanoic acid (MolPort-000-146-022). Thus, this might entail that there

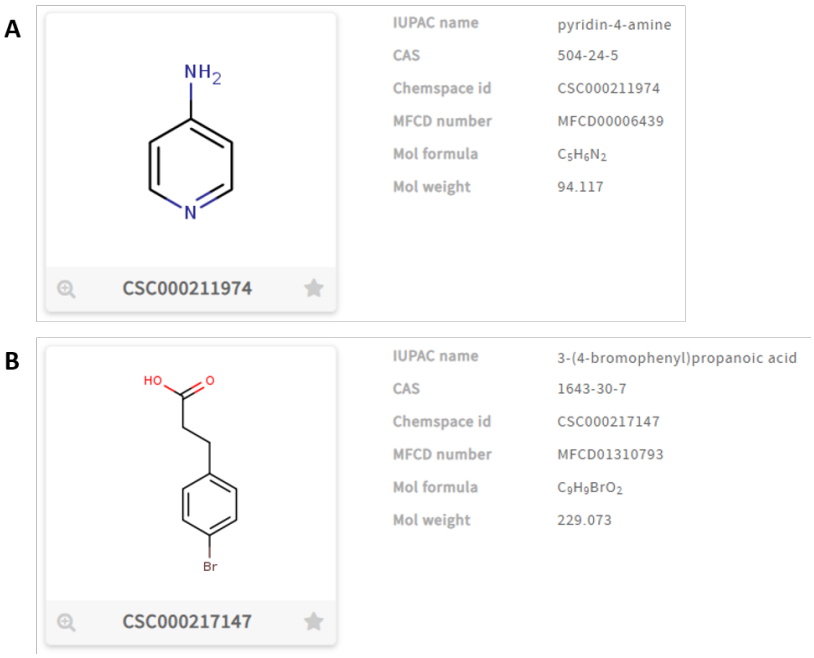


Figure 23: *Printscreen* of the 4-aminopyridine and 3-(4-bromophenyl)propanoic acid metabolites in the CHEMSPACE website (https://chem-space.com/). (**A**) The first metabolite represents the 4-aminopyridine and is identified by the ChemSpaceID CSC000211974 in the source in question. This information was retrieved from https://chem-space.com/search/5e347b6a-760c50-c1895cb4-3d04890/CSC000211974?currency=usd&uom=g. (**B**) The second metabolite represents the 3-(4-bromophenyl)propanoic acid metabolite and is identified by the ChemSpaceID CSC000217147 in the source in question. This information was retrieved from https://chem-space.com/search/5e347b93-27ce3c-c1895cb4-16c0bcd/CSC000217147?currency=usd&uom=g.

is a possibility that the bioeconomics platform combined the external sources of this other metabolite with the 4-aminopyridine information. Proof of the aforementioned theory is the inclusion of the name 3-(4-bromophenyl)propanoic acid as one of the names of the metabolite 1175, which is undoubtedly inaccurate.

On top of this, a second example with the methane metabolite demonstrates that this integration error may not derive from the bioeconomics platform, but from the source. As explained in section 3.2, this platform gathers the prices from the sources, that in turn retrieved the providers' prices. In this case, the different values were retrieved from the same MolPort ID, MolPort-018-618-244, that corresponds to the metabolite in question (Figure 24). However, as displayed in figure 24, in the source website there are various CAS numbers related to this ID. Hence, the error comes from MolPort itself, due to the lack of uniqueness in the CAS number, which should be an exclusive identifier of the metabolite (Section 3.2).

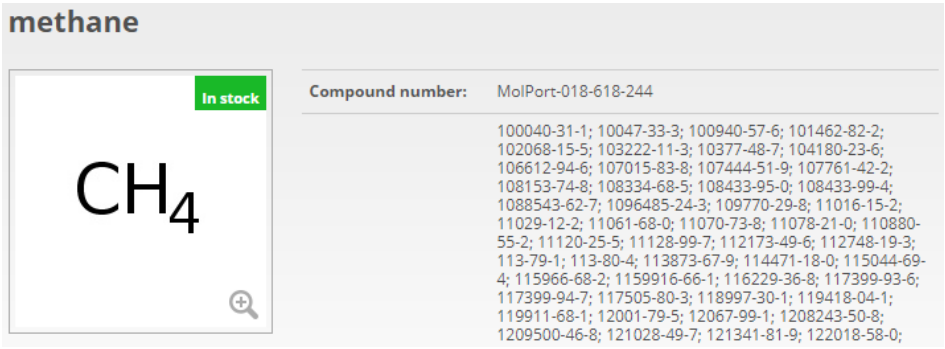


Figure 24: *Printscreen* of the methane metabolite in the MolPort website (https://www.molport.com/). As displayed in this figure, the metabolite contains several CAS numbers associated in the source in question. This information was retrieved from https://www.molport.com/shop/moleculelink/methane/18618244?searchtype=text-search&searchkey=4DDM99JL3CI141DVUF90OB.

4.3.2 *Problem Solving - Preprocessing Methods*

Considering the problems described above, the next step done in this thesis was to filter out any insignificant data and errors present in the bioanalysis database. After all, this database has information that might disrupt the price analysis.

As mentioned above, to manage these issues, the preprocessing steps were executed in *Jupyter Notebook*. Thus, no changes were done in the actual database, to maintain the raw data in a specific location.

Data Cleaning and Normalization

Since the goal of this work is to analyse the prices of the chemical compounds, the first task was to create a query that retrieved them for *Jupyter Notebook* so they could be examined, making sure that the packages required for the connection to *MySQL* were already imported. In this query, the prices to analyse were carefully chosen to be more suitable for the goal. Hence, some precedents were taken into account:

1. Noticing that, to analyse prices with **different units**, they need to be converted, it is necessary to develop a view on the units that will be managed in this analysis.

   First, as mentioned already, there are four types of units - mass units, volume units, density units, and units for the amount of substance. Therefore, to understand how much each type influences the *bioanalysis* data, a calculation was performed to find out how many prices were presented in each set. Since the database has duplicate prices, this calculation was executed, including solely different prices. Consequently, the result showed that 96.3% of the prices comprised units of mass (954912 of 991784 prices), where only 3.7% contained other units (36872 of 991784 prices). To easily

perceive how much these results actually influence the data, they were illustrated in a pie chart, as observed in figure 25.

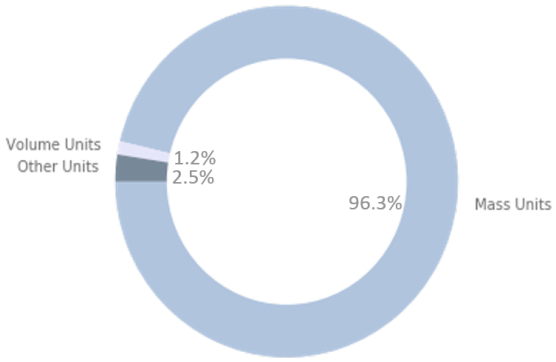


Figure 25: Representation in a pie chart of the percentage of prices for each unit category. Mass units include g, mg and kg. Volume units include l and ml. And other units include mg/ml, $mg/5ml$ and μmol. As noticed in this figure, the majority of prices belong to the mass units group (96.3% of prices).

As a result, considering these percentages, the price analysis will merely include the values that are in the metric system unit of mass (g, mg and kg). When eliminating the other types of units, it becomes possible to convert the prices into one unique unit and ergo, efficiently analyse the data.

2. Another crucial decision to make is the choice of the **time window** from which the prices participating in the analysis are collected.

   The time interval included in the database *bioanalysis* is from September 2017 to July 2019. However, some metabolites do not possess prices before 2018, or in some cases, the prices that do appear have high discrepancies in comparison to the remaining values. Other fact is that some providers also lack data from a few months of the year 2017. In addition, the database created in this dissertation imported the information from the bioeconomics platform until mid-July 2019, increasing the possibility of incomplete data in the last month. Consequently, since prices are updated once a month, some may not yet have been updated in bioeconomics, which in turn makes the best choice for the last month of the time window, June 2019.

   Hence, to make sure the data managed in the analysis was suitable for a better study, the data was compressed in a narrower period, being the time window chosen between January 2018 and June 2019.

Accordingly, taking into account the criteria explained above, the query created to connect the database prices to *Jupyter Notebook* focused only on mass unit prices that were between January 2018 and June 2019, filtering out any data that did not comprehend these limitations.

Besides, to easily apply the prices in future calculations, this query, in addition to having the columns from the bioanalysis_metabolite_price table, also generated new columns that represent the prices per unit and per gram, using a function to convert the units into grams in the latter. Finally, this query immediately solved the null provider issue that was associated with some prices. Since the provider name is "Null", it would appear as a null value in the dataframe, when in fact, it exists. So, in order to counteract this effect, a name of 'None' was given to identify the provider and to be counted as one existent value.

Table 14: First five results of the df dataframe. This dataframe was created in *Jupyter Notebook* with a query in *MySQL* that connected to the *bioanalysis* database and imported the prices enclosed between January 2018 and June 2019, associated to mass units.

date (Index)	metab_id	metab_name	price	amount	unit	provider_name	source	price_per_unit	price_per_g	year	month
2018-01-11 21:39:13	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.0	158.0	2018	1
2018-02-11 23:46:53	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.0	158.0	2018	2
2018-03-11 23:47:07	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.0	158.0	2018	3
2018-04-11 22:47:14	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.0	158.0	2018	4
2018-05-11 22:47:27	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.0	158.0	2018	5

The dataframe achieved by this query will be defined as **df**. Its first five lines are displayed above in table 14. Observing the table, it is concluded that the obtained dataframe includes a date-time index called date, and several columns, such as, metabolite id, metabolite name, price, amount, unit, provider name, source, price per unit, price per gram, year and month. Because the restrictions filtered out some of the data, the new dataframe has just **836888 prices** and **2283 metabolites**, in comparison to the 4304 metabolites and 1038707 prices contained in the bioanalysis database [5].

Furthermore, when thoroughly analysing this dataframe, some null values were found in the metab_name column (364 null values), and in both the price_per_unit and the price_per_g columns (each having 295 null values). Given what was referenced in the previous section (4.3), null values in the metab_name column were expected. However, as to the null values in the other columns, none were previously disclosed when previously analysing the price column from the database. Hence, the reason identified behind this situation was the existence of 295 rows in the dataframe with amounts equal to zero.

When examining directly the database, there were exactly 295 rows in the bioanalysis_metabolite_price table with this issue. This analysis also established that all 295 rows have prices derived from the TargetMol provider and the CHEMSPACE source. Nonetheless, when analysing the database prices on a more general basis, it appears that not all prices retrieved from these places have null amounts. In fact, the average amount value in this combination of

5 In subsection 4.3.1, evidence of duplication in prices and metabolites was confirmed. However, please take into account, that when connecting the *Jupyter Notebook* to the database, the dataframe will still include duplicates, since they were not eliminated initially from the database.

provider and source is 15.6, not being the lowest of all possible combinations. Consequently, eliminating all prices collected from TargetMol and CHEMSPACE is not an option to solve this situation. To make sure that these amounts were not generated by the code developed to create the new database, this problem was sought in the bioeconomics platform. As noticed in figure 26, the prices with the null amounts reside in the platform from where the bioanalysis retrieved the data.

	ID	Amount	Units	Price	Currency	Provider	Source	Date
1	8808219134262402108	0	g	105	USD	TargetMol	CHEMSPACE	2018-10-22 04:22:54
2	8346643701824216110	0	g	105	USD	TargetMol	CHEMSPACE	2018-11-22 04:23:14
3	1886955925346535526	0	g	105	USD	TargetMol	CHEMSPACE	2018-12-07 02:41:15
4	4752403362060110629	0	g	105	USD	TargetMol	CHEMSPACE	2019-01-07 02:41:26
5	8870242851751346960	0	g	105	USD	TargetMol	CHEMSPACE	2019-02-07 02:41:34
6	3029819730035677477	0	g	105	USD	TargetMol	CHEMSPACE	2019-03-07 02:41:44
7	2536049465887379304	0	g	105	USD	TargetMol	CHEMSPACE	2019-04-07 02:41:50
8	4532477412655713202	0.01	g	230	USD	TargetMol	CHEMSPACE	2018-11-22 04:23:14
9	329682271882631026	0.01	g	230	USD	TargetMol	CHEMSPACE	2018-12-07 02:41:15
10	2819235571330511682	0.01	g	230	USD	TargetMol	CHEMSPACE	2019-01-07 02:41:26

Figure 26: *Printscreen* of the null prices from the D-cycloserine metabolite (id_bioeco = 9198694939260524135) in the bioeconomics platform. As observed in the amount column, all of these prices have null amounts or are rounded to zero. This information was retrieved from https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/9198694939260524135.

As a result, one of the first preprocessing steps performed on the **df** dataframe was the filtering of all the null amounts. For this reason, there was a loss of 295 prices from the previous 836888 prices, making this dataframe contain now 836593 prices [6].

The next step in the preprocessing of these data is the elimination of duplicate prices. As mentioned above, the duplication problem is seen in both the bioanalysis_metabolite and the bioanalysis_metabolite_price table. Therefore, since this dataframe has information from these two tables, a solution to resolve both these duplications is essential before any performed analysis. First, the duplicated prices were removed from the **df** dataframe. In this phase, a new one was created, named **df_clean**, that originated from the utilization of the **drop_duplicates()** command, from the software library *pandas*, on the former dataframe. With this command, all rows are compared to each other, and if there are some that match in the values of all the columns, the command keeps one in the dataframe, and the others

6 Another thing to bear in mind is that this dataframe still has null values. However, since they are from the metab_name column and, hence, do not affect the calculations done in this work, the elimination of these rows is not reasonable. Hence, the **df** dataframe still has 364 null values in the metab_name column, but the other null values were eliminated.

are eliminated (Documentation, a). Since the index is a crucial part to identify the duplicates and this command only examines the columns, the alteration of the index to a column, and then back together to the index was necessary to conclude this step. Thus, the new dataframe, **df_clean**, had subsequently 825408 prices, which was 11185 prices less than the prior 836593 prices in the **df** dataframe.

Admittedly, the elimination of duplicate metabolites is still essential. In this case, the information necessary to discover the duplicates is not entirely in the generated dataframe. Thus a new one, possessing the data retrieved from the **bioanalysis_metabolite** table, was created. This dataframe had 4304 metabolites that, compared to the numbers identified in section 4.3, corroborates to the total number of metabolites, including the duplicates, presented in bioanalysis. Since the goal of this work is to analyse the df_clean prices, the duplicated metabolites were not deleted from this new dataframe. On the other hand, a dictionary with the information of the duplicates was conceived, so that it would be implemented later in the **df_clean** dataframe. In order to create the dictionary, the metabolites dataframe underwent the **duplicated()** command from *pandas*, so that all the duplicate rows could be returned (Documentation, b). Consequently, the keys of the dictionary comprised the names of the duplicate metabolites, whereas the values included a list of all the possible corresponding ID's of each key. Then, if an existent metabolite in the dictionary was found in the df_clean dataframe, the prices related to that metabolite would be gathered. Next, the values would be examined to make sure that all of them were associated with the same metabolite ID, and if not, they were saved with the ID that appeared first in the database (the ID with the lowest number in the list). That way, the prices from the duplicated metabolites would be saved as being from the same metabolite, attaining a solution for this issue.

Finally, with these changes, the **df_clean** dataframe had 825408 prices, 2278 metabolites, 126 providers, 3 sources and 3 units. Take into account that the 3 units are g, mg and kg, confirming that the prices included in this dataframe are associated with only mass units. Nevertheless, bear in mind that none of the following case studies were duplicate metabolites in the bioanalysis database.

In addition, another way to rectify these duplications in the future is to retrieve the metabolite and price data from bioeconomics platform, without the ID established by this platform.

Oscillations - Problem Solving

Different prices in each month and provider make price analysis difficult for each metabolite. Admittedly, the visualization of the price variation in a time series can not be accomplished, because there are considerable discrepancies in a single month that cause vertical lines and oscillations in the variation (as seen in the time series from section 5.1.1). As mentioned

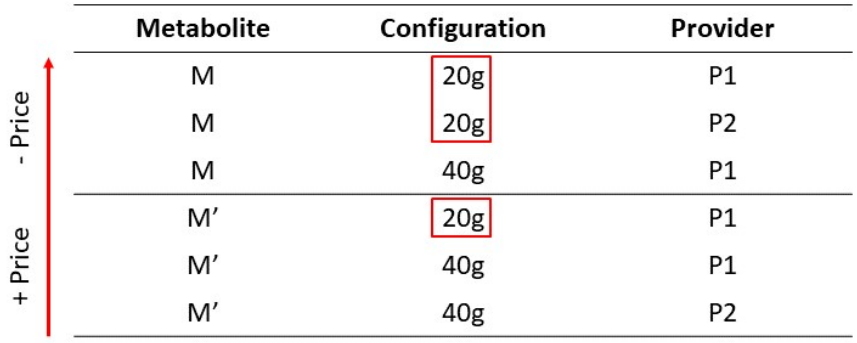
above, this issue occurs due to two factors: different configurations (amount + unit) and the integration problem.

In the first factor, the different price values happen due to the conversion done in the prior preprocessing methods. First, to understand this complication, the insight about the bioeconomics platform development, discussed in section 3.2, is vital. As already mentioned, the prices of each metabolite are updated every month (Figure 5). Consequently, what should be expected in the data retrieved from the bioeconomics platform is the presence of only one price per provider and configuration (amount + unit) in each month. However, to handle all the prices, these were converted to the same configuration, 1 gram, and the respective time series were done using these converted prices. As a consequence, this decision contributed to the presence of more than one price per month for each provider. Besides, the conversion of these values arranged them in very different ranges, creating a disparity between prices in the same metabolite, for the same provider and month. This is coherent with the law of supply and demand by Gale (1955), which confirms that the metabolite price tends to get lower as the amounts get higher. Which in its turn, contributes to bigger price intervals when analysing various configurations.

Considering the second factor, the integration problem, as described in the section above, this leads to distinct prices because the values related to a specific metabolite might belong to a different one. Because the prices in the database do not have the information about the external source ID handled in their retrieval, the distinction between all the prices that should be kept and the ones that should be eliminated cannot be executed without manually checking out the website of the platform or source. This manual check would be extremely time-consuming, so another solution is necessary. Also, due to the fact that this problem might originate from more than one step realized before the combination of the metabolites and prices in the bioeconomics platform, a solution handling solely the bioanalysis database, which does not possess the information of the price retrieval, is impossible. However, a tendency was observed when analysing this problem, that aid in dealing with this issue. As further explained in chapter 5.1, in a specific price variation for an exact set of properties, the minimum prices of the providers that have extensive discrepancies, are the ones that relate the most with the price range of the metabolite in other sets. Hence, a new assertion is taken into account when analysing these problems: the minimum price of a distinct month, provider, configuration and metabolite is the legitimate price for the properties analysed in this exact metabolite.

With both these issues in hand, a method to maneuver them was established, where, as displayed in figure 27, the analysed dataframe is filtered twice. First, a specific configuration was chosen, resulting in a price analysis between the same amount and unit property (analysis will include just one configuration). This solves some of the different price values present in the same month. Nevertheless, some distinct values still exist in some providers

when analysing each month, due to the integration problem. Therefore, another filter is applied, choosing the minimum prices, because of the tendency explained above. The method used in this step was to sort the dataframe by the price value and then, keep the first value of the same provider, unit, amount, source, year, and month, while eliminating the others.



Metabolite	Configuration	Provider
M	20g	P1
M	20g	P2
M	40g	P1
M'	20g	P1
M'	40g	P1
M'	40g	P2

Figure 27: Representation of the preprocessing method for the oscillation problem. As displayed, the prices can be related to different configurations and metabolites, due to the integration problem. To solve this situation, the first filter is implemented, this being represented by the red boxes. In this example the configuration chosen was $20g$, however, they still belong to different metabolites. It is in this phase that the second filter is applied. This last filter takes into consideration the theoretical assumption that the minimum price values belong to the correct metabolite. In addition to this explanation, it is also shown that each minimum price of a configuration will be categorized into different time series depending on the provider.

After all these filters, the existence of distinct values in one month is solved. With this solution, only one price remains per month when analysing a metabolite, and the price variation for each provider can now be visualized, without any oscillations (this will be displayed in section 5.1.1).

Bear in mind, that this method can only be implemented when the analysis of a specific metabolite is being performed. Therefore, the **df_clean** dataframe will not change.

4.3.3 *Exploratory Statistics*

After these preprocessing steps, the exploratory analysis of the data was addressed. First, one interesting aspect to understand about these data is the number of prices that exists for each metabolite. With that in mind, a bar plot to observe this analysis was generated.

However, since there is a myriad of metabolites, that plot with those metabolites in the x-axis, was not understandable. For that reason, a histogram representing the number of metabolites per number of prices was developed (Figure 28). While observing the figure 28, the first thing to notice is that the histogram is right-skewed. In other words, this implies that the mean of the number of prices is typically higher than the median. Hence, and with additional calculation, it was concluded that 98% of the metabolites in this dataframe (2234 of 2278 metabolites) have each less than 2000 prices. This information is vital to choose a suitable metabolite as a case study in later analysis, and also, to understand the typical metabolite that is present in the database. Considering the standard occurrence in the bioanalysis, some adjustments can be made to the calculations, so that they better correspond to the reality.

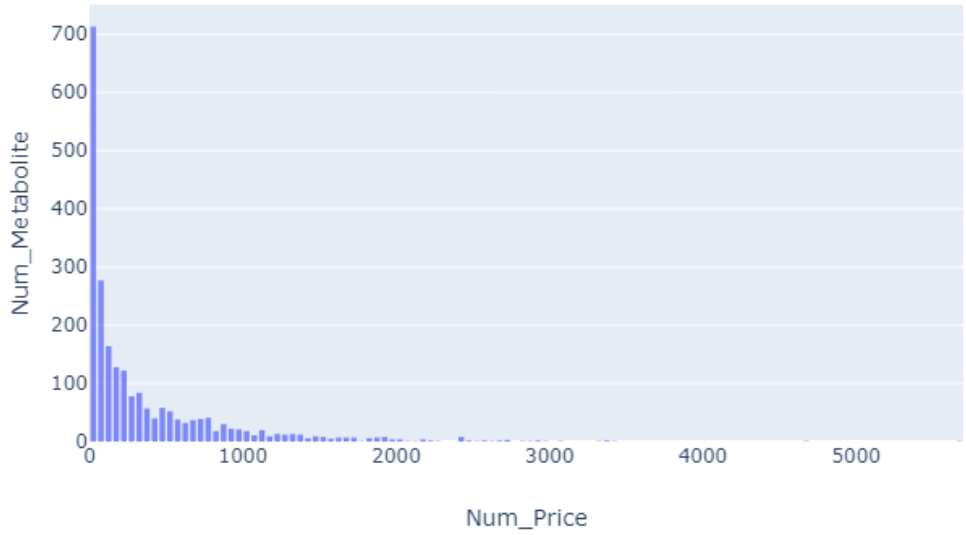


Figure 28: Histogram of the number of metabolites per number of prices in the df_clean dataframe. As observed in this plot, the majority of the metabolites in this dataframe have less than 1000 prices. Also, it is confirmed that this histogram is right skewed since the mean of the number of prices is typically higher than the median.

Moreover, an interpretation of the data confirmed that only 22.2% (506 of 2278 metabolites) of metabolites from the **df_clean** dataframe had prices for all 18 months of the time window. As a consequence, since this percentage is meager, the percentage of metabolites with prices for 90% of the months was calculated, concluding that 59.7% (1361 of 2278 metabolites) of metabolites complied with this condition [7].

In regards to the providers, this calculation was also performed, with the result indicating how many of them have prices for all 18 months, independently of the metabolite associated.

Only 53.2% providers (67 of 126 providers) have prices for all months, and 57.1% providers (72 of 126 providers) have prices for at least 90% of the months [7].

This might be helpful for the price analysis, so that the providers and metabolites which do not have prices for at least 90% of the months can be identified and filtered out, since these are not preferable in further analysis.

Another information noticed while examining the **df_clean** was the existence of providers with very identical names, possessing only slight changes. The reason for these changes is the presence of prices fetched by the bioeconomics platform from different sources. Thus, slight changes in the name occur depending on the source. Although, they will not be considerate the same since the values do not match in the same date for the same metabolite, amount and unit.

4.4 SUMMARY

To sum up, first a *Java* project was developed to create and populate a new database, *bioanalysis*, with the prices and metabolites contained in the bioeconomics platform. However, some issues were detected in the data, leading to a necessary solution for these problems. These included, missing and duplicate data, as well as, diverse problems established in specific steps of the bioeconomics related to the integration of the prices and metabolites (integration problem). Also, these price values had various units, that could not be converted to a unique one for all prices, complicating this work's goal.

Therefore, a first preprocessing method did not just solved the null values and eliminated the duplicate data, but also filtered the prices to a specific time window (January 2018 to June 2019) and a particular type of units (mass units). Whereas a second method dealt with the existence of more than one price per month and provider while analysing a specific metabolite, which originated oscillations in the price variations. Take into account that there are no duplicates in the dataframe, therefore these prices are considered different. In this case, there are two problems:

- The conversion of the prices to 1 gram is generating very distinct values per month and creating a more notable dispersion.
- The bioeconomics platform is gathering price values for the same provider, source, unit and amount, in the same month (in a matter of fact, on the same day too) - integration problem.

7 Take into account that, the results for the condition of having prices for at least 90% of the months in this time window, also includes the metabolites/providers that comply with the condition of having prices for all months.

Thus, these issues were managed by analysing a particular metabolite with prices for just one configuration, and in the months that still had more than one value present (even after choosing a specific configuration) by filtering these prices to the minimum values.

In the next chapter, the case studies will be done in the **df_clean** dataframe, which has 2278 metabolites, 126 providers, 3 sources and 825408 prices (Figure 29). All the prices will have mass units and will belong to the time window already mentioned, from January 2018 to June 2019. As a result, **df_clean** will be used from now on as the original dataframe from where the case studies data are retrieved.

Also, the bioanalysis database was not modified, maintaining all the prices, providers, units, metabolites and sources, if needed in the future, even the duplicated objects.

date	metab_id	metab_name	price	amount	unit	provider_name	source	price_per_unit	price_per_g	year	month
2018-01-11 21:39:13	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	1
2018-02-11 23:46:53	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	2
2018-03-11 23:47:07	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	3
2018-04-11 22:47:14	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	4
2018-05-11 22:47:27	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	5
2018-06-12 01:20:49	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	6
2018-07-12 02:29:45	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	7
2018-08-23 01:54:41	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	8
2018-09-23 01:54:58	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	9
2018-10-23 01:55:19	1	tobramycin	158.0	1.0	g	ACC Corporation	CHEMSPACE	158.00	158.00	2018	10

Figure 29: *Printscreen* of the output from the final version of the df_clean dataframe in *Jupyter Notebook*. The dataframe displayed in this figure contains data without duplicates and null values and includes prices with converted units (price_per_g column). Only the first ten lines of the dataframe in question are presented.

5

CASE STUDIES: 4-AMINOPYRIDINE AND METHANE

The main focus in this thesis will be the price analysis, that is, all the evolution of the respective prices in a certain period, the way providers differ in the disclosed prices, among other patterns that may be useful. For this analysis, the discovery of a metabolite that will help establish a conclusion for the other metabolites as well is crucial. Therefore, a few metabolites were chosen to participate in this analysis, the case studies. This method aids in the reduction of the considerable amount of data that is present in this dissertation and can also be a useful approach to predict the same price patterns in other metabolites, creating a way to generalize the data.

First and foremost, two metabolites from the 2278 metabolites contained in the df_clean dataframe were selected to become the **case studies** of this dissertation. Since the purpose of these metabolites is to serve as a foundation for all the analysis and algorithms designed, expecting that these algorithms can be generalized afterwards and become able to be implemented in the remaining compounds, the goal was to look for metabolites that had plenty of data so they could contain multiple information and exceptions to relate to other metabolites.

As observed in figure 30, which is an adaptation of the bar plot representing all the metabolites from the df_clean dataframe, the metabolites 187 and 1175 are the ones that have the most prices associated. However, for the first case study performed in this dissertation, the metabolite chosen was not the one with the highest number of prices, but the second higher - the 4-aminopyridine metabolite - with the ID 1175. It is with this compound that the algorithms for the preprocessing methods (noted in section 4.3.2), the price analysis and the identification and removal of outliers, are initially conceived.

5.1 4-AMINOPYRIDINE METABOLITE

The reason for not selecting, as first case study, the metabolite with the highest number of prices was because the 4-aminopyridine has, not only numerous prices, but also a higher number of providers (Table 15). As determined in table 15, the metabolite 187 (methane) has 5662 prices in comparison to 4-aminopyridine, which has 4652, almost minus 18% of

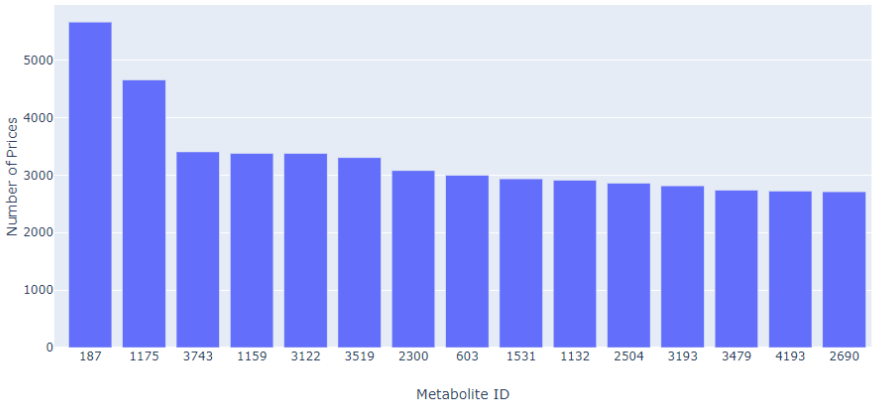


Figure 30: Bar plot of the number of prices encountered in each metabolite of the df_clean dataframe. Note that the bar plot was zoomed in the first fifteen metabolites with higher number of prices to simplify the observation. The original graph would have 2278 metabolites in the x-axis. As displayed in this figure the metabolite 187 is the one with the higher count of prices associated (5662 prices).

methane's total prices. On the other hand, methane has only 39 providers, compared to the 67 providers of 4-aminopyridine, which is 42% less.

Table 15: Output from a *SQL* query obtaining the number of prices, providers and sources of the metabolite 187 and 1175, the two metabolites with the higher number of prices in df_clean dataframe. As it is possible to notice, methane has more prices than 4-aminopyridine, but has less providers associated.

Metabolite ID	Common Name	Number of Prices	Number of Providers	Number of Sources
187	methane	5662	39	3
1175	4-aminopyridine	4652	67	3

Considering that one of this work's goals is to understand how the prices in each metabolite vary, and since the providers have a massive impact on these values, the metabolite 1175 seems to be the best choice to capture a significant part of the initial information about the price variations of the metabolites. Also, this is the information required so that the code for the analysis can be developed from this metabolite. Besides, the percentage of lacking providers in comparison to the percentage of lacking prices is much higher, leading to a preferable choice of the metabolite with a more significant number of providers, 4-aminopyridine.

As a result of this decision, the df_clean dataframe was filtered to contain only the prices related to the metabolite 1175, the 4-aminopyridine. The resulting dataframe is named **df_caseStudy** (Figure 31). This dataframe had 4652 prices, 67 providers, 3 sources (the same as in the previous dataframe) and the 3 mass units. Take into consideration that,

df_caseStudy originated from df_clean, there are no duplicated prices and metabolites and, also, no null values, in exception to some elements in the metab_name column.

date	metab_id	metab_name	price	amount	unit	provider_name	source	price_per_unit	price_per_g	year	month
2018-01-10 11:35:15	1175	4-aminopyridine	55.0	1.0	mg	ENAMINE Ltd.	MolPort	55.000000	55000.000000	2018	1
2018-01-10 11:35:15	1175	4-aminopyridine	56.0	2.0	mg	ENAMINE Ltd.	MolPort	28.000000	28000.000000	2018	1
2018-01-10 11:35:15	1175	4-aminopyridine	59.0	5.0	mg	ENAMINE Ltd.	MolPort	11.800000	11800.000000	2018	1
2018-01-10 11:35:15	1175	4-aminopyridine	78.0	10.0	mg	ENAMINE Ltd.	MolPort	7.800000	7800.000000	2018	1
2018-01-10 11:35:15	1175	4-aminopyridine	85.0	15.0	mg	ENAMINE Ltd.	MolPort	5.666667	5666.666667	2018	1

Figure 31: *Printscreen* of the df_caseStudy dataframe in *Jupyter Notebook*. As observed, this dataframe only contains the prices related to metabolite 1175 (4-aminopyridine). To simplify, this figure displays solely the first five rows of the dataframe in question.

5.1.1 *Oscillations Problem Solving*

Initially, to analyse the prices of the metabolite 1175, a time series plot was constructed, considering this is one of the main steps while managing similar data (Chatfield, 2003). The first step to create this time series is to ensure that all prices can be compared. Even though the prices are all associated to the same metabolite, they are still related to different amounts and units. Hence, the values from the **price_per_g** column, present in the df_caseStudy dataframe, were the ones employed to plot this time series. This column also exists in the df_clean dataframe, as displayed in figure 29.

Consequently, this conversion became one of the reasons for the oscillations problem. Due to this, the preprocessing method explained in section 4.3.2 will have to be applied to these prices, in addition to the other preprocessing methods performed before for the duplicate and null values.

Moreover, the goal in this time plot is to observe the variation of the 4-aminopyridine's prices throughout the time. However, with interest in comparing other impactful factors, the addition of the provider to this plot was considered. Since the providers are a factor with considerable impact on the prices, the data was grouped by each provider, so the comparison between the variations could be perceived regarding this aspect.

In figure 32, the time series is displayed. Please take into account that the label at the right of the plot is not complete with all the providers since they are 67. Because of the considerable amount of providers, to simplify the plot visualization, the rest of them were cut from the figure. Here, the problem mentioned above is undoubtedly identified. The first thing to mention in this plot is the range of the price values - these vary between 0.28 USD/g and 66880 USD/g. On the other hand, another concern is the fact that the various series observed in the plot are not what would be expected (comparing to figure 1). Even if

the range is vast, the prices for each provider seem to significantly vary within this range, displaying an abrupt variation - referred to, in this dissertation, as oscillations.

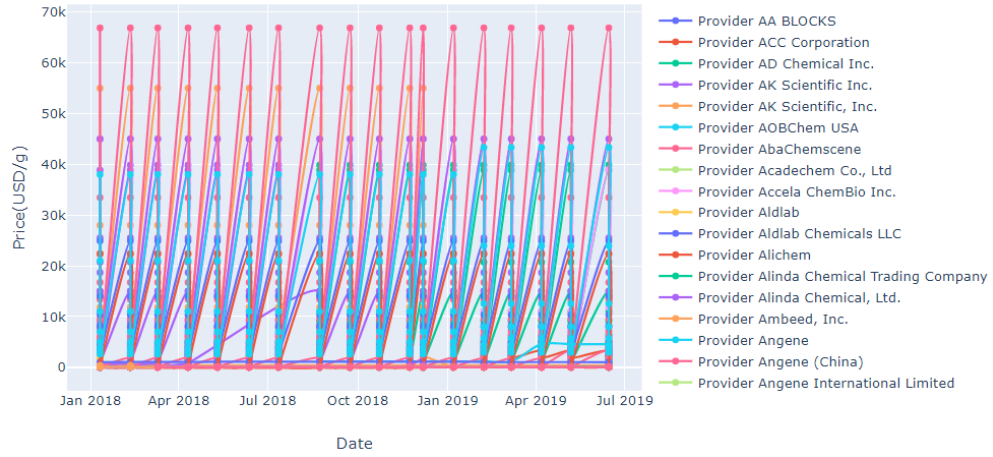


Figure 32: Time series plot of the available prices for the metabolite 1175 in the df_caseStudy dataframe. In this plot, each time series corresponds to the price evolution of a specific provider. As displayed, these series contain several price oscillations, varying within a wide value range (0.28 to 66880 USD/g). Also, there are in total 67 time series, each corresponding to a provider, however, to simplify, the subtitle only contains 18 providers.

Since figure 32 has 67 time series to look all at once, a single provider was chosen to better demonstrate the oscillation problem. A great example of this concern is the series related to the provider *ChemShuttle* (Figure 33). Observing this time series plot, the critical variations are also noticeable along the whole time window, having prices that vary from 1.6 USD/g to 9 USD/g. Bear in mind that each bullet noticed in the time series correlates to a price value from the df_caseStudy dataframe for that date and provider. This is an essential note once it means that, as perceived in figure 33, in approximate dates, this provider as three very distinct price values, that are causing these oscillations and vertical lines in the price variation.

Also, another situation to consider in these is that most of these price values are constant throughout time. When analysing the oscillations without taking into account the real values behind them, this lack of variation is not initially observed. However, the reason for this state of changelessness is critical to help the analysis of the metabolite's prices. This also confirms the oscillation problem mentioned in section 4.3.2. As visualized in this figure (33), there is more than one price per month.

The main problem with this issue is that the real variation can not be examined and therefore the price analysis can not be effectively accomplished. This matter can be noticed in most of the 67 providers. Ergo, these issues are of extreme importance, and its answer is necessary to promote better data for the analysis.

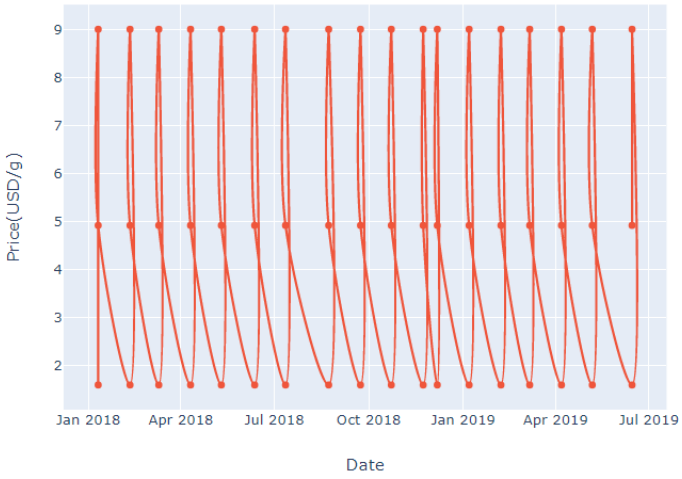


Figure 33: Time series plot of the available prices for the provider *ChemShuttle* in the metabolite 1175. In this plot, it is possible to observe the variation of the 4-aminopyridine's prices for the provider in question (between 1.6 and 9 USD/g). Furthermore, each bullet displayed in the time series corresponds to an actual value in the df_caseStudy dataframe.

With this problem in mind, to analyse the various prices that are contained in each month, the data was separated by month in the following analysis. So, to examine the data and its distribution in distinct groups, another approach was applied - the creation of boxplots. As referred above, in section 2.3, the boxplots promote the visualization and comparison of data distributed in various groups. Hence, the usage of this type of plot creates a high setting to observe how the price data of each provider is distributed in time, namely in each month of the time window, explaining this oscillation problem. Furthermore, these boxplots also help to detect possible outliers.

As a result, the next step was to create a graph for the prices grouped by month and provider, visualizing their variations through the numerous boxplots. Furthermore, to understand the variation displayed in figure 32, a plot of all prices, regardless of provider, was also developed (Figure 34).

First, in figure 34, the same range of prices as the one from the first time series (32) is presented (0.28 USD/g to 66880 USD/g). In addition, this shows that the prices distribution's main body resides below 10000 USD/g, and consequently exists a significant number of outliers. However, since the range is too broad to conclude anything, the best approach to follow is the analysis of different price properties, as the unit and provider.

Admittedly, when examining the *ChemShuttle* boxplots, the existence of more than one price per month in the same provider is corroborated (Figure 35). The prices are displayed in the bullets alongside the boxplots [1], matching the values in the time series from figure 33. As

1 Take into account that in this plot, the bullets refer to each value used to create the boxplot, and not the outliers as above. That way, the real price values, and not just their distribution, can be visualized.

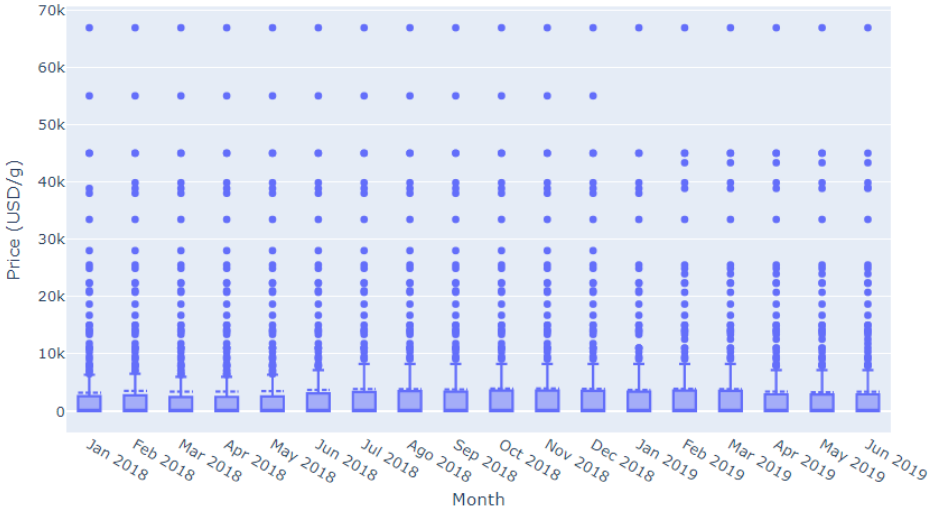


Figure 34: Boxplot of the available prices for the metabolite 1175 per month. In this figure, the 4-aminopyridine's price distributions are displayed per month, regardless of the provider associated. As observed, these distributions are comprised between 0.28 and 66880 USD/g. The bullets represented in this plot refer to the outlier values.

predicted, the observation of the prices' distribution for this provider proved that they were constant. Throughout the months of the time window, the three distinct values do not vary, being repeated over and over again in the entire time window (1.6, 4.92 and 9 USD/g). This implies that even after solving this problem, no price variation will be portrayed, which is not ideal for this type of analysis since there is a need for variation so that some information about the metabolite, the respective providers and their price evolution, can be taken out (as explained above in section 2.3).

In the end, with a more thorough analysis of the distributions, the observations performed in each provider concluded that the presence of not varying prices was observed in 34 of 67 providers, which means 50.7% of the providers have no price variation throughout the time.

Nevertheless, other providers display price variations throughout the months, comprising almost 50% of the total number of providers in the df_caseStudy dataframe (49.3% - 33 of 67 providers). To demonstrate this situation, figure 36 illustrates the example of provider ENAMINE Ltd. Nevertheless, in this provider, three distinct phases of variation can be easily observed, wherein each of them the prices are constant. These phases could signify a trend in this provider's prices. However, since the data gathered in this work still comprises a short period of time, this theory cannot be verified. As noted in section 4.3.2, these distinct constant prices arise from the prior conversion and the integration problem.

Another reason that was first thought for some of the discrepancies between these prices were the sources, since they could have gathered different values for the prices (Figure 6). Nonetheless, in the *ChemShuttle* provider when noticing the number of unique sources these

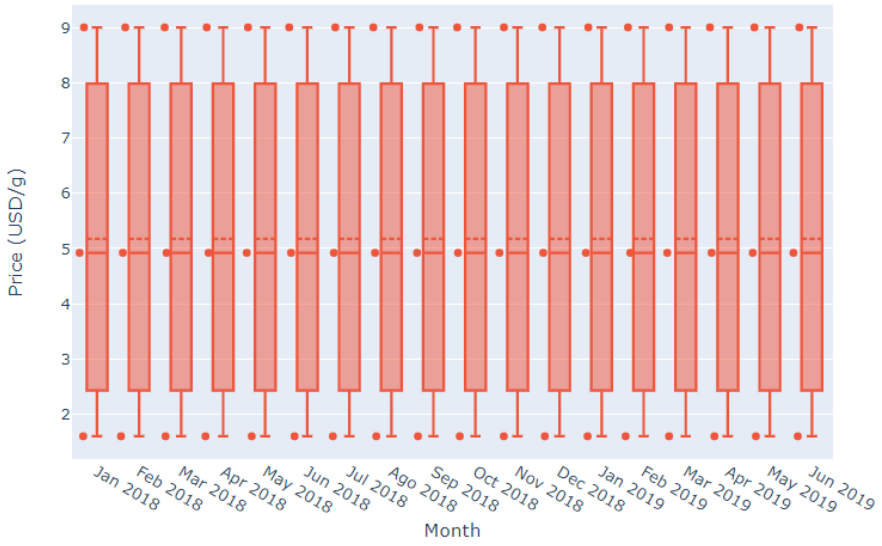


Figure 35: Boxplot of the available prices for the *ChemShuttle* provider in the metabolite 1175 per month. In this figure, the 4-aminopyridine's price distributions are displayed per month, regarding the provider associated. The bullets presented in the plot are marking the real price values available in the dataframe in question. Hence, three distinct prices can be observed (1.6, 4.92 and 9 USD/g). Also, in this plot, no outliers are presented.

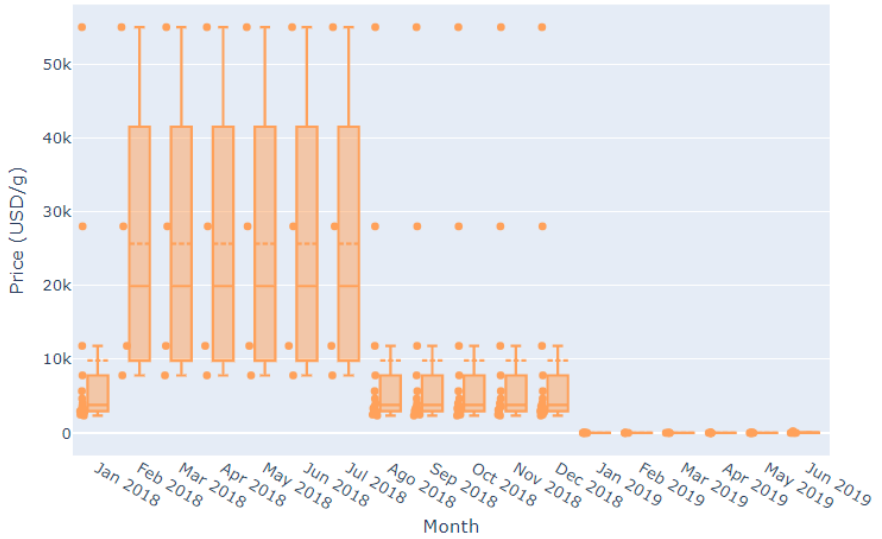


Figure 36: Boxplot of the available prices for the *ENAMINE Ltd.* provider in the metabolite 1175 per month. In this figure, the 4-aminopyridine's price distributions are displayed per month, regarding the provider associated. In this boxplot, it is not displayed any outliers, meaning that the bullets presented in the plot are marking the real price values available in the daaframe in question.

prices had associated, only one source existed, the CHEMSPACE. So, the justification for the presence of some variation between the prices of a specific month, could not be the source. Subsequently, the prices related to this provider were analysed, verifying the oscillation problem. Figure 37 presents the first three prices encountered in the dataframe.

date	metab_id	metab_name	price	amount	unit	provider_name	source	price_per_unit	price_per_g	year	month
2018-01-10 11:35:20	1175	4-aminopyridine	40.0	25.0	g	ChemShuttle	CHEMSPACE	1.60	1.60	2018	1
2018-01-10 11:35:22	1175	4-aminopyridine	45.0	5.0	g	ChemShuttle	CHEMSPACE	9.00	9.00	2018	1
2018-01-10 11:35:22	1175	4-aminopyridine	123.0	25.0	g	ChemShuttle	CHEMSPACE	4.92	4.92	2018	1

Figure 37: *Printscreen* of the df_caseStudy dataframe in *Jupyter Notebook* filtered by the *ChemShuttle* provider. As observed, this dataframe only contains the prices related to metabolite 1175 (4-aminopyridine) and the *ChemShuttle* provider. To simplify, this figure displays solely the first three rows of the dataframe in question, that are associated to January 2018.

As this figure shows, the reason why the difference between the prices 1.60 and 9.00 USD/g exists is because of the conversion among the different amounts (25 and 5g). Considering the difference between the prices 1.60 and 4.92 USD/g, they have the same provider, source, unit, amount and date, where it should only exist one price with the same properties per month. These displays the integration problem, mentioned above, where both are retrieved from different external source ID's and the higher price belongs to another metabolite (Figure 22, section 4.3.1). The two situations can be seen in the rest of the time window in the *ChemShuttle* provider, and in other providers with the same looking time series and boxplots. For example, in the ACC Corporation provider, the distinction between the prices within the same month, happens due to the price conversion.

Taking into account the second preprocessing method, explained in section 4.3.2, the data from this metabolite was filtered following the figure 27. First, a specific configuration was chosen so that the prices analysed would only belong to one amount and unit, solving the conversion issue that leads to more than one price per month. Therefore, the configuration applied in this analysis was chosen, taking into account the number of prices that were related to it. Hence, after calculating the number of prices in each configuration, the one that had the most prices in this dataframe (df_caseStudy) for this metabolite was the '25g' (542 prices). Consequently, a new dataframe, including only the prices from the df_caseStudy related to that configuration (25g), was generated, named **df_caseStudy_25g**.

A new time series plot was achieved, where all the contained series represent the price evolution of a particular provider, in which only prices for 25g are included (Figure 38). Also, this means that the number of time series in this plot is lower, compared to the plot in figure 32, since not all the existent providers have prices for 25g. For instance, the *ENAMINE Ltd.*, from figure 36, is not included in this dataframe because it did not contain any price for the 25g configuration.

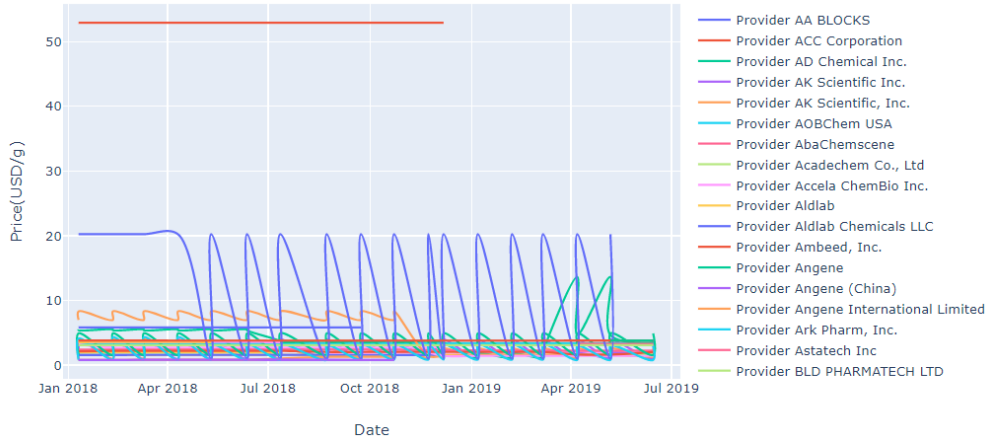


Figure 38: Time series plot of the available prices for the metabolite 1175 in the df_caseStudy_25g dataframe. The prices are now filtered by the 25g configuration. In this plot, each time series corresponds to the price evolution of a specific provider. As displayed, these series contain some price oscillations, but the majority became constant, varying between 0.8 to 52.9 USD/g.

Foremost, observing this figure (38), most of the time series are now straight throughout time. This denotes that several prices remain constant in this time window, confirming what was mentioned above. Moreover, the price range shown in this plot is much narrower (between 0.8 and 52.9 USD/g) while compared to the interval shown in figure 32 (between 0.28 and 66880 USD/g). Nonetheless, the figure 38 still displays unresolved oscillations in this plot. As previously referred, these occur due to the mixed external source ID's error. The *ChemShuttle* and the *AK Scientific Inc.* providers are two examples of this situation.

Therefore, following the first filter, a second one was applied in the months that still present more than one price, the minimum filter. For the most part, in the *ChemShuttle* example, the higher prices equate to the ones of the 3-(4-bromophenyl)propanoic acid compound, while the lower prices equate to the real prices of the 4-aminopyridine compound (Section 4.3.1). Another example to show this issue is the *AK Scientific Inc.* provider. As seen in the plot of figure 39, the bullets next to the boxplots indicate all the price values in each month. Consequently, there are three distinct prices in each month that are constant throughout the time window.

When analysing the bioeconomics platform, the fact that the higher prices belong to the wrong compound is confirmed (Figure 40). As observed in figure 40, the two higher prices (80 and 100 USD, respectively 3.2 and 4 USD/g) came from the ChemSpaceID correspondent to the 3-(4-bromophenyl)propanoic acid (CSC000217147), while the lower price (25 USD or 1 USD/g) corresponds to the 4-aminopyridine metabolite with the ChemSpaceID CSC000211974.

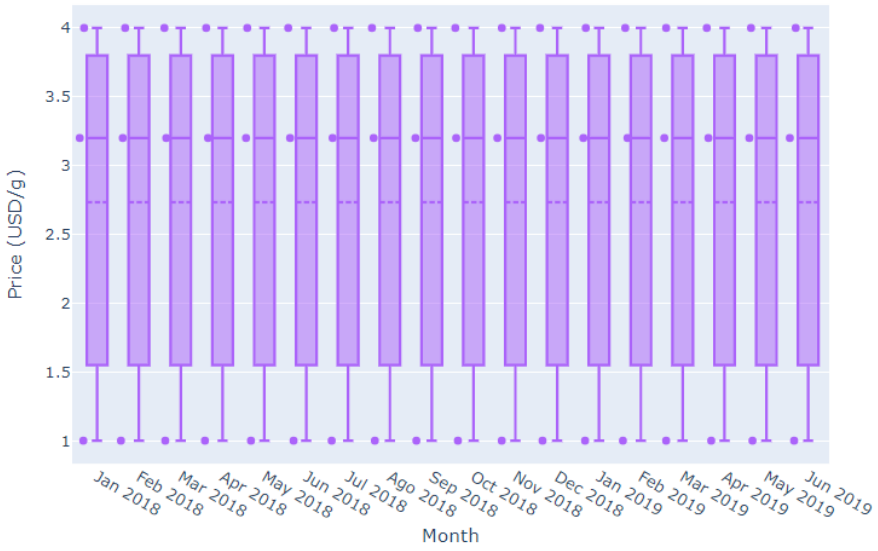


Figure 39: Boxplot of the available prices for the *AK Scientific Inc.* provider in the 25*g* of metabolite 1175 per month. In this figure, the 4-aminopyridine's price distributions are displayed per month, regarding the provider and the configuration associated. In this boxplot, it is not displayed any outliers, meaning that the bullets presented in the plot are marking the real price values available in the daaframe in question (df_caseStudy_25g).

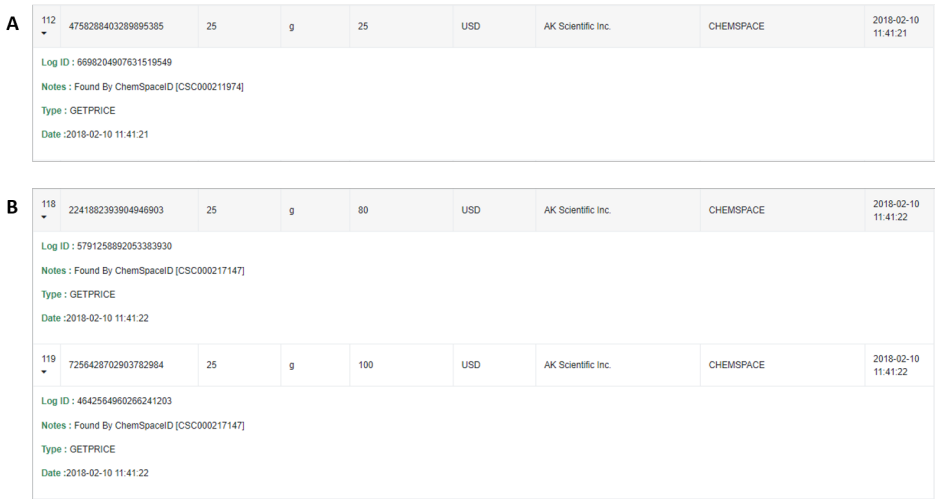


Figure 40: *Printscreen* of the prices for the *AK Scientific Inc.* provider in the 25*g* of metabolite 1175 (4-aminopyridine) in February 2018 from bioeconomics platform. This information was retrieved from https://mendel.bio.di.uminho.pt/bioeconomics/#/metabolite/8807103328882255903. (**A**) This value (25 USD) is associated to the ChemSpaceID CSC000211974. (**B**) Both these values (80 and 100 USD) are associated to the ChemSpaceID CSC000217147.

Both these examples, the *ChemShuttle* and the *AK Scientific Inc.*, demonstrate that clearly there is a tendency for the minimum prices to belong to the metabolite in question. In addition, the minimum ranges of the oscillations presented in figure 38, reside in the main price interval of the other providers.

Subsequently, for this approach, a new dataframe was created, the **df_caseStudy_25g_min**. As a result, this data frame only contained prices for 25g of 4-aminopyridine and the minimum values for each provider in each month. The outcome is observed in a new time series plot in figure 41, where the price range is situated between 0.8 and 52.9 USD/g, still considered wide. As expected, after solving the two problems described above, the price graph for this compound is now clearly displayed, with no fluctuations intruding the variation (Figure 41).

In figure 42, the providers above analysed, the *ChemShuttle* and *AK Scientific Inc.*, are displayed with both these filters, and to corroborate this preprocessing method, the two also include only one price per month and are constant throughout the time window, as expected after this approach.

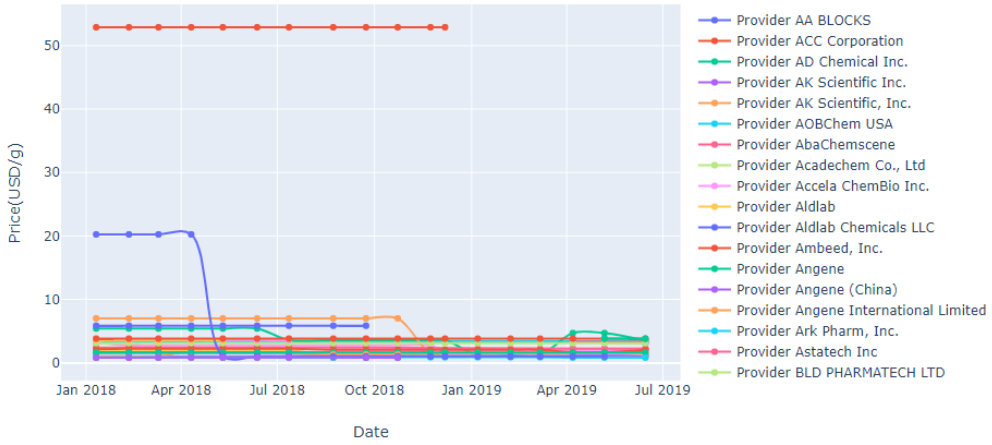


Figure 41: Time series plot of the minimum prices in each month and provider for 25g of 4-aminopyridine. Here, a time series corresponds to the price variation of each provider in the df_caseStudy_25g_min dataframe. Furthermore, the total number of time series is 34, however to better visualize the figure, the list of providers in the label was cut. Also, each bullet in these series represents an actual price value existent in that dataframe.

5.1.2 *Price Variation Analysis*

In the df_caseStudy_25g_min, the number of providers remained 34, since the only difference between this dataframe and the former was the elimination of the higher prices when there was more than one price in each month and provider. Hence, in this figure (41), 34 time series are represented, including some bullets that depict the actual price values. As illustrated,

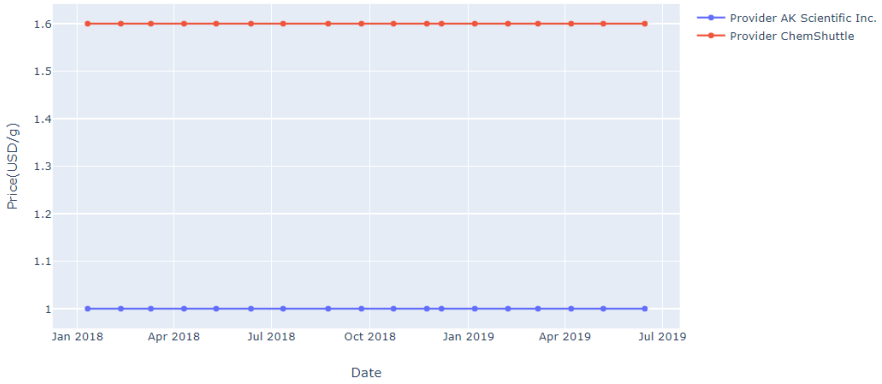


Figure 42: Time series plot of the minimum prices in each month for 25g of 4-aminopyridine in the *ChemShuttle* and *AK Scientific Inc.* providers. These time series correspond to the data from df_caseStudy_25g_min dataframe. Also, each bullet in these series represents an actual price value existent in that dataframe.

most of the time series are constant throughout time, which makes further predictions not relevant since they will have the same constant outcome. However, in some time series, a slight variation is displayed. This is the case, for example, of the *AK Scientific, Inc.*, where its price variation is displayed in figure 43. The range of this variation is kept between 1 and 2 USD/g. Nonetheless, a possible forecast analysis in this example would not be wise too, since the time window for these prices is rather short and no seasonal variation or trends are perceived yet. As a consequence, forecasting without a more considerable time window would be faulty (Montgomery et al., 2008).

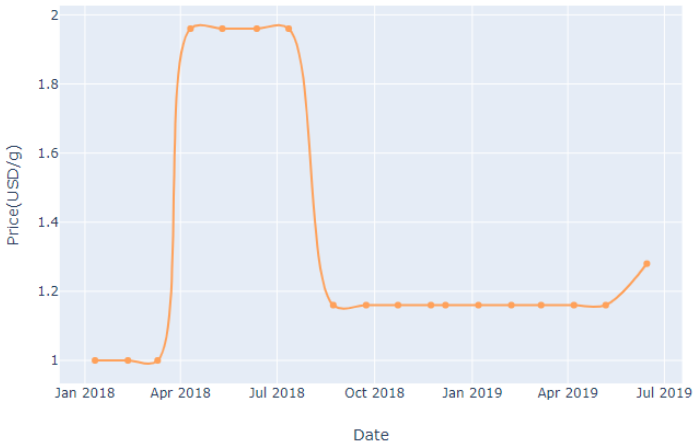


Figure 43: Time series plot of the minimum prices in each month for the *AK Scientific, Inc.* provider in the 25g of the metabolite 1175. In this plot, is observed a price variation between 1 and 2 USD/g. These prices belong to df_caseStudy_25g_min dataframe.

While examining df_caseStudy_25g_min dataframe and observing figure 41, is detected a minimum price value of 0.8 USD/g that belongs to both the *Enamine Ltd Chemspace* partner, the *FCH Group Chemspace* partner and the *UORSY Chemspace partner* providers. Hence, the price variation throughout time of these three providers is displayed in figure 44. As shown in this figure, the provider that has consistently the lowest price in the whole df_caseStudy_25g_min dataframe is the *Enamine Ltd Chemspace partner* provider, since all the respective prices of this provider are equal to the minimum value (0.8 USD/g). Nevertheless, none of these providers have prices for the year 2019, so *Fluorochem Limited* provider is also included as one of the cheapest since it has consistently the lowest price variation over a more extended time. Besides, the source of the cheaper prices in this dataframe is CHEMSPACE, since all the prices represented in figure 44 were gathered by this source, in exception to the prices from *Fluorochem Limited* that were collected from MolPort.

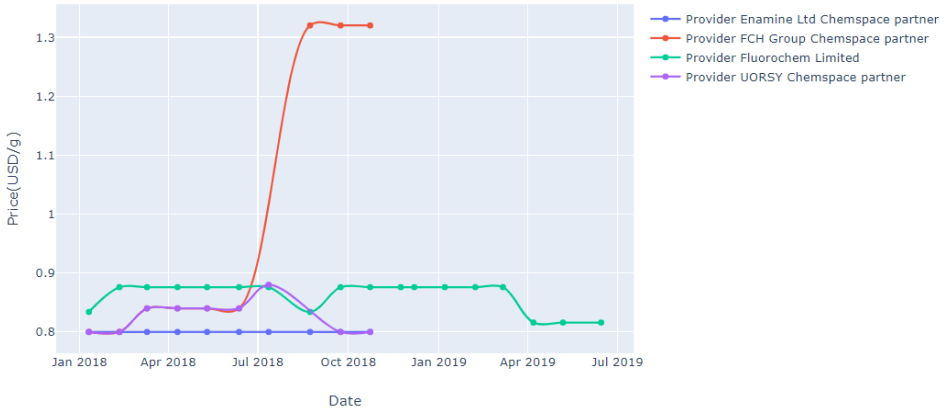


Figure 44: Time series plot of the lower price variations for 25g of the metabolite 1175 (4-aminopyridine). In this plot is represented the price variations of the three providers that include the minimum price value (0.8USD/g), *Enamine Ltd Chemspace partner*, *FCH Group Chemspace partner* and *UORSY Chemspace partner* providers. Also, another provider is displayed, the *Fluorochem Limited* provider, that contains the lowest price variation in a broader time interval. Each time series represents one of these providers. Furthermore, each bullet in the time series represents an actual price value existent in the df_caseStudy_25g_min dataframe.

In short, for the minimum values related to 25g of 4-aminopyridine, the provider with the lowest prices is the *Enamine Ltd Chemspace partner*, these being retrieved by the CHEMSPACE source.

Subsequently, additional analysis can be done with distinct configurations, so that possible correlations between the providers and sources can be discovered. The following price analysis executed considered the second configuration with the most prices, the 5g. To analyse these filtered prices, another dataframe was created, the **df_caseStudy_5g**, where it contains only the prices of the 4-aminopyridine metabolite for that amount and unit. After

applying all the steps, performed in the above configuration, to df_caseStudy_5g, solely the minimum price values were maintained in the months with more than one price.

In the end, the time series that was generated by these methods also showed a lot of constant prices, as compared with figure 41 (Figure 45). However, in comparison to the 25g configuration, this analysis includes more three providers, resulting in a plot of 37 time series. Also, as illustrated in figure (45), the range of these prices is between 1.8 and 656.76 USD/g, which is a lot wider than the respective result in 25g (between 0.8 and 52.9 USD/g). This confirms the law of supply mentioned in section 2.2, where the higher the amount sold, the less expensive it is.

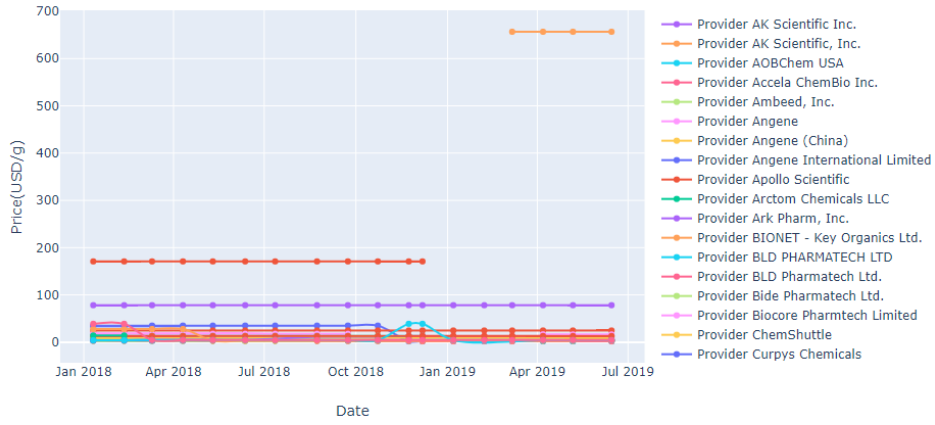


Figure 45: Time series plot of the minimum prices in each month and provider for 5g of the metabolite 1175 (4-aminopyridine). Each time series corresponds to the price variation of each provider in the df_caseStudy_25g_min dataframe. The list of providers in the label next to the plot was cut to better visualize the figure, the total number of time series is 37. Also, each bullet in the time series represents an actual price value existent in that dataframe.

Besides, this analysis confirmed that the three providers in this configuration with the lowest price value (1.8 USD/g) are *Angene (China)*, *Angene International Limited* and *BLD Pharmatech Ltd*. Nevertheless, these do not correspond to the three providers of the 25g configuration, mentioned above (Figure 46). On the other hand, there is a correlation between the *Fluorochem Limited* provider, where in both configurations these prices are consistently lower taking into account the whole time window.

This lack of correlation between the providers might occur due to the integration problem previously mentioned. To clarify, because there is a problem in the assimilation of the prices, this leads to the existence of mixed values from different metabolites. As a consequence, when comparing two configurations there is a possibility that the prices of one are related solely to the erroneous metabolite, and there is no real prices for that configuration in the metabolite in question. Since this issue cannot be solved without changing the algorithm of the bioeconomics, the preprocessment method applied in this work only sorted out the problem so the price variation could be analysed. Therefore, if hypothetically, the two

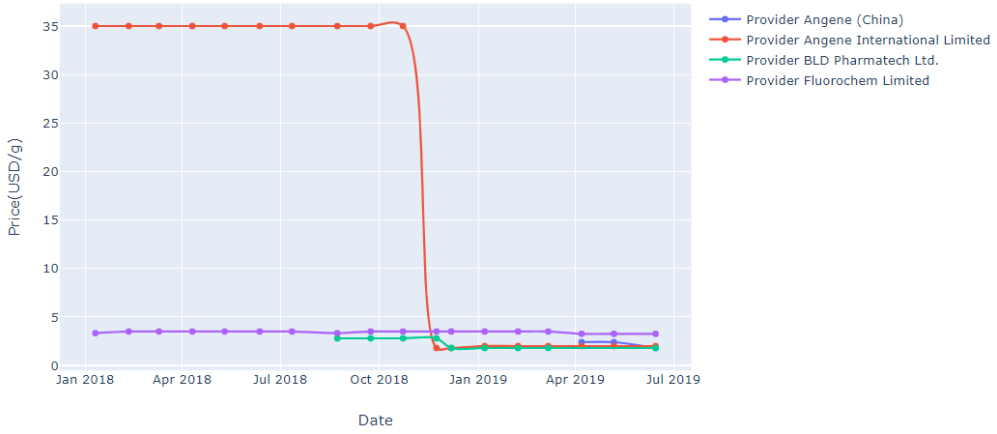


Figure 46: Time series plot of the price variations of the three providers that include the minimum price value (1.8USD/g) for 5g of the metabolite 1175 (4-aminopyridine). Each time series represents one of the three providers, the Angene(China), the Angene International Limited or the BLD Pharmatech Ltd. provider. Also, each bullet in the time series represents an actual price value existent in the df_caseStudy_5g_min dataframe.

compared configurations included different metabolite prices, the providers and sources of the lower prices will, likely, not correlate. Nevertheless, if only the provider identified in the two configurations is considered, this justification is also acceptable, since in this case, the minimum prices may belong to the correct metabolite.

5.1.3 *Outliers*

Even considering all these problems and lack of correlation, an algorithm for the outliers elimination was developed. In this step, the equations 1 and 2 from the section 2.3.3 were applied. After discovering the lower and upper whisker of the boxplot per each month, the price values that were encountered beyond each of those limits were considered outliers. Hence, in this algorithm, the providers that possess outliers were removed from the time series plot. Bear in mind that, in this case, the constant utilized to calculate these values was equal to 1.5, as presented in the equations. However, this constant can be changed in the future to different values in order to better fit this algorithms' goal.

Subsequently, when applying this algorithm to the values from 25g of 4-aminopyridine, four providers were detected with outliers in at least one month of the time window, the *SynQuest Laboratories, Inc.*, *Angene International Limited*', *ACC Corporation* and *Toronto Research Chemicals* providers. In contrast, when examining the prices from 5g of 4-aminopyridine, the six providers were identified as having outliers, the *Specs*, *ACC Corporation*, *Angene International Limited*, *UORSY Chemspace partner*, *Apollo Scientific* and *BIONET - Key Organics Ltd.* providers.

In figure 47 two plots are displayed representing the providers without outliers for the 25g configuration (**A**) and for the 5g configuration (**B**).

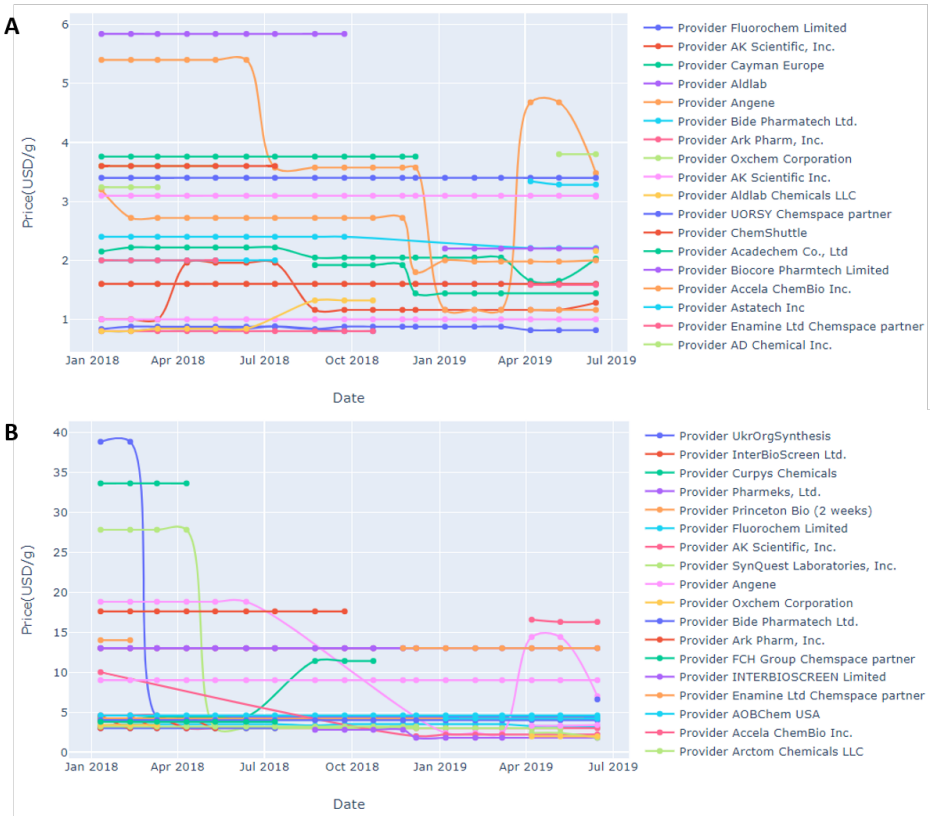


Figure 47: Time series plots of the price variations without outliers related to the 25g and 5g configurations in 4-aminopyridine. (**A**) These prices belong to df_caseStudy_25g_min dataframe. And there is 30 time series, each to a provider. (**B**) On the other hand, these prices belong to df_caseStudy_5g_min dataframe. And there is 31 time series, each to a provider.

As noticed in this figure 47, the price ranges in both these configurations became narrower. In the first configuration 25g, before removing the outliers the interval was between 0.8 and 52.9 USD/g, and now is between 0.8 and 5.84 USD/g. When considering the values related to 5g, before the elimination of the outliers the price range was among 1.8 and 656.76 USD/g, and after is between 1.8 and 38.8 USD/g, which is much more narrower than previously.

To sum up, the developed algorithm helps to better manage the data, discarding potential issues. Nevertheless, it can be improved in the future not only changing and testing with the constant to a better adjust, as mentioned above, but also, with the elimination of the providers. Over time, changes can be made to the algorithm, so that only providers with outliers in a given number of months are effectively eliminated from the analysis.

5.2 METHANE METABOLITE

In order to compare various situations and to understand if the analysis applied above can be utilized comprehensively throughout other metabolites, the same approaches were made in a second metabolite, methane (ID = 187). As referred to in section 5.1, this metabolite is the one with the highest number of prices in all df_clean dataframe (Figure 48).

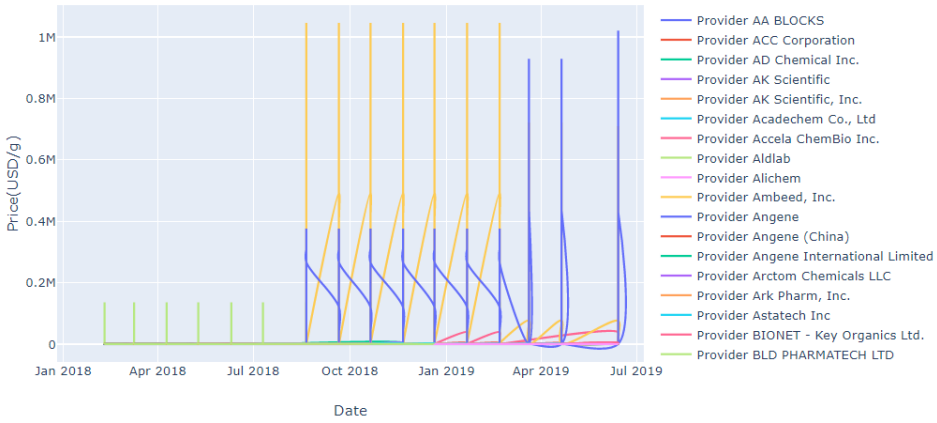


Figure 48: Time series plot of the prices for the metabolite 187. In this plot, each time series corresponds to the price evolution of a specific provider. As displayed, there are no values for January 2018 and May 2019, even though the methane's prices vary significantly.

Nonetheless, while observing figure 48, it should be noted that there are no prices for January 2018 and May 2019 in this metabolite. Subsequently, the new metabolite was subdued to the same methods executed in the section above, to filter out any additional errors and misvalues and, consequently, analyzing the price variation effectively. When visualizing the prices filtered by the configuration with the highest amount of prices in this metabolite (5mg), numerous different values were still detected in each month of the *MedChemExpress* and *MedChemExpress Europe* providers (Figure 49). These were thought to possess the same error in the external source ID, as noticed in the previous metabolite. Nonetheless, this was not the issue, since the external source ID used in the bioeconomics platform to get these prices, MolPort-018-618-244, corresponded to the exact metabolite. The figure 24 from section 4.3.1 corroborates to this situation.

However, a thing to notice in this metabolite is how few providers are related to the 5mg configuration of the methane metabolite (only four). Even so, this is the configuration with the most prices associated, due to the immense number of values included each month in only two of the providers. Furthermore, as seen in figure 49, the same MolPort ID got so many prices for the same configuration and month, which should not happen. As mentioned before in section 4.3.1, the reason for the existence of numerous prices in these two providers is related to the source.

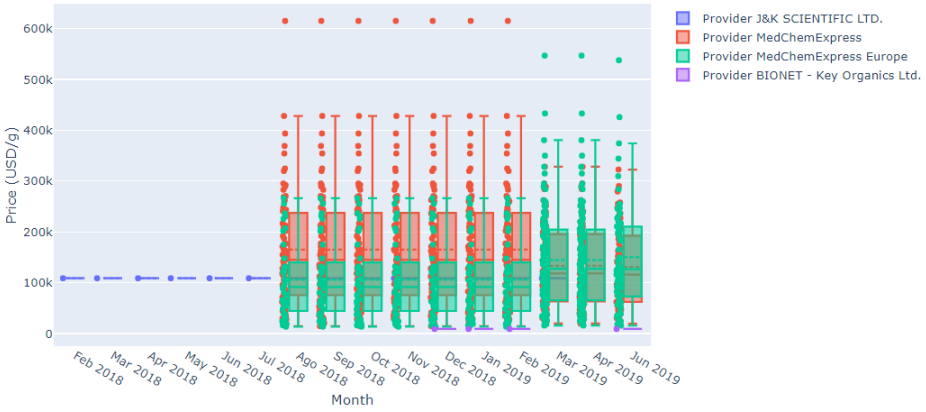


Figure 49: Boxplot of the price distribution per provider for the 5*mg* configuration of the metabolite 187. Each bullet along side the boxplots, represents an actual price value of the dataframe in question. As observed, the *MedChemExpress* and *MedChemExpress Europe* are the ones that have a greater distribution throughout the time window.

With that in mind, a thorough analysis in the MolPort source was accomplished in order to understand how many prices for the methane metabolite there were. When searching the source website, various CAS numbers related to the metabolite were encountered (Figure 24, section 4.3.1). However, as noticed above, the CAS number should be unique to the metabolite. Hence, an analysis to the price distribution per source was performed, considering all the prices related to the metabolite in question.

First thing to notice in figure 50, is that most of the prices, including the higher prices, came from the MolPort source. This may result from the various CAS numbers associated to this metabolite. Thus, this result also displays the source issue from the integration problem.

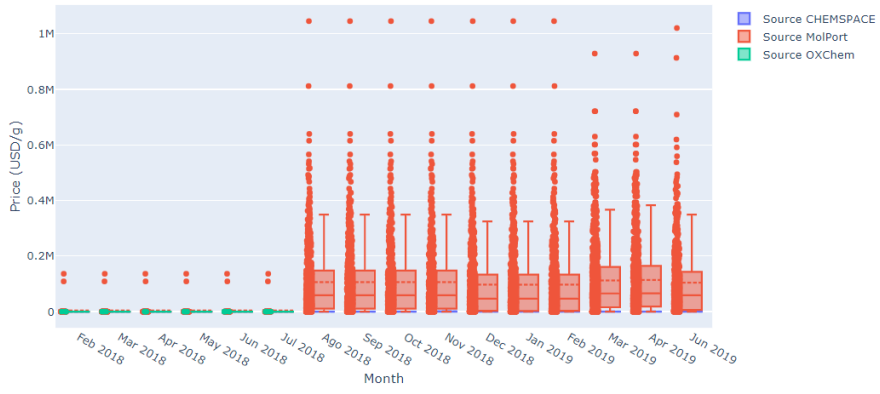


Figure 50: Boxplot of the price distribution per source of the methane metabolite (187). The bullets next to the boxplots are the actual values in the dataframe. Also, there are no prices for January 2018 and May 2019 in this metabolite. As noticed, the MolPort source is the one that has a greater distribution, leading to the previously observed variationst.

In short, in the methane metabolite, the MolPort source has 5157 prices in comparison to the 487 prices of the CHEMSPACE source and the 18 prices of the OXChem source. On the other hand, this source seems to have a fault in their website, consequently making the bioeconomics platform pick up a high number of prices that do not necessarily represent this metabolite. As a consequence, a great number of prices from this metabolite might be incorrect and cannot be used in further analysis.

Therefore, to solve this situation, the approach to these oscillations was the same as previously decided, displaying only the minimum price per provider in the months where numerous values existed. Bear in mind that most of the lower prices for the *MedChemExpress* and the *MedChemExpress Europe* have the same range as the other providers (Figure 49). Figure 51 displays the time series plot, consequently created after this method.

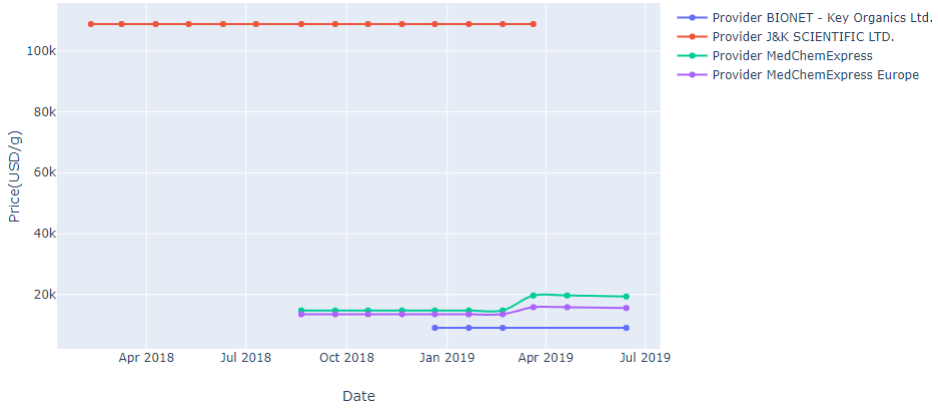


Figure 51: Time series plot of the minimum prices in each month and provider for $5mg$ of the methane metabolite (187). Each time series corresponds to the price variation of each provider in the filtered dataframe. Also, each bullet in the time series represents an actual price value existent in that dataframe. As observed, there are two providers, *BIONET - Key Organics Ltd.* and *JK SCIENTIFIC LTD.*, that do not vary.

In this representation (Figure 51), the actual number of prices expected for this configuration is obvious.

5.2.1 *Price Variation Analysis*

Observing the figure 51, the range of price values is disclosed as being situated between 9064 and 108820 USD/g. Since this is a wide-ranging interval and because it only includes four providers, the price analysis will be completed in other configurations of the methane metabolite. Bear in mind that this price interval is logical, considering that the higher the amount of a metabolite being sold, the lower its price. Moreover, in order to better compare both case studies, this analysis will be performed in the same configurations analysed in the 4-aminopyridine metabolite ($5g$ and $25g$).

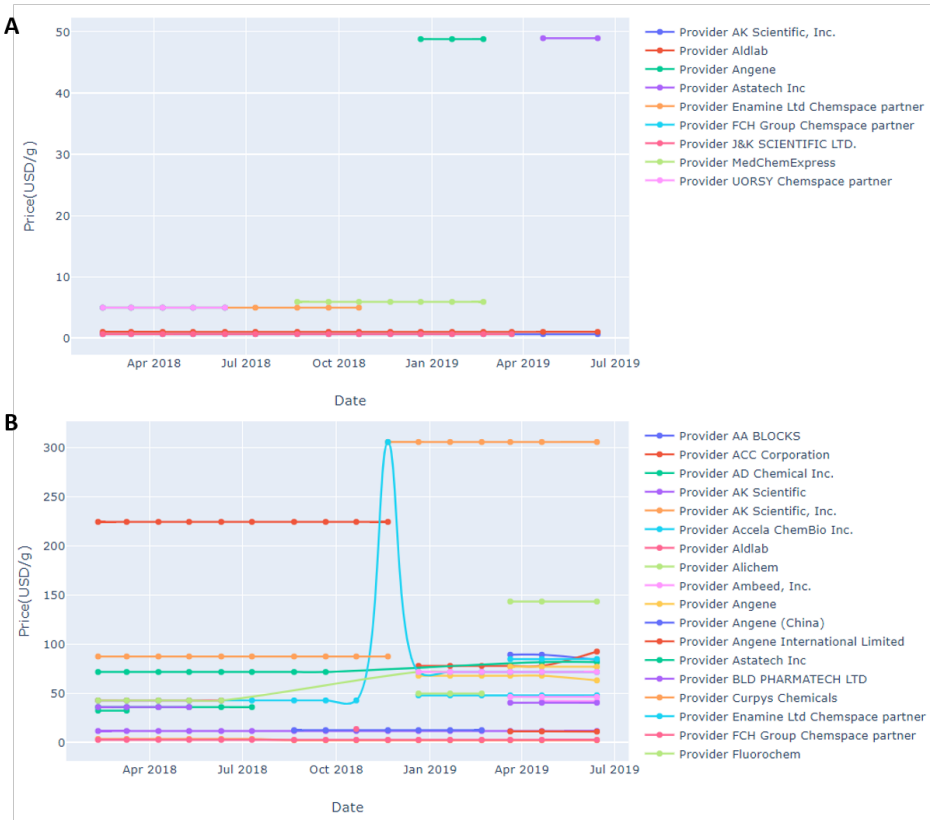


Figure 52: Time series plots of the minimum prices, in each month and provider, for 25g and 5g of the methane metabolite (187). A single time series corresponds to the price variation of a specific provider in the filtered dataframe. Also, each bullet in the time series represents an actual price value existent in that dataframe. (**A**) Time series plot of the prices related to the 25g configuration of the metabolite in question. (**B**) Time series plot of the prices related to the 5g configuration of the metabolite in question.

Comparing both of the time series plots, the first thing to notice is the number of values existent in each configuration. Since the 25g and 5g configurations had, respectively, 232 and 417 prices before the minimum filter for the integration problem, it is expected that figure 52(**B**) has more price values than figure 52(**A**). Moreover, the latter has also more providers associated (29 providers) in comparison to the former (9 providers).

Furthermore, while examining figure 52, the price range for both configurations was noted. For the 25g, the values reside between 0.64 and 48.94 USD/g, in comparison to the prices from the 5g that are located between 2.0 and 306.0 USD/g. This also confirms that the higher the configuration of a metabolite, the lower its prices will be (Section 2.2). Following this initial analysis, the minimum values were detected for each configuration. The figure above demonstrates these prices and their respective provider (Figure 53).

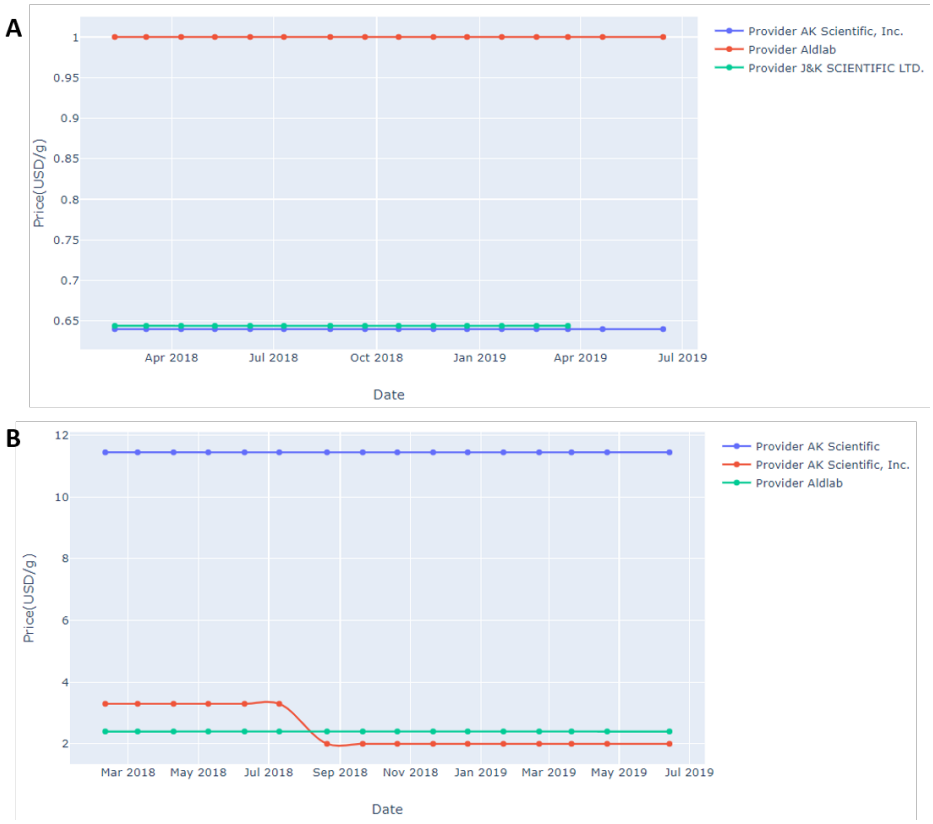


Figure 53: Time series plots of the lowest prices for 25g and 5g of the methane metabolite (187). Each bullet in the time series represents an actual price value existent in that dataframe. (**A**) Time series plot of the prices related to the 25g configuration. Here, the plot displays only the three providers with the lowest values, namely, *AK Scientific, Inc.*, *Aldlab* and *JK SCIENTIFIC LTD.* (**B**) Time series plot of the prices related to the 5g configuration. Here, the plot displays only the three providers with the lowest values, namely, *AK Scientific*, *AK Scientific, Inc.*, and *Aldlab*.

In this case, two providers are assess in both configurations as the lowest providers, the *AK Scientific, Inc.* and *Aldlab*, with the former possessing the minimum values of the respective dataframes (0.64 and 2.0 USD/g). Also, all the providers disclosed in figure 53 were fetched by the MolPort source.

In contrast to the previous case study, in this metabolite and these configurations, the cheapest provider correlates among the different situations. Although, this can be justified with the same reason given above, about the integration problem.

5.3 summary: 4-aminopyridine versus methane

Overall, these case studies served as a test for the development of algorithms and analyses. First and foremost, the preprocessing step was applied to df_clean dataframe creating new usable dataframes. This step led to available information without corruptions, such as missing and duplicate data, but also, to time series plots with a clear price variation. Therefore, a thorough examination of these variations could now be performed.

When examining how the price values vary, some detections about the cheapest providers and sources were executed. Considering the first case study, 4-aminopyridine, with the $25g$ configuration, the provider that included more consistently the minimum price value was the *Enamine Ltd Chemspace partner*. However, the least expensive providers did not have prices related to the year 2019. Hence, another provider was distinguished, the *Fluorochem Limited*, since this had the lowest prices taking into account a greater time interval. On the other hand, studying the $5g$ configuration, the providers with minimum price values did not matched the provider from the previous configuration. Whereas, the cheapest provider for the greater time interval, actually correlated to the prior analysis, the *Fluorochem Limited*. In regards to the sources related to these providers, the providers that did not correspond between configuration were all collected by CHEMSPACE, whereas the other one was gathered by MolPort. When bearing in mind the second case study, a correlation existed between the cheapest providers in both the $25g$ and $5g$ configuration (*AK Scientific, Inc.*). In addition, these prices were gathered by the MolPort source.

This lack of correlation and in some cases existence, can occur due to the integration problem, where a configuration can be present in one metabolite and the other can be found in the erroneous compound. Consequently, this leads to a fault in the analysis since there is no certainty that the values examined belong to the correct metabolite. In contrast, when considering the ranges between the configurations, the law of supply and demand mentioned in section 2.2, is confirmed, since the higher the amounts of these metabolites, the lower its prices are (Gale, 1955).

Nevertheless, to summarise both these case studies, a comparison between the two was performed. However, no correlation was discovered either. These two metabolites have distinct cheaper providers, as well as, sources. As a consequence, no relation can be withdrawn between them.

6

CONCLUSION

In short, the main goal of this dissertation was to develop algorithms to analyse the price variation of chemical compounds. Nevertheless, when exploring the data retrieved from the bioeconomics platform of SISBI, various problems were identified. Some of these issues included missing and duplicate data, normalization complications and integration problems. Therefore, several preprocessing methods were applied to this data, so that the price analysis outcome would appear more precise to the real result. Even though the first three issues could be solved after the data cleaning and normalization, the integration problem posed new complications.

Concerning the integration problem, two issues were detected. First, the fact that the bioeconomics platform contained some faults in its algorithms that led to the merge of different external source ID's related to different metabolites. Consequently, a particular metabolite included prices from other compounds, corrupting the data. Secondly, the same issue was observed, however, in this case the problem was located in the sources from which the prices were fetched. In other words, some sources had a glitch that made them gather prices from different metabolites, contributing to the existence of extra values in specific compounds. This problem ended up being dealt with a minimum filter where the lower prices of a particular month were kept, removing the additional values. However, this approach is not a solution to the real problem, since the faulty integration still happens. When applying this filter to the data, the presupposition that in every configuration there are prices related to the specific metabolite, is implied. Hence, taking into account the aforementioned assumption, this determines that when performing the analysis and comparison between the cheaper providers of a metabolite, any lack of correlation might occur due to the nonexistence of that metabolites' prices, committing a comparison between different compounds.

With that in mind, the conclusions drawn by the analysis of the price variation were ambiguous. When comparing different configurations from the same metabolite, some providers correlated, while most of them did not, this might happen because of the integration problem, as mentioned above. It was also not found any relation between the cheaper providers in different metabolites. Consequently, and due to the insufficient data for this

work, time series analysis, such as the identification of trends, seasonal variation, as well as, data predictions, could not be achieved. Another factor to take into account while analysing these time series was the lack of price variation for the majority of providers, and the lack of patterns and trends in the providers that did vary.

Besides this inconclusive price analysis, an experimental outlier algorithm was developed. Bear in mind that this algorithm only identifies the providers that contain outliers throughout the time window, and removes them from the analysis, not from the database.

Considering the conclusions withdrawn from this dissertation, some future work is required. First, in regards to the issues encountered in the data, instead of solving them after the information is added in the database, a solution in the bioeconomics platform is preferable. For example, when taking into account the duplicate prices and metabolites, the algorithms from this platform related to their collection and merge must be corrected in the future so that there is a lower possibility of duplicates existing. Alternatively, when populating the bioanalysis database, the bioeconomics platform ID should not be included, for the potential duplicate data not to be retrieved.

Furthermore, bearing in mind the integration problem detected both in the bioeconomics platform as in the sources, changes in the algorithm that retrieves the data are necessary. For instance, the opportunity to recognize if a source has other metabolites associated to a particular compound, or to restrict the merge between different metabolites in the platform itself, are all changes that could contribute to the solution of this issue in the future. These adjustments also contribute to a precise data analysis, without potential misinterpretations.

Next, due to the lack of broader time windows, the evaluation of the data fell short of what was intended. Therefore, another desirable step to take hereafter is to comprise more months in future analysis, thus being able to better examine the data and perform forecasting analysis. Also, in comparison to this dissertation, the bioanalysis database must contain all prices from bioeconomics.

Moreover, in the future, besides identifying the cheaper providers, other features, such as the sources, can also be analysed since they help to better complement and understand the information retrieved from the providers in this work.

Lastly, another approach that can be performed over time is the improvement of the algorithm for the removal of outliers. Some of the features accessible to enhance are the providers' identification and the detection of different categorized outliers, depending on the goal. In the future, rather than removing all providers that contain an outlier, the algorithm can be adapted so that exclusively providers with outliers for a specific number of months in the time window are eliminated, adjusting the removal limitation. Additionally, these can also be identified by testing distinct values of the constant applied in the equations, where higher values detect more severe outliers, leading to other analyses.

BIBLIOGRAPHY

Agnieszka, D. and Magdalena, L. (2018). Detection of outliers in the financial time series using arima models. In *2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE)*, pages 49–52. IEEE.

Benjamini, Y. (1988). Opening the box of a boxplot. *The American Statistician*, 42(4):257–262.

BiopharmaTrend.com. 19 online marketplaces facilitating life science research. https://www.biopharmatrend.com/post/32-8-online-marketplaces-facilitating-research-in-life-sciences/. Accessed: 10-09-2019.

Box, G. E., Jenkins, G., and Reinsel, G. C. (1976). Time series analysis prediction and control.

Box, G. E., Jenkins, G. M., and Reinsel, G. (1970). Time series analysis: forecasting and control holden-day san francisco. *BoxTime Series Analysis: Forecasting and Control Holden Day1970*.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brockwell, P. J., Davis, R. A., and Calder, M. V. (2002). *Introduction to time series and forecasting*, volume 2. Springer.

Carpenter, G. A. and Grossberg, S. (2016). *Adaptive resonance theory*. Springer.

Caudell, T. and Newman, D. (1993). An adaptive resonance architecture to define normality and detect novelties in time series and databases. In *IEEE World Congress on Neural Networks, Portland, Oregon*, pages 166–176.

Chandler, R. and Scott, M. (2011). *Statistical methods for trend detection and analysis in the environmental sciences*. John Wiley & Sons.

Chang, I. (1982). *Outliers in Time Series.* PhD thesis, University of Wisconsin-Madison.

Chang, I., Tiao, G. C., and Chen, C. (1988). Estimation of time series parameters in the presence of outliers. *Technometrics*, 30(2):193–204.

Chatfield, C. (2000). *Time-series forecasting*. Chapman and Hall/CRC.

Chatfield, C. (2003). *The analysis of time series: an introduction*. Chapman and Hall/CRC.

ChemSpace. Chemspace. https://chem-space.com/. Accessed January 3, 2019.

Chen, C. and Liu, L.-M. (1993). Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*, 88(421):284–297.

Cleveland, W. S. (1993). Visualizing data hobart press. *Summit, New Jersey,a paraıtre*.

Dahms, A. S. (2004). Biotechnology: What it is, what it is not, and the challenges in reaching a national or global consensus. *Biochemistry and Molecular Biology Education*, 32(4):271–278.

Dasgupta, D. and Forrest, S. (1996). Novelty detection in time series data using ideas from immunology. In *Proceedings of the international conference on intelligent systems*, pages 82–87.

Documentation, P. Pandas 0.25.1 documentation - pandas.dataframe.drop_duplicates. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html. Accessed August 8, 2019.

Documentation, P. Pandas 0.25.1 documentation - pandas.dataframe.drop_duplicates. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html. Accessed August 8, 2019.

Dornburg, V., Hermann, B. G., and Patel, M. K. (2008). Scenario projections for future market potentials of biobased bulk chemicals.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2006). Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 19(1):1–16.

eMolecules, I. emolecules. https://www.emolecules.com/. Accessed January 5, 2019.

Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12.

European Commission (2002). *Life Sciences and Biotechnology: A strategy for Europe*. European Commission: Brussels, Belgium.

European Commission (2011). *Bio-based Economy for Europe: State of Play and Future Potential—Part 1*. European Commission: Luxembourg, Belgium.

European Commission (2012). *Innovating for Sustainable Growth: A Bioeconomy for Europe*. European Commission: Brussels, Belgium.

Fox, A. J. (1972). Outliers in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 350–363.

Gale, D. (1955). The law of supply and demand. *Mathematica scandinavica*, pages 155–169.

García, S., Luengo, J., and Herrera, F. (2015). *Data preprocessing in data mining*. Springer.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., and Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1):9.

Gavrilescu, M. and Chisti, Y. (2005). Biotechnology—a sustainable alternative for chemical industry. *Biotechnology Advances*, 23(7):471 – 499.

Grossberg, S. (2013). Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, 37:1–47.

Gutierrez, F. (2014). *Introducing Spring framework: a primer*. Apress.

Heller, S. R., McNaught, A., Pletnev, I., Stein, S., and Tchekhovskoi, D. (2015). Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1):23.

Hermann, B., Blok, K., and Patel, M. K. (2007). Producing bio-based bulk chemicals using industrial biotechnology saves energy and combats climate change. *Environmental science & technology*, 41(22):7915–7921.

Hoaglin, D. C. and Velleman, P. F. A. (1981). Applications, basics and computing of exploratory data analysis. Technical report.

Jazwinski, A. H. (2007). *Stochastic processes and filtering theory*. Courier Corporation.

Jones, M., Nikovski, D., Imamura, M., and Hirata, T. (2014). Anomaly detection in real-valued multidimensional time series. In *International Conference on Bigdata/Socialcom/Cybersecurity. Stanford University, ASE*. Citeseer.

Karlin, S. and Taylor, H. M. (1975). A first course in stochastic processes.

Kempson, R. E. (1988). A genstat primer. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 37(1):86–87.

Kendall, M. G. and Ord, J. K. (1990). *Time series*. New York: Oxford University Press.

Kozma, R., Kitamura, M., Sakuma, M., and Yokoyama, Y. (1994). Anomaly detection by neural network models and statistical time series analysis. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, volume 5, pages 3207–3210. IEEE.

Lokko, Y., Heijde, M., Schebesta, K., Scholtès, P., Van Montagu, M., and Giacca, M. (2018). Biotechnology and the bioeconomy—towards inclusive and sustainable industrial development. *New biotechnology*, 40:5–10.

Masse, M. (2011). *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. " O'Reilly Media, Inc.".

McCormick, K. and Kautto, N. (2013). The bioeconomy in europe: An overview. *Sustainability*, 5(6):2589–2608.

Metcalfe, A. V. and Cowpertwait, P. S. (2009). *Introductory time series with R*. Springer.

Mills, T. C. and Mills, T. C. (1991). *Time series techniques for economists*. Cambridge University Press.

MolPort. Molport. https://www.molport.com/. Accessed January 3, 2019.

MongoDB, I. Mongodb. https://www.mongodb.com/. Accessed January 3, 2019.

Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2008). *Introduction to time series analysis and forecasting*. John Wiley & Sons.

Oxchem. Oxchem. www.ox-chem.com. Accessed January 3, 2019.

Pandit, S. M., Wu, S.-M., et al. (1983). *Time series and system analysis with applications*, volume 3. Wiley New York.

Pivotal Software, I. Spring boot. https://spring.io/projects/spring-boot. Accessed January 3, 2019.

Pollard, D. J. and Woodley, J. M. (2007). Biocatalysis for pharmaceutical intermediates: the future is now. *Trends in biotechnology*, 25(2):66–73.

Powell, W. W. and Brantley, P. (1992). *Competitive Cooperation in Biotechnology: Learning through Networks?*, pages 366–394. Harvard Business School Press, Boston.

Quintana-Garcia, C. and Benavides-Velasco, C. A. (2004). Cooperation, competition, and innovative capability: a panel data of european dedicated biotechnology firms. *Technovation*, 24(12):927–938.

Raña, P., Aneiros, G., and Vilar, J. (2015). Detection of outliers in functional time series. *Environmetrics*, 26(3):178–191.

RJa, L. and Rubin, D. (1987). Statistical analysis with missing data.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

SAS Institute Inc. (2019). Time Series. http://support.sas.com/documentation/cdl/en/etsug/60372/HTML/default/viewer.htm#etsug_arima_sect056.htm.

Spiegelhalter, D., Grigg, O., Kinsman, R., and Treasure, T. (2003). Risk-adjusted sequential probability ratio tests: applications to bristol, shipman and adult cardiac surgery. *International Journal for Quality in Health Care*, 15(1):7–13.

Standing, S., Standing, C., and Love, P. E. D. (2010). A review of research on e-marketplaces 1997–2008. *Decision Support Systems*, 49(1):41 – 51.

Sun, Y. and Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334.

TMIC. Drugbank. https://www.drugbank.ca/. Accessed January 3, 2019.

Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Reading, Mass.

Van Beuzekom, B. and Arundel, A. (2009). Oecd biotechnology statistics 2009. *Paris: Organization for Economic Cooperation and Development*.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186.

Walls, C. (2016). *Spring Boot in action*. Manning Publications.

Webb, P., Syer, D., Long, J., Nicoll, S., Winch, R., Wilkinson, A., Overdijk, M., Dupuis, C., and Deleuze, S. Spring boot reference guide.

Wei, W. W. (2013). Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*.

Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.

Whittle, P., Whittle, P., Whittle, P., Mathématicien, N.-Z., Whittle, P., Mathematician, N. Z., and Britain, G. (1963). *Prediction and regulation by linear least-square methods*. English Universities Press London.

Woodley, J. M. (2017). Bioprocess intensification for the effective production of chemical products. *Computers & Chemical Engineering*, 105:297–307.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210.