





## Article

# Open-Source Software Development in Cheminformatics: A Qualitative Analysis of Rationales

Johannes Perna<sup>1,2,\*</sup> , Aleksi Takala<sup>1</sup>, Veysel Ciftci<sup>3</sup>, José Hernández-Ramos<sup>4</sup> , Lizethly Cáceres-Jensen<sup>4</sup>  and Jorge Rodríguez-Becerra<sup>5</sup> 

<sup>1</sup> The Unit of Chemistry Teacher Education, Department of Chemistry, Faculty of Science, University of Helsinki, A.I. Virtasen Aukio 1, 00560 Helsinki, Finland; aleksi.takala@helsinki.fi

<sup>2</sup> Faculty of Education, University of Ljubljana, 1000 Ljubljana, Slovenia

<sup>3</sup> Omnia, Lakelankatu 1, 02770 Espoo, Finland; uveysel@hotmail.com

<sup>4</sup> Physical & Analytical Chemistry Laboratory (PachemLab), Department of Chemistry, Faculty of Basic Science, Universidad Metropolitana de Ciencias de la Educación, Santiago 7760197, Chile; jose.hernandez@umce.cl (J.H.-R.); lyzethly.caceres@umce.cl (L.C.-J.)

<sup>5</sup> Escuela de Postgrado, Universidad Tecnológica Metropolitana, Santiago 8940000, Chile; jorge.rodriguez@utem.cl

\* Correspondence: johannes.perna@helsinki.fi; Tel.: +358-50-348-0567

**Featured Application:** Cheminformatics is an emerging discipline of chemistry, and software engineering has a central role in this multidisciplinary research field. The background of cheminformatics is industry driven and has mainly produced closed-software solutions. However, the development of the field requires open-source technology. The purpose of this article is to explore the rationales behind open-source software development in cheminformatics. The acquired knowledge is important for the field in general from an intrinsic perspective, but it is particularly interesting from a cheminformatics education perspective. Through an understanding of reasons why open-source development is being carried out in cheminformatics, the field can build educational objectives through research-based knowledge.



**Citation:** Perna, J.; Takala, A.; Ciftci, V.; Hernández-Ramos, J.; Cáceres-Jensen, L.; Rodríguez-Becerra, J. Open-Source Software Development in Cheminformatics: A Qualitative Analysis of Rationales. *Appl. Sci.* **2023**, *13*, 9516. <https://doi.org/10.3390/app13179516>

Academic Editor: Vito Conforti

Received: 21 July 2023

Revised: 7 August 2023

Accepted: 21 August 2023

Published: 22 August 2023

**Abstract:** This qualitative research explored the rationales of open-source development in cheminformatics. The objective was to promote open science by mapping out and categorizing the reasons why open-source development is being carried out. This topic is important because cheminformatics has an industrial background and open-source is the key solution in promoting the growth of cheminformatics as an independent academic field. The data consisted of 87 research articles that were analyzed using qualitative content analysis. The analysis produced six rationale categories: (1) Develop New Software, (2) Update Current Features, Tools, or Processes, (3) Improve Usability, (4) Support Open-source Development and Open Science, (5) Fulfill Chemical Information Needs, and (6) Support Chemistry Learning and Teaching. This classification can be used in designing rationales for future software development projects, which is one of the largest research areas in cheminformatics. In particular, there is a need to develop cheminformatics education for which software development can serve as an interesting multidisciplinary framework.

**Keywords:** open source; cheminformatics; software development; qualitative research; content analysis



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Cheminformatics has been used in chemistry research since the field adopted computers in the 1940s [1]. Depending on the perspective, the first cheminformatics paper was published in 1946 by King et al. [2] or in 1957 by Ray and Kirsch [3]. According to Chen [1], King et al. [2] may be the first scholars who applied computers in chemistry research. However, according to Willet [4], the first actual cheminformatics paper was published in 1957 [3], where Ray and Kirsch described an algorithm for substructure searching.

Willet [4] argues that the importance of this article was its application of graph theory in searching and visualizing chemical structures. The algorithms have been significantly developed over the decades, but applying graph theory has been the foundation of major cheminformatics applications such as structure searching and registration, substructure searching, and similarity searching [4]. These applications are still the key techniques used in four traditional cheminformatics research areas, which are (1) chemical databases, (2) computer-assisted structure elucidation systems, (3) computer-assisted synthesis design systems, and (4) modeling and visualization tools [1]. On a practical level, cheminformatics scholars are developing methods, for example, to search large databases efficiently or model physical, chemical, and biological properties of molecules for predictive purposes [4].

Although cheminformatics as a field has been around for many decades, it has developed to a recognized and independent sub-discipline of chemistry only in recent decades [4,5]. Many scholars find this strange, as computers are widely used in chemistry and it has even been speculated that most novel chemistry entities in the future will be discovered *in silico* [6]. One reason is that much of the cheminformatics research has been conducted in industrial laboratories, not academia. Because of this, many applications and methods are not published due to intellectual-property-rights issues [4].

In this regard, the key solution for developing the field is to support free and open-source software (FOSS), which removes all limitations on users or applications [7]. This also includes industrial applications, which are a major component in the field of cheminformatics [4]. The potential of open-source development in cheminformatics has been recognized decades ago. There are multiple articles that describe the benefits of open source. For example, Wegner et al. [8] emphasized the importance of open-source resources in supporting the needs of the pharmaceutical industry by connecting chemistry and computer sciences via cheminformatical thinking. Gezelter [9] wrote that open source and open data should be a standard in chemical research. It would enable reproducibility which supports scientific reliability. Open source code would also lower the research costs significantly.

Derived from the described background, this article aims to promote open cheminformatics by exploring the rationales set for open-source projects declared as cheminformatics software development in the research literature. We argue that the topic is important because software development is one of the most active research areas in cheminformatics [1]. Even though the field produces a great deal of new software, according to our knowledge, there is no earlier research that has explored open-source software development in cheminformatics in a larger scale. Open source is especially important considering the hindrance that the industrial background has caused for the development of open cheminformatics [4].

A recent review revealed that cheminformatics research strongly emphasizes software, databases, and web applications developed for data analysis. This trend is likely continuing because of the current highly active research topics, such as machine learning, which demand up-to-date technological solutions [5]. In this context, cheminformatics-related programming skills are essential for chemistry research and education. They enable the creation of novel needs-based information solutions for solving specific tasks more efficiently than existing software [10].

In this case study, the rationales of open-source projects were analyzed by exploring cheminformatics articles ( $N = 87$ ) using qualitative content analysis [11]. Qualitative case study as a methodological approach produces descriptive narrative accounts that enable an understanding of the diverse reasons behind the cheminformatics open-source development [12]. The aim is not to offer a systematic synthesis of the reasons but rather to open a discussion of the possibilities, challenges, and solutions.

This research will be useful for the field of cheminformatics in general. Because software development is an essential cheminformatics research topic [5], there is an intrinsic rationale to explore the motivational factors [13]. In addition, the acquired knowledge will be particularly important for cheminformatics education. By generating an understanding of the reasons behind open-source cheminformatics, we will build a base for educational cheminformatics software development. Software development offers great possibilities

for promoting current pedagogical frameworks, such as computational thinking [14] in chemistry. We hope that our work will inspire and encourage chemists and future chemists to engage with cheminformatics-related open-source software development. We argue that this research can support cheminformatics education significantly because, as mentioned, software development has a central role in cheminformatics research [15,16].

Based on the described justifications, we have formulated the following research question for guiding the research: Why is open-source development being carried out in cheminformatics? The field can start research-based discussions on future design objectives when the rationales are mapped and classified. To answer the set research question, we applied a qualitative research strategy based on the general-to-specific order. Namely, we will first define open-source development and review its possibilities and challenges from a general perspective and reflect this insight on the context of chemistry software development (see Section 2). After the wider perspective, the focus will be moved to the specific rationales behind cheminformatics-specific open-source development via an inductive qualitative content analysis [11] (see Sections 3 and 4).

## 2. Open-Source Software Development in Chemistry

As mentioned, one major reason hindering the development of cheminformatics as an academic field has been its industry-driven background [4,17]. Cheminformatics has been used especially in drug discovery by the medical industry [8]. However, during the last two decades, cheminformatics has grown into an independent chemistry sub-discipline with academic objectives and research traditions growing alongside the industrial tradition [18].

In contrast to industrial cheminformatics research that mainly produces closed software, the key idea behind academic cheminformatics research is open science, which means open access to the literature, data, standards, and source code [19]. The closed approach is understandable for industrial stakeholders, but in an academic cheminformatics context, a closed approach makes it impossible to reproduce research settings and verify test results. This approach is against scientific practices, and Gezelter [9] suggests that access to data and source code should be a standard practice in the chemical research literature. Krylov et al. [20] partly disagree with Gezelter [9]. They agree that models and algorithms can be considered as scientific results and therefore published openly for peer-review. However, they argue that scientific software is a product which contains intellectual property rights and should not be published open by default. In addition, they highlight that professional software development includes time-consuming phases such as testing and documenting that are difficult to conduct without hired employees. Jacob [21] disagrees with Krylov et al. [20]; Jacob pointed out that all available code is produced via tax money to some extent. In conclusion, all these perspectives have solid arguments; the scientific discussion around chemistry FOSS is very active.

Open science practices for chemistry have been developed systematically over two decades. For example, the Blue Obelisk movement made important contributions in 2005–2011 to bring together researchers to develop open data procedures, open-source software, and open standards (e.g., Chemical Markup Language, InChI, OpenSMILES, and QSAR-ML) as resources for the chemistry community [19]. A recent perspective paper makes an important contribution to defining FOSS and evaluating its educational possibilities and challenges in the context of educational computational chemistry [7]. In their article, Lehtola and Karttunen define the following three main criteria for FOSS: it can be freely used, modified, and redistributed by anyone. Their definition emphasizes the demand for freedom but also fulfills the other demands for open software [22,23]. Note that in this article, free software and open software are considered synonymous.

In addition to reproducibility, FOSS has many practical advantages. First, it lowers research costs. Scholars can reuse software components in building their own solutions. Second, FOSS is an open declaration of skills and time spent on a project, which can be included in a CV [9]. A FOSS project may also be much more versatile and offer freedom for creative programmers and software architects, and can possibly lead to future job offers

in commercial companies [24,25]. Altogether, it seems that intrinsic motivation is a strong factor behind open-source development. Bitzer et al. [26] have summarized three initiators for intrinsic motivation: (1) a need for particular software or feature, (2) a possibility to have fun in a creative project, or (3) a desire to give a gift to the developer community that supports the public good. In education, open-source projects have been used as an educational context. For example, Pereira [27] studied the benefits of using open-source software in final degree projects. He found out that open-source projects offer practical real-world software engineering examples that increase students' skills, knowledge, and confidence. In addition, students can include the project contribution to their portfolio.

However, working with FOSS solutions does not exclude business. Lehtola and Karttunen [7] discuss multiple sustainable business models (e.g., maintenance and support) for FOSS that have worked both with consumer products and science solutions. On the company side, engaging with the FOSS project enables direct communication with developer communities [24]. On the other hand, Krylov et al. [20] find these business models naive. They speculate that this model may encourage FOSS developers to publish poor-quality solutions to create financial opportunities by making the software usable via commercial services. According to them, it would be better to make a top-of-the-line solution for a small fee.

In the last two decades, it has been recognized that there is an imbalance in the development of computational science, possibly because the foundations on which some software has been built are not adapted to keep pace with changes in hardware and applications. This results in software infrastructures that must constantly deal with the problem of continuity in maintaining these packages, which translates into lower productivity than expected by researchers and the industry [28,29]. In this sense, it seems clear that software sustainability requires time for development and, therefore, funding.

According to Gezelter [9], the sustainability of software development is a major challenge for chemical open-source software. Scientific software is often developed by domain scientists without experience in software engineering. This may lead to complex software architectures, code that is difficult to re-use, and short-lived or bad archiving practices [9]. For example, in the history of open-source development, defect management has been a challenge in even major projects such as Apache and Mozilla. This challenge could be addressed via proper project management [30]. On the other hand, open-source development offers quality tools such as critical peer review and free idea sharing [31]. Some papers published from open-source development lifecycle models can help developers make decisions on good work practices. For example, Saini and Kaur [32] have described and evaluated the advantages and disadvantages of different models.

There are also more critical perspectives. Hauschild et al. [33] argue that scientific institutions and funders are many times more interested in novel software solutions rather than quality. According to Stahl [17], the threshold for open-source projects is low, leading to a wide range of quality. A professional level is difficult to achieve in a hobby-based project without proper software engineering skills. Projects may encounter challenges, e.g., domain knowledge, legal issues, and technical skills. In addition, ideally, open-source projects have thousands of developers, enabling versatile idea transfer and peer-review. However, in reality, most projects have only a few developers at best. This is especially true in science fields that have fewer available developer candidates. For example, developing chemistry software requires interest towards the field and subject knowledge [17].

The wide range of quality results in varying user support. Swarts [34] studied the kinds of support questions and requests users have made when using open-source chemistry software. His data consisted of 25 open-source chemistry packages that had a support forum and documentation. Documentation was either task-based, focusing on the use of contexts and generalized principles of how software works, or feature-based, focusing on what the software can do via its features. According to his research, users had the following three problem areas: to understand how the program works (transparency); how to learn to use it (learnability); and how easy it is to use (usability). Software that offered

task-based documentation seemed to have more challenges in usability, whereas feature-based documentation led to challenges in transparency and learnability [34]. The above description is one example of a potential quality challenge among chemistry open-source development. A general solution for improving the quality would be a close collaboration between chemists and computer scientists [8].

Another approach to quality improvement would be to adopt good practices from other closely related research fields, such as health informatics [33]. In health informatics, the quality of medical device software is maintained via several regulations. These regulations offer standards for the whole software development lifecycle including planning, architectural design, testing, verification, maintenance, and documentation [35].

In summary, software development is the most active research topic inside cheminformatics [5]. Moreover, the discussion around this topic is extremely rigorous. This topic is essential because all chemistry research uses some kind of computer software at some research stage [21]. Therefore, computer literacy skills in general are important for all chemists. Such skills help researchers understand how computers and software work, which enables, e.g., more efficient testing of hypotheses [36].

### 3. Methods

The research was conducted via a qualitative approach to match the set aim. This research is classified as a case study of retrieved articles [12]. The selected articles report on open-source software development projects published in the field of cheminformatics. By analyzing the aims, justifications, and outcomes described by the authors, we can provide qualitative answers to the set research question by mapping out the diversity of rationales that have been driving software development in the field.

#### 3.1. Data Gathering

The data were retrieved between 2019–2023 using Google Scholar, article databases accessed via information-retrieval tools offered by the University of Helsinki (e.g., PubMed, Scopus, and ProQuest Databases), and directly from cheminformatics journals, such as the Journal of Cheminformatics and Journal of Chemical Information and Modeling. The first data retrieval cycle was conducted in 2019. The data sample was updated during 2020–2023 by adding new software.

Inside cheminformatics journals, the data were gathered via search phrases such as “open source” and “software development”. For the larger multidisciplinary databases, such as Google Scholar or the University of Helsinki search tools, we used cheminformatics-specific strings, such as (“open-source software” and “cheminformatics”), (cheminformatics and “software development”), and (“open source” and cheminformatics). This information-seeking strategy resulted in 87 relevant research articles addressing the open-source software development of cheminformatics (see Appendix A). These 87 documents are the case of the study. Note that the set aim and research question did not guide this research to produce a systematic review of rationales. However, the qualitative approach set an important requirement for the data: the number of articles must be sufficiently large such that the number of main categories forming during the analysis will saturate [12].

#### 3.2. Data Analysis

The rationales were analyzed using inductive content analysis [11]. The analysis was performed in small iterative cycles, in which one researcher contributed to the analysis and another reviewed the work.

1. First, the articles were read and pre-screened to ensure that they addressed open-source development in cheminformatics. During the pre-screening, articles were listed in an Excel sheet with some additional notes for the following stages.
2. After pre-screening, the relevant 87 articles were imported into the ATLAS.ti 9 software [37]. Then, the articles were read one by one, and all paragraphs related to



- rationales were highlighted. Abstracts, introductions, discussions, and conclusions were read with additional care as they are usually the locations for rationale statements.
3. After the screening of rationales, highlights were simplified and reduced into subcategories using the ATLAS.ti coding feature.
  4. Last, subcategories were classified into main categories that were saturated during the analysis (see Table 1). The main categories were formed via the ATLAS.ti code group feature.
  5. A Cohen's  $\kappa$  inter-rater reliability test was conducted to ensure the reliability and validity of the analysis.  $\kappa$  is a statistical model that illustrates the degree of agreement among raters. In the inter-rater reliability test, an expert outside the analysis, usually a member of the research group or larger research community, repeats the classification based on the prepared class descriptions. This process is used in qualitative content analysis to improve the validity and reliability of the analysis. A  $\kappa$  value  $> 0.80$  indicates a strong level of agreement [38]. In the  $\kappa$  verification phase, two authors that did not participate in the analysis re-categorized approximately 15% of the original highlights into the main categories via a blind process.

**Table 1.** Analysis procedure.

Original Expression	Sub-Category	Main Category
<i>"Despite these efforts, no general purpose deterministic structure generator has been developed in an open source format so far." [39]</i>	No available open-source alternative	Develop New Software
<i>"The ChemoPy package aims at providing the user with comprehensive implementations of these descriptors in a unified framework to allow easy and transparent computation." [40]</i>	Clear workflow	Improve Usability

#### 4. Results

The initial analysis produced nine rationale categories for open-source software development in cheminformatics. However, during the  $\kappa$  phase, the number of main categories was reduced to six, providing a  $\kappa$  value of 0.83, which indicates strong agreement of the main categories and their descriptions [38].

Next, we describe the main categories and provide a few examples from each category. A comprehensive list of all main and sub-categories can be found in Appendix B.

##### 4.1. Develop New Software

There is a need to design software if scholars feel that a certain type of software does not exist. The lack of available solutions can focus on general or more specific needs. An example from a more general need would be the development of an open-source software alternative, mobile version, or a comprehensive solution that is unifying (such as features, databases, platforms, or frameworks).

- There is no available open-source alternative.  
*"Despite these efforts, no general purpose deterministic structure generator has been developed in an open source format so far." [39]*
- There is a need for, e.g., cross-platform, cross-database, and web-based mobile solutions that are not available.  
*"The increasing number of organic and inorganic structures promotes the development of the "Big Data" in chemistry and material science, and raises the need for cross-platform and web-based methods to search, view and edit structures." [41]*

- There is a lack of software for certain chemical tasks.

*"However, for specific requirements of in-house databases and processes no such solutions exist."* [42]

An example of a specific need would be to design software for implementing new features, tools, or processes:

- Features such as advanced search features, annotation of search results or the inter-conversion of chemical files.

*"A frequent problem in computational modeling is the interconversion of chemical structures between different formats. While standard interchange formats exist (for example, Chemical Markup Language) and de facto standards have arisen (for example, SMILES format), the need to interconvert formats is a continuing problem due to the multitude of different application areas for chemistry data, differences in the data stored by different formats (0D versus 3D, for example), and competition between software along with a lack of vendor neutral formats."* [10]

- Tools for, e.g., importing, exporting, editing, and visualization.

*"The field of molecular graphics is dominated by viewers with little or no editing capabilities,..."* [43]

- Processes, e.g., based on open file formats, programmatic file conversion, or process automation for decreasing manual work.

*"The success of implementing Jmol SMILES and Jmol SMARTS within Jmol simply provides an example of the continued power of SMILES and SMARTS in the cheminformatics open-source community."* [44]

*"Processing a large number of compounds through the scheme can be a time-intensive activity, in an effort to automate such evaluations...."* [45]

With these specific needs, there are already other software alternatives, but scholars have experienced challenges in using them, such as software bugs, slow performance, or poor architecture with, e.g., databases. The rationale can also relate to improving the research infrastructure if there are only closed-license options available that are expensive and prevent software development.

- Software bugs, slow performance, or poor architecture.

*"However, users of those programs must contend with several issues, including software bugs, insufficient update frequencies, and software licensing constraints."* [46]

- Specific needs, such as limited database options or need for workflows for sensitive data.

*"The local installation is a good alternative to online solutions without the inconvenient of sending sensitive structures over the Internet".* [47]

- The rationale can also relate to improving the research infrastructure if there are only closed-license options available that are expensive and prevent software development.

*"However, these solutions may be costly especially if they also require a commercial relational database management system (RDBMS)."* [42]

#### 4.2. Update Current Features, Tools, or Processes

The rationale of software development can be justified by improving or updating already existing parts of the software. These projects often lead to new versions, which developers are familiar with. For example,

- Improving features for the next software generation.

*"These include: deposited data sets from neglected disease screening; crop protection data; drug metabolism and disposition data and bioactivity data from patents. A number of improvements and new features have also been incorporated."* [48]

- Updating processes and workflows, for example, based on new algorithms or standards.  
*"In this work, the Ertl algorithm for automated FG detection and extraction is implemented on the basis of the Chemistry Development Kit (CDK) [3–6] with a new Java class ErtlFunctionalGroupsFinder to extend its open applicability for molecular research."* [49]  
*"In order to facilitate the use of the database, a key objective of the ChEMBL compound curation process is to standardise the chemical structures stored in the database and to assign a unique identifier to each distinct chemical structure regardless of the source."* [50]
- Fixing errors found from the current version.  
*"We investigate the reason for the predictivity difference with CheS-Mapper. We highlight the prediction error difference for each compound to determine which compounds are predicted more accurately by which approach."* [51]

#### 4.3. Improve Usability

Improving usability is one major rationale for cheminformatics software development. Many authors have aimed to produce user-friendly solutions by developing graphical, command-line, or web-based interfaces, producing interactive instructions or ready-made scripts that reduce the need for programming. On the other hand, improvements in usability can also mean faster calculations or easier installation and maintenance processes, which save researchers' time. For example,

- Clear workflow;  
*"The ChemoPy package aims at providing the user with comprehensive implementations of these descriptors in a unified framework to allow easy and transparent computation."* [40]  
*"The more concise, clear, and accessible a toolkit is, the less time they spend learning syntax and the more time they spend solving chemistry problems. Ruby is designed to be intuitive, concise, and powerful."* [52]
- New user interface;  
*"Along with a new architecture and user interface, this version will include internationalization, interactive instructions..."* [46]
- Ready-made resources;  
*"Unlike various other open source software packages, the primary focus of MayaChemTools is to provide out-of-the-box scripts to appeal to a wider audience."* [53]
- Easy installation.  
*"Second, it is easy to use. Mordred can be installed using only one command, whereas other Python molecular descriptor calculation libraries (e.g., cinfony, ChemoPy) have more dependencies that require manual installation."* [46]

#### 4.4. Support Open-Source Development and Open Science

In addition to software and specific features, open-source development also produces open resources, such as

- Technological and chemical frameworks for developers (e.g., related to HTML5, CSS, JavaScript, Ruby, MySQL, SMIRKS, and QSAR).  
*"SMIRKS package would provide the opportunity for development of new tools for resolving various reaction-oriented chemical information problems."* [54]
- Standards that support the work of software developers (e.g., standards for computational processes, data management, and software development).  
*"The establishment of infrastructure in academic institutions is particularly difficult due to missing standards or policies in data handling and storage..."* [55]



The rationale can also be related to science politics aiming to support open science. For example,

- To support open data standards and open data policies.  
*“TB Mobile is a simple to use app with useful functionality for viewing and manipulating data about compounds with activity against Mtb, their targets and other related information. The app represents a significant development in the effort to make accessible drug discovery data freely available in a form that is highly useful to scientists in general, not just cheminformatics experts.” [56]*
- To support FOSS thinking in general and build cooperation between academia and industry. Enabling this cooperation is dependent on the selected licenses.  
*“The use of this license is intended to achieve the secondary goal of allowing the integration of the software into proprietary software, thus facilitating scientific cooperation between industry and academia by eliminating the need to overcome license limitations. This way, the use of the library in both commercial and noncommercial environments is encouraged.” [57]*
- To support sustainable science via open-source development.  
*“In reviewing options for a sustainable future solution that also removed the dependence on commercial software it became apparent that none of the existing toolkits fitted the ChEMBL group’s requirements. Therefore, the decision was made to build a curation pipeline around the widely used open-source RDKit toolkit and its implementation of the MolVS molecule validation and standardisation tool.” [50]*

#### 4.5. Fulfil Chemical Information Needs

Another approach for cheminformatics software development is not to improve the technology itself but to aim to produce some specific knowledge raised from the chemical information needs. This means research groups need some chemical knowledge that current software options do not provide. To fulfill the specific information need, they must develop software that produces it. For example,

- To develop software that produces specific chemical information, such as extract bioactivity data;  
*“However, the current SureChEMBL system only extracts compound structures from the patents and not associated bioactivity data.” [48]*
- To screen chemical space systematically and efficiently;  
*“Despite the increasing throughput of screening technologies, the almost infinite chemical space remains out of reach, calling for tools dedicated to the analysis and selection of the compound collections intended to be screened.” [58]*
- To process ADMET models in drug discovery.  
*“AZOrange is a general Open Source platform for machine learning, However, developed to meet the increasing demand for ADMET models in drug discovery in particular.” [59]*

#### 4.6. Support Chemistry Learning and Teaching

Cheminformatics knowledge is needed to produce chemistry education software and applications that aim to support learning and teaching. The past research literature offers some examples of use cases, such as the following:

- Support spatial learning by implementing augmented reality;  
*“... , we propose a technological solution to aid the spatial learning process by automatically creating a link between two-dimensional (2D) representations of chemical structures and three-dimensional (3D) molecular visualization.” [60]*

- Support learning of structure–property–relationships via calculations;  
*“The idea was to have students develop a chemical intuition about how molecular structure affects molecular properties, without performing the underlying calculations by hand (which would be nearly impossible for all but the simplest chemical systems).”* [61]
- Support chemical reading.  
*“However, no study has investigated whether a reading-aloud system used prior to a tactile system can give those with visual disabilities greater understanding of chemical structures drawn in textbooks, chemical literature, and patents or even on a computer screen.”* [62]

## 5. Discussion

The analysis produced six main rationale categories for open-source cheminformatics software development (see Table 2). The perspective of the rationale can be either general or specific. For example, new software can be developed with the aim to produce an open alternative to compete with the currently closed solutions [39]. Alternatively, the rationale may be focused on solving some specific challenge, such as developing a locally installed software that does not require sending sensitive data or structural trade secrets over the internet [47]. Usability-related rationales (3) mainly have general objectives, whereas updates (2) and information needs (5) have more specific aims. Rationales 1 (new software), 4 (open science), and 6 (learning) may have both general and specific aims. Most rationales produce technological outcomes (1–4), such as new software, frameworks, interfaces, and processes. Chemical information needs to (5) produce content-driven solutions. Open-source development can also be used to promote open science, which has a science politics perspective (4) [63]. Last, if the rationale is to support teaching and learning (6), cheminformatics knowledge is used to develop educational technology [64].

**Table 2.** A Synthesis of findings.

#	Rationale	Perspective	Outcome
1	Develop New Software	General/Specific	Technological
2	Update Current Features, Tools, or Processes	Specific	Technological
3	Improve Usability	General	Technological
4	Support Open-source Development and Open Science	General/Specific	Technological/Political
5	Fulfill Chemical Information Needs	Specific	Content-driven
6	Support Chemistry Learning and Teaching	General/Specific	Pedagogical

The work started in 2005 by the Blue Obelisk movement [19] seems to have a strong impact, as many of the analyzed projects have been using the central standards defined by Blue Obelisk. Developing standards is an important objective for the field. The advantage of standards is clear documentation. Such standards are easily applied by software development professionals and are needed in multidisciplinary cheminformatics projects. Domain experts are the best at designing chemistry but lack software development skills [9]. However, the analysis shows that some scientific software is developed ad hoc by scientists rather than software engineers, either because of the need for a quick solution, lack of resources, or the excitement of writing software code. In good agreement with Blanton [29], this situation could hinder sustainable software development in cheminformatics due to funding variability, adherence to good software development practices, iterative software peer review processes, and even formal training of scientists in software engineering.

According to the earlier research literature, the diversity of the quality of open-source solutions and maintenance processes were the biggest challenges in the development of scientific software [9]. This may be the case, but the acquired data did not consider quality issues. The authors usually do not describe software or process faults in the articles.

However, one of the rationale categories (2) was updating current features. In these papers, the authors reported bugs and errors that they aimed to fix [51]. In addition, there were no descriptions of software development project management, so this cannot be evaluated with these data. One solution to improve the quality challenge would be to adopt good practices from other domains that have already developed quality management systems for scientific software development. See some examples from health informatics [33,35].

Improving usability is another major rationale that is related to many other rationales, such as updating or developing new features. Usability can mean, e.g., the improvement of interface, performance, maintenance, or operating logic. The user must experience confidence in using the software. One way to do this is to improve the instructions [65]. With instructions, the developers must decide whether the emphasis is on tasks or features [34]. The ideal solution would be to assist users in understanding how the software is working to avoid the black-box effect [39] and help navigating inside the software

Lastly, the most central challenge hindering academic development has been the industrial background. Industrial solutions cannot be published under open license because of trade secrets and intellectual property rights [4]. While this is understandable, we see this as an opportunity and not a challenge. Strong industrial relations are a strength that enables cheminformatics to grow both academically and industrially [18]. Open-source development can bridge these two stakeholders, from which there are already good examples [47,57]. Industry can provide jobs and research resources for academia, and academia can produce new scientifically validated solutions. Through scientific practices, the developed solutions are automatically peer evaluated [31].

In conclusion, this research produced six main rationale categories for open-source cheminformatics software development. The results are reliable, as we retrieved extensive article data that led to a saturation of findings. Note that the subcategories did not and cannot ever saturate, as science progresses through a mechanism of finding research gaps from earlier research. Appendix B can be used as a list of ideas to see what objectives others have set. In addition, the analysis procedure was conducted via the latest requirements for qualitative content analysis ( $\kappa = 0.83$ ) [11,37,38].

This classification can be used for designing rationales for future software development projects, one of the biggest research areas in cheminformatics [1,5]. In particular, there is a considerable need for developing cheminformatics education [15,16,66]. Open-source software development can serve as a good educational context, as it promotes multidisciplinary content knowledge (e.g., chemistry knowledge, cheminformatics skills, and computer sciences) and teaching about open science and scientific practices.

**Author Contributions:** Conceptualization, J.P.; methodology, J.P.; validation, J.H.-R. and J.R.-B.; formal analysis, J.P., A.T. and V.C.; investigation, A.T.; data curation, V.C.; writing—original draft preparation, J.P.; writing—review and editing, J.P., A.T., J.H.-R., J.R.-B. and L.C.-J.; visualization, J.P.; supervision, J.P.; project administration, J.P.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding. The Open access funding is provided by University of Helsinki.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Third party data cannot be shared due to copyrights. The article data were obtained from publishers' databases using the paid access provided by the University of Helsinki. The articles are listed alphabetically in Appendix A.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Analyzed Documents

1. Ambure, P.; Halder, A.K.; González Díaz, H.; Cordeiro, M.N.D.S. QSAR-Co: An Open Source Software for Developing Robust Multitasking or Multitarget Classification-Based QSAR Models. *J. Chem. Inf. Model.* **2019**, *59*, 2538–2544, doi:10.1021/acs.jcim.9b00295.
2. Andrei, R.M.; Callieri, M.; Zini, M.F.; Loni, T.; Maraziti, G.; Pan, M.C.; Zoppè, M. Intuitive Representation of Surface Properties of Biomolecules Using BioBlender. *BMC Bioinformatics* **2012**, *13*, S16, doi:10.1186/1471-2105-13-S4-S16.
3. Bauer, M.A.; Berleant, D.; Cornell, A.P.; Belford, R.E. WikiHyperGlossary (WHG): An Information Literacy Technology for Chemistry Documents. *Journal of Cheminformatics* **2015**, *7*, 22, doi:10.1186/s13321-015-0073-7.
4. Bento, A.P.; Gaulton, A.; Hersey, A.; Bellis, L.J.; Chambers, J.; Davies, M.; Krüger, F.A.; Light, Y.; Mak, L.; McGlinchey, S.; et al. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res* **2014**, *42*, D1083–D1090, doi:10.1093/nar/gkt1031.
5. Bento, A.P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L.J.; De Veij, M.; Leach, A.R. An Open Source Chemical Structure Curation Pipeline Using RDKit. *Journal of Cheminformatics* **2020**, *12*, 51, doi:10.1186/s13321-020-00456-1.
6. Bergwerf, H. MolView: An Attempt to Get the Cloud into Chemistry Classrooms. *DivCHED CCCE: Committee on Computers in Chemical Education* **2015**.
7. Berthold, M.R.; Cebron, N.; Dill, F.; Gabriel, T.R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME—The Konstanz Information Miner: Version 2.0 and Beyond. *SIGKDD Explor. Newsl.* **2009**, *11*, 26–31, doi:10.1145/1656274.1656280.
8. Bienfait, B.; Ertl, P. JSME: A Free Molecule Editor in JavaScript. *Journal of Cheminformatics* **2013**, *5*, 24, doi:10.1186/1758-2946-5-24.
9. Burger, M.C. ChemDoodle Web Components: HTML5 Toolkit for Chemical Graphics, Interfaces, and Informatics. *Journal of Cheminformatics* **2015**, *7*, 35, doi:10.1186/s13321-015-0085-3.
10. Cao, D.-S.; Xu, Q.-S.; Hu, Q.-N.; Liang, Y.-Z. ChemoPy: Freely Available Python Package for Computational Biology and Chemoinformatics. *Bioinformatics* **2013**, *29*, 1092–1094, doi:10.1093/bioinformatics/btt105.
11. Capoferri, L.; van Dijk, M.; Rustenburg, A.S.; Wassenaar, T.A.; Kooi, D.P.; Rifai, E.A.; Vermeulen, N.P.E.; Geerke, D.P. ETOX ALLIES: An Automated Pipeline for Linear Interaction Energy-Based Simulations. *Journal of Cheminformatics* **2017**, *9*, 58, doi:10.1186/s13321-017-0243-x.
12. Carrió, P.; López, O.; Sanz, F.; Pastor, M. ETOXlab, an Open Source Modeling Framework for Implementing Predictive Models in Production Environments. *Journal of Cheminformatics* **2015**, *7*, 8, doi:10.1186/s13321-015-0058-6.
13. Chen, P.; Wang, Y.; Yan, H.; Gao, S.; Xu, Z.; Li, Y.; Mo, Q.; Huang, J.; Tao, J.; Pan, G.; et al. 3DStructGen: An Interactive Web-Based 3D Structure Generation for Non-Periodic Molecule and Crystal. *Journal of Cheminformatics* **2020**, *12*, 7, doi:10.1186/s13321-020-0411-2.
14. Clark, A.M.; Sarker, M.; Ekins, S. New Target Prediction and Visualization Tools Incorporating Open Source Molecular Fingerprints for TB Mobile 2.0. *Journal of Cheminformatics* **2014**, *6*, 38, doi:10.1186/s13321-014-0038-2.
15. Dallakian, P.; Haider, N. FlaME: Flash Molecular Editor—A 2D Structure Input Tool for the Web. *Journal of Cheminformatics* **2011**, *3*, 6, doi:10.1186/1758-2946-3-6.
16. Dong, J.; Yao, Z.-J.; Wen, M.; Zhu, M.-F.; Wang, N.-N.; Miao, H.-Y.; Lu, A.-P.; Zeng, W.-B.; Cao, D.-S. BioTriangle: A Web-Accessible Platform for Generating Various Molecular Representations for Chemicals, Proteins, DNAs/RNAs and Their Interactions. *Journal of Cheminformatics* **2016**, *8*, 34, doi:10.1186/s13321-016-0146-2.
17. Dong, J.; Yao, Z.-J.; Zhang, L.; Luo, F.; Lin, Q.; Lu, A.-P.; Chen, A.F.; Cao, D.-S. PyBioMed: A Python Library for Various Molecular Representations of Chemicals, Proteins and DNAs and Their Interactions. *Journal of Cheminformatics* **2018**, *10*, 16, doi:10.1186/s13321-018-0270-2.

18. Ekins, S.; Clark, A.M.; Sarker, M. TB Mobile: A Mobile App for Anti-Tuberculosis Molecules with Known Targets. *Journal of Cheminformatics* **2013**, *5*, 13, doi:10.1186/1758-2946-5-13.
19. Enciso, M.; Meftahi, N.; Walker, M.L.; Smith, B.J. BioPPSy: An Open-Source Platform for QSAR/QSPR Analysis. *PLOS ONE* **2016**, *11*, e0166298, doi:10.1371/journal.pone.0166298.
20. Fatemah, A.; Rasool, S.; Habib, U. Interactive 3D Visualization of Chemical Structure Diagrams Embedded in Text to Aid Spatial Learning Process of Students. *J. Chem. Educ.* **2020**, *97*, 992–1000, doi:10.1021/acs.jchemed.9b00690.
21. Fritsch, S.; Neumann, S.; Schaub, J.; Steinbeck, C.; Zielesny, A. ErtlFunctionalGroups-Finder: Automated Rule-Based Functional Group Detection with the Chemistry Development Kit (CDK). *Journal of Cheminformatics* **2019**, *11*, 37, doi:10.1186/s13321-019-0361-8.
22. Gaulton, A.; Bellis, L.J.; Bento, A.P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; et al. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res* **2012**, *40*, D1100–D1107, doi:10.1093/nar/gkr777.
23. Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A.P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L.J.; Cibrián-Uhalte, E.; et al. The ChEMBL Database in 2017. *Nucleic Acids Res* **2017**, *45*, D945–D954, doi:10.1093/nar/gkw1074.
24. Guilloux, V.L.; Arrault, A.; Colliandre, L.; Bourg, S.; Vayer, P.; Morin-Allory, L. Mining Collections of Compounds with Screening Assistant 2. *Journal of Cheminformatics* **2012**, *4*, 20, doi:10.1186/1758-2946-4-20.
25. Gütlein, M.; Karwath, A.; Kramer, S. CheS-Mapper—Chemical Space Mapping and Visualization in 3D. *Journal of Cheminformatics* **2012**, *4*, 7, doi:10.1186/1758-2946-4-7.
26. Gütlein, M.; Karwath, A.; Kramer, S. CheS-Mapper 2.0 for Visual Validation of (Q)SAR Models. *Journal of Cheminformatics* **2014**, *6*, 41, doi:10.1186/s13321-014-0041-7.
27. Hanson, R.M. Jmol SMILES and Jmol SMARTS: Specifications and Applications. *Journal of Cheminformatics* **2016**, *8*, 50, doi:10.1186/s13321-016-0160-4.
28. Hanson, R.M.; Prilusky, J.; Renjian, Z.; Nakane, T.; Sussman, J.L. JSmol and the Next-Generation Web-Based Representation of 3D Molecular Structure as Applied to Proteopedia. *Israel Journal of Chemistry* **2013**, *53*, 207–216, doi:10.1002/ijch.201300024.
29. Hanwell, M.D.; Curtis, D.E.; Lonie, D.C.; Vandermeersch, T.; Zurek, E.; Hutchison, G.R. Avogadro: An Advanced Semantic Chemical Editor, Visualization, and Analysis Platform. *Journal of Cheminformatics* **2012**, *4*, 17, doi:10.1186/1758-2946-4-17.
30. He, Y.; Liew, C.Y.; Sharma, N.; Woo, S.K.; Chau, Y.T.; Yap, C.W. PaDEL-DDPREDICTOR: Open-Source Software for PD-PK-T Prediction. *J Comput Chem* **2013**, *34*, 604–610, doi:10.1002/jcc.23173.
31. Hildebrandt, A.; Dehof, A.K.; Rurainski, A.; Bertsch, A.; Schumann, M.; Toussaint, N.C.; Moll, A.; Stöckel, D.; Nickels, S.; Mueller, S.C.; et al. BALL—Biochemical Algorithms Library 1.3. *BMC Bioinformatics* **2010**, *11*, 531, doi:10.1186/1471-2105-11-531.
32. Hofmann, A.; Coster, M.J.; Taylor, P. Disseminating a Free, Practical Java Tool To Interactively Generate and Edit 2D Chemical Structures. *J. Chem. Educ.* **2019**, *96*, 1262–1267, doi:10.1021/acs.jchemed.9b00073.
33. Hoksza, D.; Škoda, P.; Voršilák, M.; Svozil, D. Molpher: A Software Framework for Systematic Chemical Space Exploration. *Journal of Cheminformatics* **2014**, *6*, 7, doi:10.1186/1758-2946-6-7.
34. Jeliazkova, N.; Jeliazkov, V. AMBIT RESTful Web Services: An Implementation of the OpenTox Application Programming Interface. *Journal of Cheminformatics* **2011**, *3*, 18, doi:10.1186/1758-2946-3-18.
35. Jensen, J.H.; Kromann, J.C. The Molecule Calculator: A Web Application for Fast Quantum Mechanics-Based Estimation of Molecular Properties. *J. Chem. Educ.* **2013**, *90*, 1093–1095, doi:10.1021/ed400164n.



36. Jessop, D.M.; Adams, S.E.; Willighagen, E.L.; Hawizy, L.; Murray-Rust, P. OSCAR4: A Flexible Architecture for Chemical Text-Mining. *Journal of Cheminformatics* **2011**, *3*, 41, doi:10.1186/1758-2946-3-41.
37. Kamijo, H.; Morii, S.; Yamaguchi, W.; Toyooka, N.; Tada-Umezaki, M.; Hirobayashi, S. Creating an Adaptive Technology Using a Cheminformatics System To Read Aloud Chemical Compound Names for People with Visual Disabilities. *J. Chem. Educ.* **2016**, *93*, 496–503, doi:10.1021/acs.jchemed.5b00217.
38. Khoshouie, E.; Ayub, A.F.M.; Mesrinejad, F. Molecular Workbench Software as Computer Assisted Instruction to Aid the Learning of Chemistry. *Journal of Educational and Social Research* **2014**, *4*, 373.
39. Kiener, J. Molecule Database Framework: A Framework for Creating Database Applications with Chemical Structure Search Capability. *J Cheminform* **2013**, *5*, 48, doi:10.1186/1758-2946-5-48.
40. Kochev, N.; Avramova, S.; Jeliazkova, N. Ambit-SMIRKS: A Software Module for Reaction Representation, Reaction Search and Structure Transformation. *Journal of Cheminformatics* **2018**, *10*, 42, doi:10.1186/s13321-018-0295-6.
41. Kochev, N.T.; Paskaleva, V.H.; Jeliazkova, N. Ambit-Tautomer: An Open Source Tool for Tautomer Generation. *Molecular Informatics* **2013**, *32*, 481–504, doi:10.1002/minf.201200133.
42. Kolšek, K.; Mavri, J.; Sollner Dolenc, M.; Gobec, S.; Turk, S. Endocrine Disruptome—an Open Source Prediction Tool for Assessing Endocrine Disruption Potential through Nuclear Receptor Binding. *J Chem Inf Model* **2014**, *54*, 1254–1267, doi:10.1021/ci400649p.
43. Korf, A.; Jeck, V.; Schmid, R.; Helmer, P.O.; Hayen, H. Lipid Species Annotation at Double Bond Position Level with Custom Databases by Extension of the MZmine 2 Open-Source Software Package. *Anal. Chem.* **2019**, *91*, 5098–5105, doi:10.1021/acs.analchem.8b05493.
44. Kratochvíl, M.; Vondrášek, J.; Galgonek, J. Interoperable Chemical Structure Search Service. *Journal of Cheminformatics* **2019**, *11*, 45, doi:10.1186/s13321-019-0367-2.
45. Kujawski, J.; Bernard, M.K.; Janusz, A.; Kuźma, W. Prediction of Log P: ALOGPS Application in Medicinal Chemistry Education. *J. Chem. Educ.* **2012**, *89*, 64–67, doi:10.1021/ed100444h.
46. Lamprecht, M.R.; Sabatini, D.M.; Carpenter, A.E. CellProfiler: Free, Versatile Software for Automated Biological Image Analysis. *Biotechniques* **2007**, *42*, 71–75, doi:10.2144/000112257.
47. Lätti, S.; Niinivehmas, S.; Pentikäinen, O.T. Rocker: Open Source, Easy-to-Use Tool for AUC and Enrichment Calculations and ROC Visualization. *Journal of Cheminformatics* **2016**, *8*, 45, doi:10.1186/s13321-016-0158-y.
48. Lawson, K.R.; Lawson, J. LICSS—A Chemical Spreadsheet in Microsoft Excel. *Journal of Cheminformatics* **2012**, *4*, 3, doi:10.1186/1758-2946-4-3.
49. Lee, M.-L.; Aliagas, I.; Feng, J.A.; Gabriel, T.; O'Donnell, T.J.; Sellers, B.D.; Wiswedel, B.; Gobbi, A. Chemalot and Chemalot\_knime: Command Line Programs as Workflow Tools for Drug Discovery. *Journal of Cheminformatics* **2017**, *9*, 38, doi:10.1186/s13321-017-0228-9.
50. Leguy, J.; Cauchy, T.; Glavatskikh, M.; Duval, B.; Da Mota, B. EvoMol: A Flexible and Interpretable Evolutionary Algorithm for Unbiased de Novo Molecular Generation. *Journal of Cheminformatics* **2020**, *12*, 55, doi:10.1186/s13321-020-00458-z.
51. Liang, L.; Ma, C.; Du, T.; Zhao, Y.; Zhao, X.; Liu, M.; Wang, Z.; Lin, J. Bioactivity-Explorer: A Web Application for Interactive Visualization and Exploration of Bioactivity Data. *Journal of Cheminformatics* **2019**, *11*, 47, doi:10.1186/s13321-019-0370-7.
52. López-Fernández, H.; de S. Pessôa, G.; Arruda, M.A.Z.; Capelo-Martínez, J.L.; Fdez-Riverola, F.; Glez-Peña, D.; Reboiro-Jato, M. LA-ImageS: A Software for Elemental Distribution Bioimaging Using LA-ICP-MS Data. *Journal of Cheminformatics* **2016**, *8*, 65, doi:10.1186/s13321-016-0178-7.
53. Monge, A.; Arrault, A.; Marot, C.; Morin-Allory, L. Managing, Profiling and Analyzing a Library of 2.6 Million Compounds Gathered from 32 Chemical Providers. *Mol Divers* **2006**, *10*, 389–403, doi:10.1007/s11030-006-9033-5.

54. Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T. Mordred: A Molecular Descriptor Calculator. *Journal of Cheminformatics* **2018**, *10*, 4, doi:10.1186/s13321-018-0258-y.
55. Murrell, D.S.; Cortes-Ciriano, I.; van Westen, G.J.P.; Stott, I.P.; Bender, A.; Malliavin, T.E.; Glen, R.C. Chemically Aware Model Builder (Camb): An R Package for Property and Bioactivity Modelling of Small Molecules. *Journal of Cheminformatics* **2015**, *7*, 45, doi:10.1186/s13321-015-0086-2.
56. O'Boyle, N.M.; Banck, M.; James, C.A.; Morley, C.; Vandermeersch, T.; Hutchison, G.R. Open Babel: An Open Chemical Toolbox. *Journal of Cheminformatics* **2011**, *3*, 33, doi:10.1186/1758-2946-3-33.
57. O'Boyle, N.M.; Hutchison, G.R. Cinfony—Combining Open Source Cheminformatics Toolkits behind a Common Interface. *Chemistry Central Journal* **2008**, *2*, 24, doi:10.1186/1752-153X-2-24.
58. O'Boyle, N.M.; Morley, C.; Hutchison, G.R. Pybel: A Python Wrapper for the OpenBabel Cheminformatics Toolkit. *Chemistry Central Journal* **2008**, *2*, 5, doi:10.1186/1752-153X-2-5.
59. Patlewicz, G.; Jeliaskova, N.; Safford, R.J.; Worth, A.P.; Aleksiev, B. An Evaluation of the Implementation of the Cramer Classification Scheme in the Toxtree Software. *SAR and QSAR in Environmental Research* **2008**, *19*, 495–524, doi:10.1080/10629360802083871.
60. Pavlov, D.; Rybalkin, M.; Karulin, B.; Kozhevnikov, M.; Savelyev, A.; Churinov, A. Indigo: Universal Cheminformatics API. *Journal of Cheminformatics* **2011**, *3*, P4, doi:10.1186/1758-2946-3-S1-P4.
61. Peironcelly, J.E.; Rojas-Chertó, M.; Fichera, D.; Reijmers, T.; Coulier, L.; Faulon, J.-L.; Hankemeier, T. OMG: Open Molecule Generator. *Journal of Cheminformatics* **2012**, *4*, 21, doi:10.1186/1758-2946-4-21.
62. Perna, J. Edumol: Avoim ja Ilmainen Molekyyylimallinnusovellus Kemian Opetuksen Tueksi. *LUMAT: International Journal on Math, Science and Technology Education* **2015**, *3*, 960–975, doi:10.31129/lumat.v3i7.977.
63. Puranen, J.S.; Vainio, M.J.; Johnson, M.S. Accurate Conformation-Dependent Molecular Electrostatic Potentials for High-Throughput in Silico Drug Discovery. *Journal of Computational Chemistry* **2010**, *31*, 1722–1732, doi:10.1002/jcc.21460.
64. Rahman, S.A.; Bashton, M.; Holliday, G.L.; Schrader, R.; Thornton, J.M. Small Molecule Subgraph Detector (SMSD) Toolkit. *Journal of Cheminformatics* **2009**, *1*, 12, doi:10.1186/1758-2946-1-12.
65. Rijnbeek, M.; Steinbeck, C. OrChem—An Open Source Chemistry Search Engine for Oracle®. *Journal of Cheminformatics* **2009**, *1*, 17, doi:10.1186/1758-2946-1-17.
66. Ropp, P.J.; Kaminsky, J.C.; Yablonski, S.; Durrant, J.D. Dimorphite-DL: An Open-Source Program for Enumerating the Ionization States of Drug-like Small Molecules. *Journal of Cheminformatics* **2019**, *11*, 14, doi:10.1186/s13321-019-0336-9.
67. Ropp, P.J.; Spiegel, J.O.; Walker, J.L.; Green, H.; Morales, G.A.; Milliken, K.A.; Ringe, J.J.; Durrant, J.D. Gypsum-DL: An Open-Source Program for Preparing Small-Molecule Libraries for Structure-Based Virtual Screening. *Journal of Cheminformatics* **2019**, *11*, 34, doi:10.1186/s13321-019-0358-3.
68. Rosen, J.; Miguet, L.; Pérez, S. Shape: Automatic Conformation Prediction of Carbohydrates Using a Genetic Algorithm. *J Cheminform* **2009**, *1*, 16, doi:10.1186/1758-2946-1-16.
69. Salentin, S.; Schreiber, S.; Haupt, V.J.; Adasme, M.F.; Schroeder, M. PLIP: Fully Automated Protein–Ligand Interaction Profiler. *Nucleic Acids Res* **2015**, *43*, W443–W447, doi:10.1093/nar/gkv315.
70. Sander, T.; Freyss, J.; von Korff, M.; Rufener, C. DataWarrior: An Open-Source Program For Chemistry Aware Data Visualization And Analysis. *J. Chem. Inf. Model.* **2015**, *55*, 460–473, doi:10.1021/ci500588j.
71. Scalfani, V.F.; Williams, A.J.; Tkachenko, V.; Karapetyan, K.; Pshenichnov, A.; Hanson, R.M.; Liddie, J.M.; Bara, J.E. Programmatic Conversion of Crystal Structures into 3D Printable Files Using Jmol. *Journal of Cheminformatics* **2016**, *8*, 66, doi:10.1186/s13321-016-0181-z.

72. Smith, R.; Williamson, R.; Ventura, D.; Prince, J.T. Rubabel: Wrapping Open Babel with Ruby. *Journal of Cheminformatics* **2013**, *5*, 35, doi:10.1186/1758-2946-5-35.
73. Stålring, J.C.; Carlsson, L.A.; Almeida, P.; Boyer, S. AZOrange—High Performance Open Source Machine Learning for QSAR Modeling in a Graphical Programming Environment. *Journal of Cheminformatics* **2011**, *3*, 28, doi:10.1186/1758-2946-3-28.
74. Steinbeck, C. SENECA: A Platform-Independent, Distributed, and Parallel System for Computer-Assisted Structure Elucidation in Organic Chemistry. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1500–1507, doi:10.1021/ci000407n.
75. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500, doi:10.1021/ci025584y.
76. Sud, M. MayaChemTools: An Open Source Package for Computational Drug Discovery. *J. Chem. Inf. Model.* **2016**, *56*, 2292–2297, doi:10.1021/acs.jcim.6b00505.
77. Sydow, D.; Morger, A.; Driller, M.; Volkamer, A. TeachOpenCADD: A Teaching Platform for Computer-Aided Drug Design Using Open Source Packages and Data. *Journal of Cheminformatics* **2019**, *11*, 29, doi:10.1186/s13321-019-0351-x.
78. Sykora, V.J.; Leahy, D.E. Chemical Descriptors Library (CDL): A Generic, Open Source Software Library for Chemical Informatics. *J. Chem. Inf. Model.* **2008**, *48*, 1931–1942, doi:10.1021/ci800135h.
79. Tcheremenskaia, O.; Benigni, R.; Nikolova, I.; Jeliaskova, N.; Escher, S.E.; Batke, M.; Baier, T.; Poroikov, V.; Lagunin, A.; Rautenberg, M.; et al. OpenTox Predictive Toxicology Framework: Toxicological Ontology and Semantic Media Wiki-Based OpenToxipedia. *J. Biomed Semantics* **2012**, *3* Suppl 1, S7, doi:10.1186/2041-1480-3-S1-S7.
80. Tosco, P.; Balle, T.; Shiri, F. Open3DALIGN: An Open-Source Software Aimed at Unsupervised Ligand Alignment. *J. Comput Aided Mol Des* **2011**, *25*, 777–783, doi:10.1007/s10822-011-9462-9.
81. Tremouilhac, P.; Nguyen, A.; Huang, Y.-C.; Kotov, S.; Lütjohann, D.S.; Hübsch, F.; Jung, N.; Bräse, S. Chemotion ELN: An Open Source Electronic Lab Notebook for Chemists in Academia. *Journal of Cheminformatics* **2017**, *9*, 54, doi:10.1186/s13321-017-0240-0.
82. Valdés-Martini, J.R.; Marrero-Ponce, Y.; García-Jacas, C.R.; Martínez-Mayorga, K.; Barigye, S.J.; Vaz d’Almeida, Y.S.; Pham-The, H.; Pérez-Giménez, F.; Morell, C.A. QuBiLS-MAS, Open Source Multi-Platform Software for Atom- and Bond-Based Topological (2D) and Chiral (2.5D) Algebraic Molecular Descriptors Computations. *J. Cheminform* **2017**, *9*, 35, doi:10.1186/s13321-017-0211-5.
83. Willighagen, E.L.; Mayfield, J.W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliaskova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; et al. The Chemistry Development Kit (CDK) v2.0: Atom Typing, Depiction, Molecular Formulas, and Substructure Searching. *Journal of Cheminformatics* **2017**, *9*, 33, doi:10.1186/s13321-017-0220-4.
84. Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): A New Open-Source Player in the Drug Discovery Field. *Journal of Cheminformatics* **2015**, *7*, 26, doi:10.1186/s13321-015-0078-2.
85. Xia, B.; Tai, Z.-F.; Gu, Y.-C.; Li, B.-J.; Ding, L.-S.; Zhou, Y. MyMolDB: A Micromolecular Database Solution with Open Source and Free Components. *Journal of Computational Chemistry* **2011**, *32*, 2942–2948, doi:10.1002/jcc.21874.
86. Yap, C.W. PaDEL-Descriptor: An Open Source Software to Calculate Molecular Descriptors and Fingerprints. *J. Comput Chem* **2011**, *32*, 1466–1474, doi:10.1002/jcc.21707.
87. Zhu, Q.; Lajiness, M.S.; Ding, Y.; Wild, D.J. WENDI: A Tool for Finding Non-Obvious Relationships between Compounds and Biological Properties, Genes, Diseases and Scholarly Publications. *Journal of Cheminformatics* **2010**, *2*, 6, doi:10.1186/1758-2946-2-6.

### Appendix B. Comprehensive List of All Main and Sub-Categories

#	Main Category	Sub-Category
1	Develop New Software (completely new software, features, tools, or processes)	Cross-knowledge domain solutions
		Cross-platform solutions
		Databases are limited
		Decrease the black-box effect
		Difficult to use
		Editing capabilities
		Expensive relational database system
		Facilitates compilation
		Facilitates maintenance
		First MOPAC optimization solution
		Fix software bugs
		Free library management and screening software
		High costs
		Import and export tools
		Improve performance
		Insufficient update frequencies
		Interconversion of chemical structures
		Lack of database creator software
		Limited database access
		Limited number of R solutions
		Local installation to avoid online solution
		Loop support
		Manual work processes
		Mobile application
		Mobile-compatible solutions
		Module that combines multiple toolkits
		Need for supervision
		New architecture
		No suitable open-source solutions available
		No suitable solutions available
		Object-oriented suite
		Open molecular descriptor platform for macromolecules
		Open-source solutions
		Poor modularity
		Practical solutions
		Programmatic file conversion
		Reduce costs
		Reduce duplication of work
		Reduce errors
		Restrictive licenses
		Scoring functions missing from OS solutions
		Sensitive data
		Visualization tools
		Web-based solutions
		Workflow based on open file formats
2	Update Current Features, Tools, or Processes	Access to experimental data
		Apply a new algorithm
		Automated affinity prediction
		Correct an error from the previous version
		Elaboration of protein motion
		Ensure stability
		Improve performance
		Improved animation workflow
		Improved machine learning model
		Improved workflow
		Improves defects present in available solutions
		Methods for searching chemical space efficiently
		Multiple new features or improvements
		Multiple types of data sources
		New data sources added
		New force field
		New operating logic
		Software update

#	Main Category	Sub-Category
3	Improve Usability	Clear workflow
		Command line interface
		Easy to install
		Enable machine learning without programming
		Faster calculations
		Interactive instructions
		Language versions
		Optimize speed and accuracy
		Ready-made scripts
		Reduce functional complexity
		Reduce the need for programming skills
		Reduced time
		Systematic work practices
		User-friendly graphical interface
		User-friendly operating logic
4	Support Open-Source Development and Open Science	User-friendly search tools
		User-friendly web interface
		Allow integration to commercial software
		Code transparency
		CSS examples
		Develop open alternatives
		Free software
		HTML5 technology
		JavaScript frameworks
		Lack of extensibility
		MySQL database
		Open data
		Open license
		Open solution
		Open-source SMIRKS package
		Open technology or framework for developers
		Policies for data management
		Possibility to publish research results open
		Ruby framework
		SMIRKS specification
		Standards for computational processes
		Standards for data management
		Standards for software development
		Starting point for database integration
		Support cooperation between academia and industry
5	Fulfill Chemical Information Needs	Support QSAR standards
		Support sustainable science via OS
		Supports integration
		Tested technology for developers
		Ability to process larger molecules
		Analyze relationships and patterns
		Analyze the relevance of search results
		Bioactivity data are not extracted
		Calibrated conformation-dependent molecular electrostatic potentials
		Chemical ontologies
		Combined data visualizations
		Comprehensive information mining from public bioactivity databases
		Curated data
		Describe objects and their relationships
		Efficient and reliable in silico PD-PK-T prediction methods
		Expansion to other areas of chemistry
		Facilitate extensive drug molecule studies
		Generate MCS and rank solutions via multiple variables
		Generation of tautomeric forms
		Handle crystal structures
		Identify reliable data
		Improved descriptor handling
		Model motion from structural data
		More detailed algorithm description
		More time to solve chemistry



#	Main Category	Sub-Category
5	Fulfill Chemical Information Needs	No need for prior dataset
		Online prediction of Log P
		Process ADMET models
		QSAR model visualization tools
		QSAR predictions on the production environment
		RDF information model
		Reach large audience
		Reliable method for predicting endocrine disruption
		Representation of surface physico-chemical properties of proteins
		Screen multiple targets at once
		Solutions for experimental data management
		Specific calculation method
		Specification of SMILES and SMARTS dialects
		Structural relations
		Support chemical decision making
		Support for multiple vs. analysis methods
		Support phenotypic screening
6	Support Chemistry Learning and Teaching	Support recursive atom expressions
		Systematic discovery of chemical space
		Visual validation of models
		Visualize dynamical forces on intermolecular interactions
		Implement AR into chemistry teaching
		Support chemical reading
		Support chemistry teaching and learning
		Support learning of the structure–property relationship
		Support spatial learning
		Text to speech

## References

- Chen, W.L. Chemoinformatics: Past, Present, and Future. *J. Chem. Inf. Model.* **2006**, *46*, 2230–2255. [\[CrossRef\]](#) [\[PubMed\]](#)
- King, G.W.; Cross, P.C.; Thomas, G.B. The Asymmetric Rotor III. Punched-Card Methods of Constructing Band Spectra. *J. Chem. Phys.* **1946**, *14*, 35–42. [\[CrossRef\]](#)
- Ray, L.C.; Kirsch, R.A. Finding Chemical Records by Digital Computers. *Science* **1957**, *126*, 814–819. [\[CrossRef\]](#) [\[PubMed\]](#)
- Willett, P. Chemoinformatics: A History. *WIREs Comput. Mol. Sci.* **2011**, *1*, 46–56. [\[CrossRef\]](#)
- Willett, P. The Literature of Chemoinformatics: 1978–2018. *IJMS* **2020**, *21*, 5576. [\[CrossRef\]](#)
- Brown, N. Chemoinformatics—An Introduction for Computer Scientists. *ACM Comput. Surv.* **2009**, *41*, 1–38. [\[CrossRef\]](#)
- Lehtola, S.; Karttunen, A.J. Free and Open Source Software for Computational Chemistry Education. *WIREs Comput. Mol. Sci.* **2022**, *12*, e1610. [\[CrossRef\]](#)
- Wegner, J.K.; Sterling, A.; Guha, R.; Bender, A.; Faulon, J.-L.; Hastings, J.; O’Boyle, N.; Overington, J.; Van Vlijmen, H.; Willighagen, E. Cheminformatics. *Commun. ACM* **2012**, *55*, 65–75. [\[CrossRef\]](#)
- Gezelter, J.D. Open Source and Open Data Should Be Standard Practices. *J. Phys. Chem. Lett.* **2015**, *6*, 1168–1169. [\[CrossRef\]](#)
- O’Boyle, N.M.; Banck, M.; James, C.A.; Morley, C.; Vandermeersch, T.; Hutchison, G.R. Open Babel: An Open Chemical Toolbox. *J. Cheminform.* **2011**, *3*, 33. [\[CrossRef\]](#)
- Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*, 2nd ed.; Sage: Thousand Oaks, CA, USA, 2004; ISBN 978-0-7619-1544-7.
- Cohen, L.; Manion, L.; Morrison, K. *Research Methods in Education*, 6th ed.; Routledge: London, UK; New York, NY, USA, 2007; ISBN 978-0-415-37410-1.
- Alenezi, M. Internal Quality Evolution of Open-Source Software Systems. *Appl. Sci.* **2021**, *11*, 5690. [\[CrossRef\]](#)
- Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying Computational Thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [\[CrossRef\]](#)
- Jirat, J.; Cech, P.; Znamenacek, J.; Simek, M.; Skuta, C.; Vanek, T.; Dibuszova, E.; Nic, M.; Svozil, D. Developing and Implementing a Combined Chemistry and Informatics Curriculum for Undergraduate and Graduate Students in the Czech Republic. *J. Chem. Educ.* **2013**, *90*, 315–319. [\[CrossRef\]](#)
- Kim, S.; Bucholtz, E.C.; Briney, K.; Cornell, A.P.; Cuadros, J.; Fulfer, K.D.; Gupta, T.; Hepler-Smith, E.; Johnston, D.H.; Lang, A.S.I.D.; et al. Teaching Cheminformatics through a Collaborative Intercollegiate Online Chemistry Course (OLCC). *J. Chem. Educ.* **2021**, *98*, 416–425. [\[CrossRef\]](#) [\[PubMed\]](#)
- Stahl, M.T. Open-Source Software: Not Quite Endsville. *Drug Discov. Today* **2005**, *10*, 219–222. [\[CrossRef\]](#)
- Wild, D.J. Grand Challenges for Cheminformatics. *J. Cheminform.* **2009**, *1*, 1. [\[CrossRef\]](#)
- O’Boyle, N.M.; Guha, R.; Willighagen, E.L.; Adams, S.E.; Alvarsson, J.; Bradley, J.-C.; Filippov, I.V.; Hanson, R.M.; Hanwell, M.D.; Hutchison, G.R.; et al. Open Data, Open Source and Open Standards in Chemistry: The Blue Obelisk Five Years on. *J. Cheminform.* **2011**, *3*, 37. [\[CrossRef\]](#)

20. Krylov, A.I.; Herbert, J.M.; Furche, F.; Head-Gordon, M.; Knowles, P.J.; Lindh, R.; Manby, F.R.; Pulay, P.; Skylaris, C.-K.; Werner, H.-J. What Is the Price of Open-Source Software? *J. Phys. Chem. Lett.* **2015**, *6*, 2751–2754. [CrossRef]
21. Jacob, C.R. How Open Is Commercial Scientific Software? *J. Phys. Chem. Lett.* **2016**, *7*, 351–353. [CrossRef]
22. Free Software Foundation, Inc. What Is Free Software? Available online: <https://www.gnu.org/philosophy/free-sw.html> (accessed on 29 September 2021).
23. Opensource.org. The Open Source Definition. Available online: <https://opensource.org/osd> (accessed on 29 September 2021).
24. Lerner, J.; Pathak, P.A.; Tirole, J. The Dynamics of Open-Source Contributors. *Am. Econ. Rev.* **2006**, *96*, 114–118. [CrossRef]
25. Hars, A.; Ou, S. Working for Free? Motivations for Participating in Open-Source Projects. *Int. J. Electron. Commer.* **2002**, *6*, 25–39.
26. Bitzer, J.; Schrettl, W.; Schröder, P.J.H. Intrinsic Motivation in Open Source Software Development. *J. Comp. Econ.* **2007**, *35*, 160–169. [CrossRef]
27. Pereira, J. Leveraging Final Degree Projects for Open Source Software Contributions. *Electronics* **2021**, *10*, 1181. [CrossRef]
28. President's Information Technology Advisory Committee. *Computational Science: Ensuring America's Competitiveness*; National Coordination Office for Information Technology Research and Development: Washington, DC, USA, 2005; p. 117.
29. Blanton, B.; Lenhardt, W.C. A Scientist's Perspective on Sustainable Scientific Software. *J. Open Res. Softw.* **2014**, *2*, e17. [CrossRef]
30. Koponen, T. Life Cycle of Defects in Open Source Software Projects. In *Proceedings of the Open Source Systems*; Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, M., Succi, G., Eds.; Springer: Boston, MA, USA, 2006; pp. 195–200.
31. Johnson, J.P. Collaboration, Peer Review and Open Source Software. *Inf. Econ. Policy* **2006**, *18*, 477–497. [CrossRef]
32. Saini, M.; Chahal, K. A Review of Open Source Software Development Life Cycle Models. *Int. J. Softw. Eng. Appl.* **2014**, *8*, 417–434. [CrossRef]
33. Hauschild, A.-C.; Eick, L.; Wienbeck, J.; Heider, D. Fostering Reproducibility, Reusability, and Technology Transfer in Health Informatics. *iScience* **2021**, *24*, 102803. [CrossRef]
34. Swarts, J. Open-Source Software in the Sciences: The Challenge of User Support. *J. Bus. Tech. Commun.* **2019**, *33*, 60–90. [CrossRef]
35. Hauschild, A.-C.; Martin, R.; Holst, S.C.; Wienbeck, J.; Heider, D. Guideline for Software Life Cycle in Health Informatics. *iScience* **2022**, *25*, 105534. [CrossRef]
36. Theisen, K.J. Programming Languages in Chemistry: A Review of HTML5/JavaScript. *J. Cheminform.* **2019**, *11*, 11. [CrossRef]
37. ATLAS.ti Scientific Software Development GmbH. *ATLAS.Ti 9 Software*; ATLAS.ti Scientific Software Development GmbH: Berlin, Germany, 2021.
38. McHugh, M.L. Interrater Reliability: The Kappa Statistic. *Biochem. Med.* **2012**, *22*, 276–282. [CrossRef]
39. Peironecely, J.E.; Rojas-Chertó, M.; Fichera, D.; Reijmers, T.; Coulier, L.; Faulon, J.-L.; Hankemeier, T. OMG: Open Molecule Generator. *J. Cheminform.* **2012**, *4*, 21. [CrossRef] [PubMed]
40. Cao, D.-S.; Xu, Q.-S.; Hu, Q.-N.; Liang, Y.-Z. ChemoPy: Freely Available Python Package for Computational Biology and Chemoinformatics. *Bioinformatics* **2013**, *29*, 1092–1094. [CrossRef]
41. Chen, P.; Wang, Y.; Yan, H.; Gao, S.; Xu, Z.; Li, Y.; Mo, Q.; Huang, J.; Tao, J.; Pan, G.; et al. 3DStructGen: An Interactive Web-Based 3D Structure Generation for Non-Periodic Molecule and Crystal. *J. Cheminform.* **2020**, *12*, 7. [CrossRef] [PubMed]
42. Kiener, J. Molecule Database Framework: A Framework for Creating Database Applications with Chemical Structure Search Capability. *J. Cheminform.* **2013**, *5*, 48. [CrossRef]
43. Hanwell, M.D.; Curtis, D.E.; Lonie, D.C.; Vandermeersch, T.; Zurek, E.; Hutchison, G.R. Avogadro: An Advanced Semantic Chemical Editor, Visualization, and Analysis Platform. *J. Cheminform.* **2012**, *4*, 17. [CrossRef]
44. Hanson, R.M. Jmol SMILES and Jmol SMARTS: Specifications and Applications. *J. Cheminform.* **2016**, *8*, 50. [CrossRef]
45. Patlewicz, G.; Jeliazkova, N.; Safford, R.J.; Worth, A.P.; Aleksiev, B. An Evaluation of the Implementation of the Cramer Classification Scheme in the ToxTree Software. *SAR QSAR Environ. Res.* **2008**, *19*, 495–524. [CrossRef]
46. Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T. Mordred: A Molecular Descriptor Calculator. *J. Cheminform.* **2018**, *10*, 4. [CrossRef]
47. Carrió, P.; López, O.; Sanz, F.; Pastor, M. ETOXlab, an Open Source Modeling Framework for Implementing Predictive Models in Production Environments. *J. Cheminform.* **2015**, *7*, 8. [CrossRef]
48. Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A.P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L.J.; Cibrián-Uhalte, E.; et al. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945–D954. [CrossRef] [PubMed]
49. Fritsch, S.; Neumann, S.; Schaub, J.; Steinbeck, C.; Zielesny, A. ErtlFunctionalGroupsFinder: Automated Rule-Based Functional Group Detection with the Chemistry Development Kit (CDK). *J. Cheminform.* **2019**, *11*, 37. [CrossRef] [PubMed]
50. Bento, A.P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L.J.; De Veij, M.; Leach, A.R. An Open Source Chemical Structure Curation Pipeline Using RDKit. *J. Cheminform.* **2020**, *12*, 51. [CrossRef]
51. Gütlein, M.; Karwath, A.; Kramer, S. CheS-Mapper 2.0 for Visual Validation of (Q)SAR Models. *J. Cheminform.* **2014**, *6*, 41. [CrossRef]
52. Smith, R.; Williamson, R.; Ventura, D.; Prince, J.T. Rubabel: Wrapping Open Babel with Ruby. *J. Cheminform.* **2013**, *5*, 35. [CrossRef]
53. Sud, M. MayaChemTools: An Open Source Package for Computational Drug Discovery. *J. Chem. Inf. Model.* **2016**, *56*, 2292–2297. [CrossRef]
54. Kochev, N.; Avramova, S.; Jeliazkova, N. Ambit-SMIRKS: A Software Module for Reaction Representation, Reaction Search and Structure Transformation. *J. Cheminform.* **2018**, *10*, 42. [CrossRef]

55. Tremouilhac, P.; Nguyen, A.; Huang, Y.-C.; Kotov, S.; Lütjohann, D.S.; Hübsch, F.; Jung, N.; Bräse, S. Chemotion ELN: An Open Source Electronic Lab Notebook for Chemists in Academia. *J. Cheminform.* **2017**, *9*, 54. [[CrossRef](#)] [[PubMed](#)]
56. Ekins, S.; Clark, A.M.; Sarker, M. TB Mobile: A Mobile App for Anti-Tuberculosis Molecules with Known Targets. *J. Cheminform.* **2013**, *5*, 13. [[CrossRef](#)]
57. Sykora, V.J.; Leahy, D.E. Chemical Descriptors Library (CDL): A Generic, Open Source Software Library for Chemical Informatics. *J. Chem. Inf. Model.* **2008**, *48*, 1931–1942. [[CrossRef](#)]
58. Guilloux, V.L.; Arrault, A.; Colliandre, L.; Bourg, S.; Vayer, P.; Morin-Allory, L. Mining Collections of Compounds with Screening Assistant 2. *J. Cheminform.* **2012**, *4*, 20. [[CrossRef](#)] [[PubMed](#)]
59. Stålring, J.C.; Carlsson, L.A.; Almeida, P.; Boyer, S. AZOrange—High Performance Open Source Machine Learning for QSAR Modeling in a Graphical Programming Environment. *J. Cheminform.* **2011**, *3*, 28. [[CrossRef](#)] [[PubMed](#)]
60. Fatemah, A.; Rasool, S.; Habib, U. Interactive 3D Visualization of Chemical Structure Diagrams Embedded in Text to Aid Spatial Learning Process of Students. *J. Chem. Educ.* **2020**, *97*, 992–1000. [[CrossRef](#)]
61. Jensen, J.H.; Kromann, J.C. The Molecule Calculator: A Web Application for Fast Quantum Mechanics-Based Estimation of Molecular Properties. *J. Chem. Educ.* **2013**, *90*, 1093–1095. [[CrossRef](#)]
62. Kamijo, H.; Morii, S.; Yamaguchi, W.; Toyooka, N.; Tada-Umezaki, M.; Hirobayashi, S. Creating an Adaptive Technology Using a Cheminformatics System To Read Aloud Chemical Compound Names for People with Visual Disabilities. *J. Chem. Educ.* **2016**, *93*, 496–503. [[CrossRef](#)]
63. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500. [[CrossRef](#)]
64. Bauer, M.A.; Berleant, D.; Cornell, A.P.; Belford, R.E. WikiHyperGlossary (WHG): An Information Literacy Technology for Chemistry Documents. *J. Cheminform.* **2015**, *7*, 22. [[CrossRef](#)]
65. Bergwerf, H. MolView: An Attempt to Get the Cloud into Chemistry Classrooms. *DivCHED CCCE Comm. Comput. Chem. Educ.* **2015**, *9*, 1–9.
66. Wild, D.J. Cheminformatics for the Masses: A Chance to Increase Educational Opportunities for the next Generation of Cheminformaticians. *J. Cheminform.* **2013**, *5*, 32. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.