Title Page

Abstract

Contents

[Introduction 4](#_Toc136335055)

[Literature review 5](#_Toc136335056)

[Overview of SMILES notation 5](#_Toc136335057)

[Methodology 6](#_Toc136335058)

[Molecular graph construction 6](#_Toc136335059)

[SMILES validation 7](#_Toc136335060)

[Substring occurrences counting 8](#_Toc136335061)

[Molecular formula extraction 9](#_Toc136335062)

[Limitations and Challenges 10](#_Toc136335063)

[Results and Discussion 11](#_Toc136335064)

[Implementation details 12](#_Toc136335065)

[Program Description 13](#_Toc136335066)

[Appendix 14](#_Toc136335067)

[References 15](#_Toc136335068)

# Introduction

Chemoinformatics, a field at the intersection of chemistry and computer science, deals with chemical information storage, processing, and analysis. Chemical compounds are fundamental entities in chemistry, and treating information about them plays a crucial role in various fields, including drug discovery, material science, and environmental studies. To represent and explore these compounds effectively, a standardized notation called Simplified Molecular Input Line Entry System (SMILES) has been developed [1].

SMILES notation provides a compact and human-readable linear string representation of chemical structures. It has gained immense popularity in the field of chemoinformatics due to its simplicity and compatibility with computational algorithms. SMILES notation represents atoms using atomic symbols and bonds between atoms are denoted by various symbols. It also accommodates stereochemistry, isotopes, charges, and other structural features. The concise nature of SMILES notation allows for the efficient storage and exchange of chemical information, making it a preferred choice in chemical databases and computational workflows.

The primary objective of this project was to create an efficient and accurate tool for SMILES processing and analysis. We aimed to develop a program that could process SMILES strings, construct molecular graph representations, and by that, validate their structure, extract molecular formulas, and perform substring occurrence counting.

Validating the SMILES strings includes ensuring that they adhere to the syntax and rules of the SMILES notation and that corresponding molecules obey basic chemical rules of bond formation. SMILES validation contributes to data quality control and data integration efforts. By enforcing validation checks, researchers can identify and eliminate inconsistencies, discrepancies, or outliers in large chemical databases. This enhances the reliability of datasets, making them more suitable for systematic analysis and comparison. Validated SMILES also facilitate data exchange and interoperability between different software systems, allowing seamless integration and collaboration among researchers.

Throughout this report, we will provide a detailed account of our methodology, implementation details, and the results obtained from our program.

By developing this comprehensive SMILES processing and analysis program, we aim to contribute to the field of computational chemistry and enable researchers to extract valuable insights from chemical compounds. This project serves as a stepping stone toward advancing our understanding of molecular structures and their implications in various scientific disciplines.

# Literature review

## Overview of SMILES notation

SMILES (Simplified Molecular Input Line Entry System) notation is a widely used linear string representation for describing chemical structures. It provides a concise and human-readable format that can be easily stored, processed, and exchanged across various computational platforms. The development of SMILES notation can be attributed to Weininger and colleagues, who introduced it in the late 1980s as a means to encode chemical structures in a simple and unambiguous manner (Weininger, 1988). Since its inception, SMILES notation has become an integral part of chemoinformatics and is widely utilized in diverse applications such as drug discovery, chemical informatics, and computational chemistry.

SMILES notation follows a set of rules to represent chemical structures. At its core, SMILES represents atoms using atomic symbols, and bonds between atoms are denoted by various symbols. For example, carbon is represented by "C," nitrogen by "N," and so on. Single bonds are implicit and not explicitly shown, while double bonds are represented by "=", triple bonds by "#", and aromatic bonds by lowercase letters such as "c" or "n". Additionally, parentheses "()" are used to group atoms, and branches are represented by using a branching symbol, such as "(", ")", or "[]". The SMILES notation also accounts for stereochemistry, isotopes, and charges, providing a comprehensive representation of chemical structures.

The importance of SMILES notation lies in its ability to compactly encode complex chemical structures while maintaining structural and chemical information. It allows for the rapid exchange of chemical data, facilitating efficient database searching, similarity analysis, and structure-based modeling (Weininger et al., 1989). SMILES notation serves as a standard representation for chemical structures in chemoinformatics and computational chemistry, enabling seamless integration of diverse computational tools and methods. Its simplicity and conciseness make it a valuable tool for researchers working with large chemical databases and exploring chemical spaces.

SMILES notation has been extensively adopted and integrated into various software libraries and platforms, further solidifying its importance in chemoinformatics. Software packages such as RDKit (Landrum, 2016) and Open Babel (O'Boyle et al., 2011) provide robust implementations of SMILES parsing, generation, and manipulation, allowing researchers to effectively leverage the power of SMILES notation in their computational workflows. The ubiquity of SMILES notation in chemoinformatics underscores its significance as a standardized and versatile representation for chemical structures.

In conclusion, SMILES notation has revolutionized the field of chemoinformatics by providing a compact, human-readable, and standardized representation for chemical structures. Its development and widespread adoption have enabled efficient storage, retrieval, and analysis of chemical information. By adhering to specific rules and conventions, SMILES notation captures the essential features of chemical structures, facilitating seamless integration into computational workflows and supporting diverse applications in drug discovery, chemical informatics, and beyond.

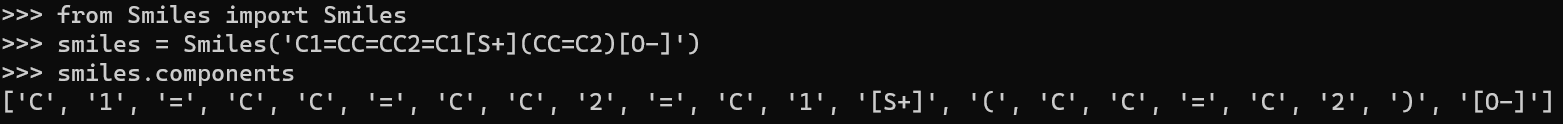
# **Methodology**

## **Molecular graph construction**

The construction of a molecular graph object from a SMILES string in our program involves a two-step process. This process ensures the accurate representation of the chemical structure and the relationships between atoms within the molecule.

In the first step, we break down the SMILES string into its individual components. Components are defined as groups of characters that represent either individual atoms or specific patterns within the SMILES notation. Most components consist of single characters, which represent the atomic symbols, bonds, branch opening or closing, etc. However, there are certain cases where multiple characters need to be treated as a single entity.

For example, atoms enclosed in square brackets are considered as a single component. Additionally, character '%' in SMILES doesn’t have personal meaning but shows that the next two numbers in the string must be treated as one component (two-digit cycle marker).



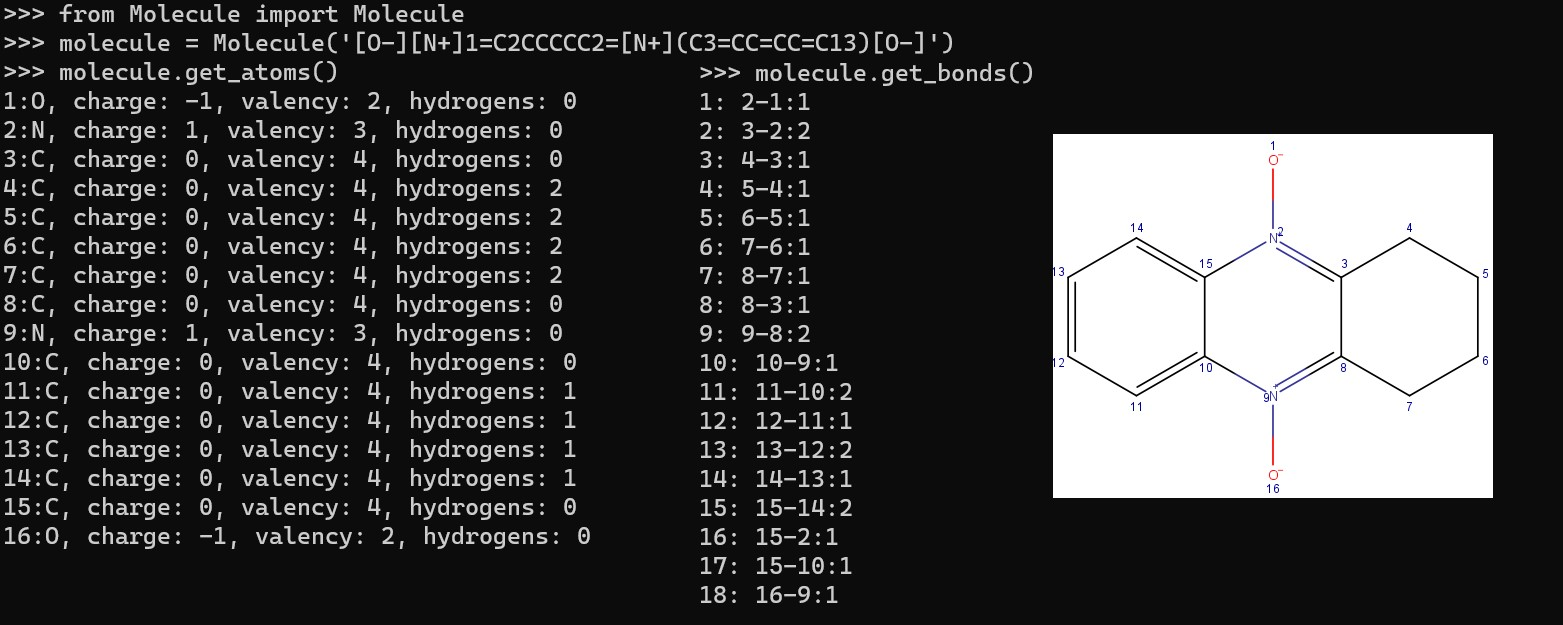
**Picture 1.** Example of breaking a SMILES string into components.

In the second step, we utilize the list of components obtained from the first step to build the molecular graph. The molecular graph is represented by a Structure object consisting of two dictionaries: one representing the atoms in the molecule and the other representing the bonds between atoms.

The dictionary representing atoms contains key-value pairs, where the keys are numbers of atoms in the molecule (assign from left to right while reading the SMILES), and the values contain Atom objects, providing information about the atom's properties, such as the atomic symbol, aromaticity, mass, and connectivity.

The dictionary representing bonds contains key-value pairs, where the keys are numbers of bonds (assign from left to right while reading the SMILES), and the values contain Bond objects, providing information about bonded atoms and the bond order.

By organizing the atoms and bonds in dictionaries, we establish a clear and structured representation of the molecular graph. This representation allows for efficient querying, manipulation, and analysis of the chemical structure.



**Picture 2.** Example of construction of molecular graph.

## **SMILES validation**

The validation of SMILES strings in our program involves multiple steps to ensure the integrity and correctness of the input. This validation process occurs during the breaking down of the SMILES into components, the construction of the molecular graph object, and the final step of checking atom valencies.

In the first step of validation, we examine the SMILES string during the process of breaking it down into components. Here, we primarily focus on identifying invalid symbols or characters that are not part of the SMILES notation. These could include special characters, numbers, or any other non-standard symbols that are not recognized in the context of SMILES. Additionally, we check for empty SMILES strings, which would indicate an invalid input.

The next step of validation occurs during the construction of the molecular graph object. Here, we carefully examine the relationships between different components to ensure they adhere to the rules of the SMILES notation. For example, the first component in the SMILES string must be an atom, as it represents the starting point of the chemical structure. Subsequently, bonds must be followed by atoms or cycle markers, ensuring a valid connectivity pattern.

During this step, we also verify that the SMILES string is well-formed and consistent. We check for proper balance and nesting of parentheses, ensuring that atoms and bonds are correctly grouped within brackets when necessary. This ensures that the structural information is properly encoded and accurately represented in the resulting molecular graph.

In the final step of validation, we examine the valencies of all the atoms in the molecule. Valencies represent the number of bonds an atom can form and play a crucial role in determining the stability and chemical properties of the molecule. We ensure that each atom in the molecular graph satisfies its defined valency, taking into account the number and types of bonds connected to it.

By conducting these validation steps, we ensure that the SMILES string is free from invalid symbols, follows the syntax and rules of the SMILES notation, and adheres to proper atom valencies. This rigorous validation process helps maintain data integrity, prevents errors in subsequent analyses, and ensures that the resulting molecular graph accurately represents the chemical structure.

Overall, our validation procedure encompasses various checks, including symbol validation, relationship verification, and valency examination. These steps collectively contribute to the robustness and reliability of our program, allowing for accurate representation and analysis of chemical compounds based on SMILES strings.

## **Substring occurrences counting**

Our program includes a feature that allows users to count occurrences of substrings from an external list within a set of loaded SMILES strings. This functionality provides valuable insights into the presence and frequency of specific chemical patterns or motifs within the compounds.

To utilize this feature, users can provide a text file containing the desired SMILES substrings. Each substring represents a specific chemical pattern or substructure of interest. These substrings can range from simple atomic symbols to more complex functional groups or ring systems.

Upon loading the text file with the substrings, the program proceeds to process each SMILES string individually and count the occurrences of each substring within the respective SMILES. The program employs an efficient algorithm to scan through the SMILES strings and identify instances of the target substrings.

For each loaded SMILES string, the program generates a CSV (Comma-Separated Values) file that reports the occurrence count of each substring. The CSV file serves as a convenient format for data storage and analysis, allowing users to easily import the results into spreadsheet software for further manipulation and visualization.

The resulting CSV file consists of columns representing the SMILES strings and rows corresponding to each of the substrings from the external list. The values in the cells represent the count of occurrences for each substring within each SMILES string.

This functionality enables users to gain insights into the prevalence and distribution of specific chemical motifs within their dataset. By quantifying the occurrences of desired substructures, researchers can identify compounds with specific features of interest, analyze trends across the dataset, and gain a deeper understanding of the chemical characteristics present in the compounds.

Overall, the counting of substrings from an external list in the loaded SMILES strings provides a powerful tool for exploring and analyzing chemical patterns. The resulting CSV file presents a structured and easily interpretable summary of the occurrence counts, facilitating further investigation and enabling researchers to make informed decisions based on the distribution of specific substructures within their chemical dataset.

## **Molecular formula extraction**

The process of extracting the molecular formula from a molecule involves iterating over all the atom objects in the molecule and counting the occurrences of each atom along with the number of attached hydrogen atoms. This iterative procedure enables the formation of the molecular formula.

Within a for loop, the algorithm sequentially evaluates each atom object in the molecule. For each atom, the algorithm keeps track of the number of occurrences by incrementing a counter associated with that particular atom. Additionally, the algorithm considers the number of hydrogen atoms attached to the atom.

By accounting for the hydrogen atoms attached to each atom, the algorithm accurately reflects the elemental composition of the molecule. Hydrogen atoms are an integral part of many organic molecules, and including them in the molecular formula ensures a comprehensive representation of the compound.

During each iteration, the algorithm increments the occurrence count for the corresponding atom and adds the count of attached hydrogen atoms to the total. This process is repeated for every atom in the molecule.

After completing the loop and evaluating all the atoms, the algorithm constructs the molecular formula based on the accumulated occurrence counts. The molecular formula provides a concise representation of the elemental composition of the molecule, indicating the types and quantities of atoms present.

By following this iterative procedure, our algorithm successfully extracts the molecular formula from the given molecule. The resulting formula serves as a valuable descriptor that summarizes the elemental composition of the compound, providing crucial information for further analysis and interpretation.

In conclusion, the process of extracting the molecular formula involves iterating over the atom objects in the molecule, counting the occurrences of each atom, and considering the number of attached hydrogen atoms. This approach allows for the accurate formation of the molecular formula, providing an essential representation of the elemental composition of the molecule.

# Limitations and Challenges

Handling complex and ambiguous SMILES strings

Dealing with large-scale datasets

Performance limitations and optimization opportunities

# **Results and Discussion**

## **SMILES validation quality evaluation**

In our evaluation of SMILES validation quality, we compared the validation results obtained using our program with those of RDKit, a widely-used chemoinformatics library. The dataset consisted of 80,836 SMILES strings obtained from ChEMBL [] and NPAtlas [] databases of natural compounds (**Appendix I**). This dataset contains great variety of both simple and complex organic substances with branched carbon skeleton and multiple cyclic systems.

**Picture X**. Results of SMILES strings validation by RDKit and our algorithm.

We observed that RDKit failed to validate 7 structures, while our algorithm identified 124 structures as not valid (including all not valid structures from RDKit). This difference is caused by additional atom valencies check in our algorithm, which cause fail of validation of such structures as carbocations, carbanions, radicals, elements with not-standard valencies and heavy elements not allowed in our algorithm (**Appendix II**).

Upon further analysis of the structures not validated by our algorithm, we observed several recurring patterns. These included carbocations and anions, which are charged species, radicals, which contain unpaired electrons, structures with heavy atoms that may have unusual valencies, and elements with non-standard valencies. These findings highlight the effectiveness of our algorithm in detecting and flagging chemically unstable or non-standard organic molecules.

The ability of our algorithm to identify and exclude such structures is particularly valuable in drug design, where the focus is on developing stable and bioactive compounds. Removing non-standard or unstable molecules from datasets is essential for maintaining data integrity, ensuring reliable analyses, and reducing the risk of erroneous results or misleading conclusions.

By using our algorithm to clean datasets from not standard and unstable organic molecules, researchers and drug designers can improve the quality and reliability of their compound libraries. This, in turn, enhances the efficiency and accuracy of subsequent drug discovery processes, such as virtual screening, lead optimization, and molecular modeling.

Furthermore, the identification of chemically unstable or non-standard molecules allows researchers to gain insights into the limitations and challenges associated with specific chemical features. By analyzing the types of structures that are frequently flagged as not valid, researchers can identify recurring issues and develop strategies to overcome or address them in future compound design and synthesis efforts.

In conclusion, our algorithm demonstrated strong performance in SMILES validation when compared to RDKit. The identification of non-standard and chemically unstable structures provides valuable insights and enables the cleaning of datasets, making it particularly useful in drug design. By implementing our algorithm, researchers can enhance the quality of their compound libraries, improve the reliability of analyses, and accelerate the discovery of novel, stable, and effective drug candidates.

Speed

Rd kit

Execution Time: 17.462661743164062

Average Time: 0.00021602580215701004

Average Time per 1000 0.21602580215701003

Our algorithm

Execution Time: 15.19392991065979

Average Time: 0.00018795994248428657

Average Time per 1000 0.18795994248428657

## Molecular formula extraction and validation results

In the validation process of our algorithm for molecular formula extraction, we performed a comparison between our program and RDKit, a widely-used chemoinformatics library. This comparison was conducted on a dataset consisting of SMILES strings that had successfully passed the validation process in our algorithm.

During the comparison, we extracted the molecular formulas using both our program and RDKit for each SMILES string in the dataset. Remarkably, we found that the extracted formulas from both methods were consistently identical for all SMILES strings. This outcome provides strong evidence that our algorithm for molecular formula extraction is functioning correctly and producing reliable results.

The validation of molecular formula extraction is an essential step in ensuring the accuracy and integrity of the chemical information obtained from SMILES strings. By demonstrating that our algorithm consistently yields the same results as RDKit, a well-established and trusted chemoinformatics library, we establish the robustness and effectiveness of our approach.

Moreover, the successful validation of molecular formula extraction indirectly indicates the accuracy and correctness of our algorithm for constructing the molecular graph object. The molecular formula is a fundamental property that encapsulates the elemental composition of a compound, and its extraction relies on correctly interpreting and organizing the atom information within the molecular graph.

The consistent agreement between the extracted formulas from our program and RDKit demonstrates that our algorithm for constructing the molecular graph object is capable of accurately representing the structural information encoded in the SMILES strings. This finding further strengthens the confidence in our algorithm's ability to capture the connectivity, bonding, and atom information, which are crucial aspects of the molecular graph construction process.

Overall, the validation of our algorithm for molecular formula extraction through a comparison with RDKit confirms its reliability and accuracy. This validation also indirectly verifies the correctness of our algorithm for constructing the molecular graph object. The consistent agreement between the results obtained from our program and RDKit provides a strong basis for confidence in the performance and functionality of our algorithm in processing and analyzing chemical information encoded in SMILES strings.

# Implementation details

Programming languages and libraries used

System architecture and design

Data preprocessing and storage

# Program Description

Conclusion

Summary of the project

Accomplishments and contributions

Future directions and enhancements

# Appendix

# References

[1] Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Modeling, 28(1), 31-36.

<https://www.ebi.ac.uk/chembl/>

https://www.npatlas.org/