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.

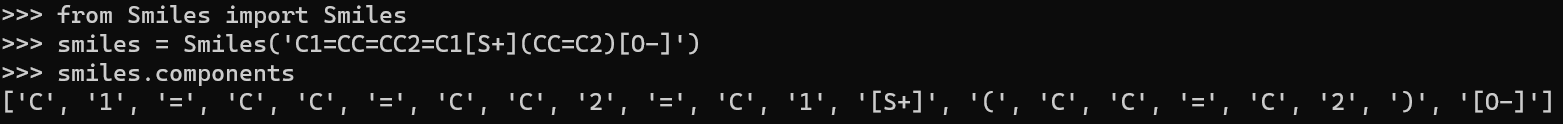
# **Methodology**

## **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).



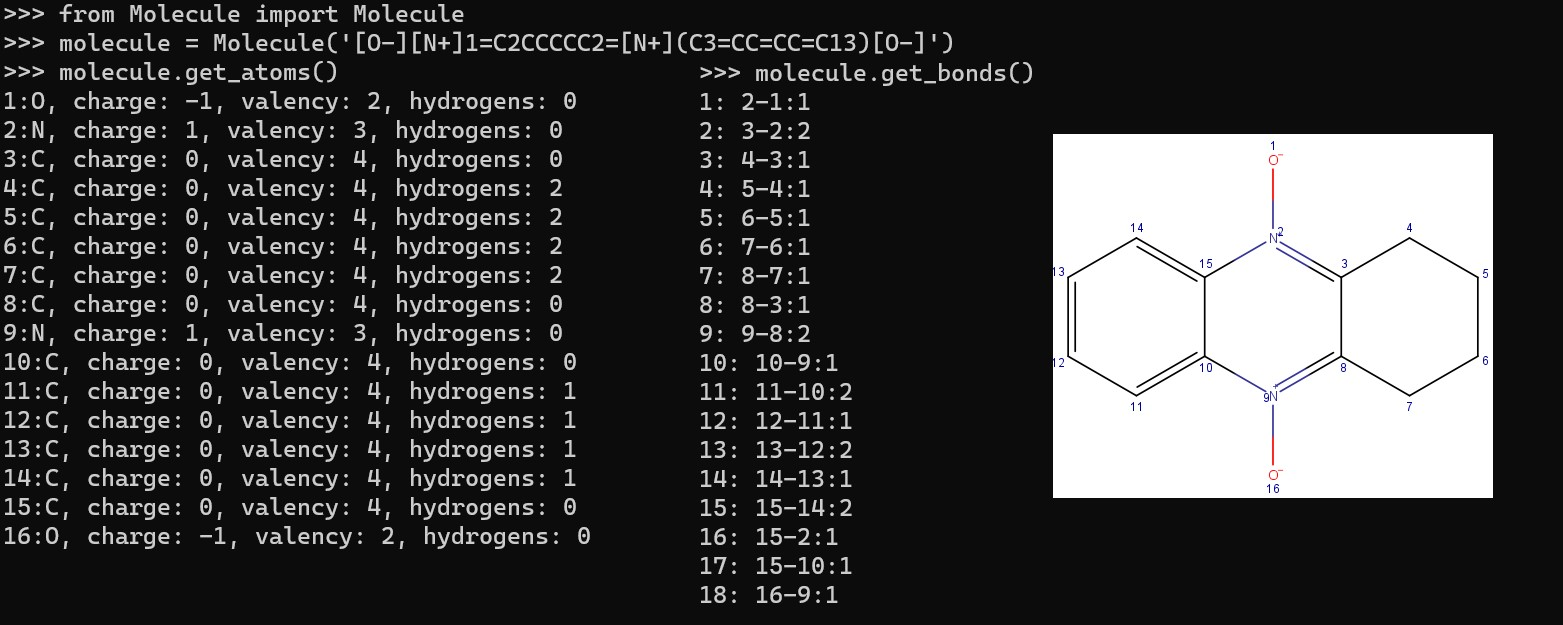
**Picture 1.** Example of breaking a SMILES string into components.

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**

# Limitations and Challenges

Handling complex and ambiguous SMILES strings

Dealing with large-scale datasets

Performance limitations and optimization opportunities

# Results and Discussion

SMILES validation performance evaluation

Molecular graph construction accuracy assessment

Molecular formula extraction and validation results

Substring occurrence counting analysis

# 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.