Fragment Explorer: A Comprehensive Tool for Fragment-Based Drug Design. PyQT Application

Thesis submitted to the SASTRA Deemed to be University
in partial fulfillment of the requirement
for the award of the degree of

B.Tech Bioinformatics

Submitted by

NAME: ANAGHAA K
(Reg. No.: 124013063)

May 2024

SCHOOL OF CHEMICAL AND BIOTECHNOLOGY

THANJAVUR, TAMIL NADU, INDIA – 613 401

SCHOOL OF CHEMICAL AND BIOTECHNOLOGY

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the thesis titled "**Fragment Explorer: A Comprehensive Tool for Fragment-Based Drug Design. PyQT Application**" submitted as a requirement for the award of the degree of B.Tech Bioinformatics to the SASTRA Deemed to be University, is a bonafide record of the work done by **Ms. Anaghaa. K (Reg. No.124013063**) during the final semester of the academic year 2023-24, in the School of Chemical and biotechnology, under my supervision. This thesis has not formed the basis for the award of any degree, diploma, associate ship, fellowship or other similar title to any candidate of any university.

Signature of Project Supervisor :

Name with Affiliation **:**M.Udayakumar, Assistant Professor – III

Date **: 27-05-2024**

Project *Viva voc*e held on _____

Examiner 1 **Examiner 2** **Examiner 3** **Examiner 4**

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF CHEMICAL AND BIOTECHNOLOGY

THANJAVUR – 613 401

Declaration

I declare that the thesis titled "**Fragment Explorer: A Comprehensive Tool for Fragment-Based Drug Design. PyQT Application**" submitted by me is an original work done by me under the guidance of M.Udayakumar, Assistant Professor – III, School of Chemical and Biotechnology, SASTRA Deemed to be University, Thanjavur during the final semester of the academic year 2023-24, in the School of Chemical and Biotechnology. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the thesis. This thesis has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidate(s) **:**

Name of the candidate(s) **: Anaghaa. K**

Date **: 27-05-2024**

Acknowledgments

I first thank my guide M.Udayakumar, Assistant Professor – III, SCBT-SASTRA Deemed to be University whose expertise was valuable in formulating my research work. He consistently steered me in the right direction whenever I needed it. I thank the review panel for their validation of my project and their valuable suggestions.

I would like to thank SASTRA DEEMED UNIVERSITY for its infrastructure and facilities.

Table of Contents

List of Figures

List of Tables

Abstract

Finding viable therapeutic candidate fragments in the large chemical space is a major difficulty for fragment-based drug design (FBDD). This problem is addressed by the computational tool Fragment Explorer, which provides an extensive feature set to optimise FBDD procedures. The potential of Fragment Explorer is investigated in this work with respect to fragment generation using BRICS and RECAP algorithms, fragment filtering according to physicochemical and pharmacological criteria, maximum common substructure identification, scaffold hopping for core structure analysis, and informative 2D fragment visualization. Through efficient utilisation of these functionalities, Fragment Explorer facilitates informed decision-making for researchers and expedites the identification of plausible therapeutic options. The present study underscores the potential of Fragment Explorer to transform FBDD and underscores its significance in pinpointing efficacious fragments for prospective treatment advancement.

keywords: Fragment-based drug design, RDKit, Scaffoldgraph, BRICS, RECAP, Maximum common substructure.

CHAPTER 1
INTRODUCTION

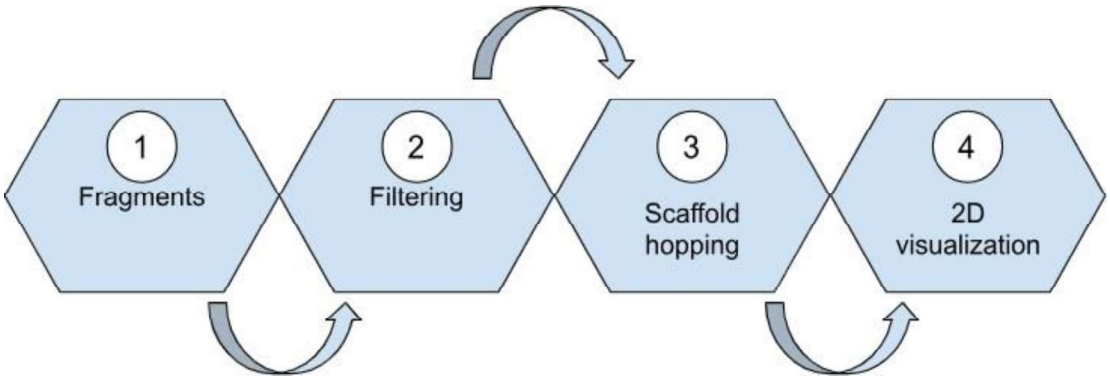
Traditional drug discovery is a systematic process that involves a large lot of in vitro and in vivo testing in an effort to find potential drugs that target certain biological systems. This method is thorough, but it takes a lot of effort, money, and time. Despite these challenges, it has proven successful in manufacturing a wide range of pharmaceuticals[1].

Fragment-based drug design (FBDD), utilizing natural materials, innovative scaffolds, and small molecular fragments from popular medications, has revolutionized this field. These fragments, which are typically fewer than 250 daltons, bind poorly at first but can be developed into potent therapeutic opportunities. Through increasing the binding affinity and efficacy of these fragments, FBDD seeks to significantly expedite the drug discovery process[1].

To expedite the design of fragment-based medications, computational tools such as Fragment Explorer are important. Users may easily browse fragment libraries, apply rules of three and other filtering criteria, and use the resulting fragments for scaffold hopping with the aid of Fragment Explorer. The technique of scaffold hopping involves incorporating new structures into a molecule to enhance specific features. Additionally, by highlighting the shared properties of the fragments, the application may display the maximum common substructure (MCS) between fragments formed by scaffold hopping, aiding in compound creation.

Drug discovery cannot be successful unless possible treatment candidates are identified efficiently and promptly. Fragment Explorer was developed to address this need. It has several powerful capabilities designed to generate, filter, prioritize, and display molecular fragments in an efficient manner. The program uses the Maximum Common Substructure (MCS) approach for comprehensive similarity searches against known ligands.

Strict limitations based on industry standards are also in place, such as the fragment screening Rule of 3. Unique BRICS and RECAP algorithms ensure precise molecular fragmentation, while scaffold hopping techniques look at various structural changes to increase fragment diversity and efficiency. Useful 2D visualizations facilitate the evaluation and analysis of fragmented data by researchers. Fragment Explorer also makes use of QT design, leveraging the computational capacity of Python for algorithmic implementation to create a complex yet user-friendly graphical user interface (GUI). This ensures that researchers can navigate and use the interface with ease, freeing them up to focus on analytical work instead of being sidetracked by challenging navigation. By merging these state-of-the-art features and addressing some of the shortcomings of traditional drug discovery methodologies, Fragment Explorer enhances the efficacy and efficiency of fragment-based drug synthesis. These computational technologies accelerate and improve the drug design process while using fewer resources. Eventually, this leads to the development of cutting-edge and effective pharmaceuticals. All things considered, Fragment Explorer and associated computational tools bridge the gap between outdated methods and more modern fragment-based approaches, which is a significant advancement in the field of drug development. These resources help researchers reduce costs, expedite the development of novel medications, and streamline the drug design process .



Fig(1.1)This is the representation of the main objective of the Fragment Explorer.

CHAPTER 2
OBJECTIVE

A computational tool called Fragment Explorer was developed to improve and expedite fragment-based drug design procedures. Its primary objective is to identify efficient fragments with the potential to become drug candidates. To achieve this, Fragment Explorer offers a range of powerful features designed to generate, filter, scaffold hopping, maximum common substructure and visualize molecular fragments effectively.

Problem statement: The challenge of navigating a large chemical space to identify fragments with potential for further optimization as drug candidates.

Principal Objective: Aims to identify efficient fragments that have the potential to become drug candidates.

To revolutionize fragment-based drug design by offering a user-friendly interface, intelligent filtering options, scaffold hopping, and visualization tools, ultimately aiding in the search for effective therapeutic candidates.

Secondary Objectives:

1. Fragmentation using BRICS and RECAP Algorithms:
    a. Uses the BRICS and RECAP algorithms to efficiently fragment molecules. Facilitates the production of various sets of molecular fragments for investigation[2].
    b. Utilizing advanced algorithms to break down larger molecules into smaller, manageable fragments.

2. Fragment Filtering:
    a. Creates criteria-based filters, such as the rule of three. Sort fragments according to critical physicochemical and pharmacological characteristics[3].

3. Scaffold hopping:

    a. ScaffoldGraph is an open-source cheminformatics library, built using RDKit and NetworkX(study of the structure, dynamics, and functions of complex networks.), for the generation and analysis of scaffold networks and scaffold trees.

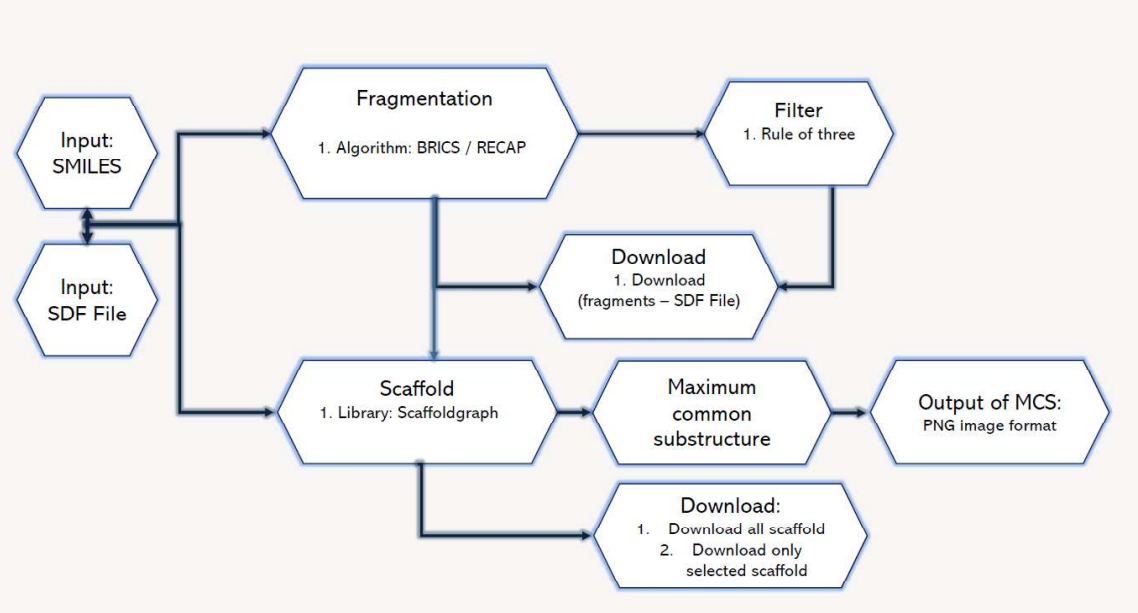
    b. Fragments are generated by simplifying a molecule to its core scaffold while removing the side chains and substituents. Its core scaffold while removing side chains and substituents[4].

4. Maximum common substructure:

    a. Using the maximum common substructure algorithm from RDKit user can identify common substructures from the scaffold[5].

5. Visualization Tools:

    a. Offers 2D representations of fragments to help comprehend their characteristics and relationships. Facilitates better decision-making and data interpretation for researchers.



Fig(2.1) Working of Fragment Explorer

With its extensive toolkit, Fragment Explorer aims to transform fragment-based drug creation. It provides informative visualization for in-depth research, scaffold hopping for exploring various chemical scaffolds, intelligent filtering utilizing rules like the rule of three, and efficient fragment creation. This technology reduces trial and error, offers practical insights, and encourages creativity in research, all of which help to expedite the drug discovery process. Because of its intuitive interface, which improves accessibility, it is a useful tool for researchers looking to find potential new treatments and develop medicine.

CHAPTER-3
METHODOLOGY

In order to efficiently fragment molecules, select fragments according to criteria, look for similarities, perform scaffold hopping, and offer clear 2D visualizations, Fragment Explorer was created. While similarity searches are made possible by the Maximum Common Substructure (MCS) technique, advanced algorithms are employed for fragmentation and filtering. By utilizing scaffold hopping techniques to investigate structural alterations and providing an intuitive interface for analysis and navigation, fragment-based drug design becomes more effective in identifying promising candidates.

3.1 Fragmentation:

RDKit, an open-source cheminformatics toolkit well-known for its adaptability in molecular modeling and analysis, is employed by the Fragmentation module. We use two different methods in this module: BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) and RECAP (Retrosynthetic Combinatorial Analysis Procedure) [2].

By using 11 more straightforward principles that are only dependent on bond types, RECAP produces a targeted collection of fragments that place less emphasis on synthetic relevance. However, BRICS uses sixteen cleavage rules that consider the surrounding chemical environment of bonds, resulting in fragments that are more likely to be synthesized. The efficacy of fragment-based drug design is increased by this deliberate use of algorithms, which guarantees a thorough approach to molecular fragmentation and satisfies the needs of both the synthetic feasibility emphasized by BRICS and the ease of use of RECAP[2].

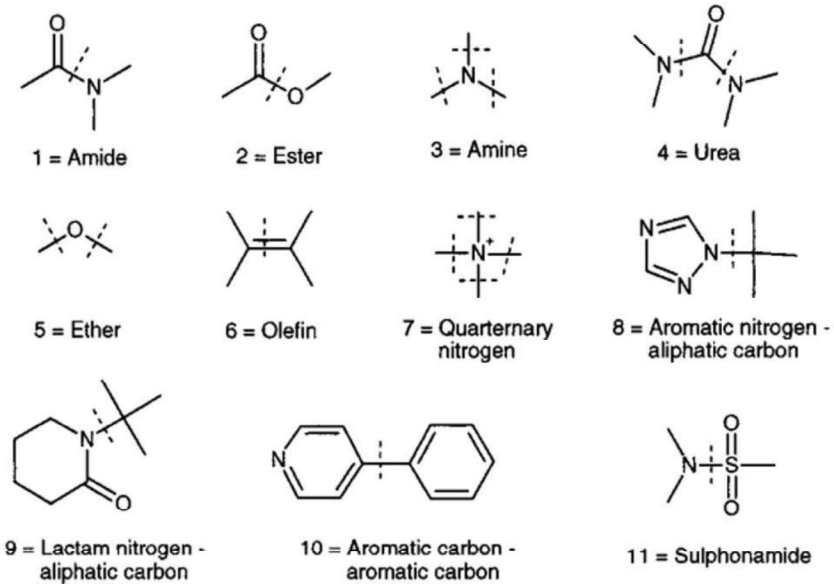
Feature displays are

1. Aromatic rings
2. Hydrophobic atoms
3. Hydrogen bonds Donors
4. Hydrogen bond Acceptors
5. LogP

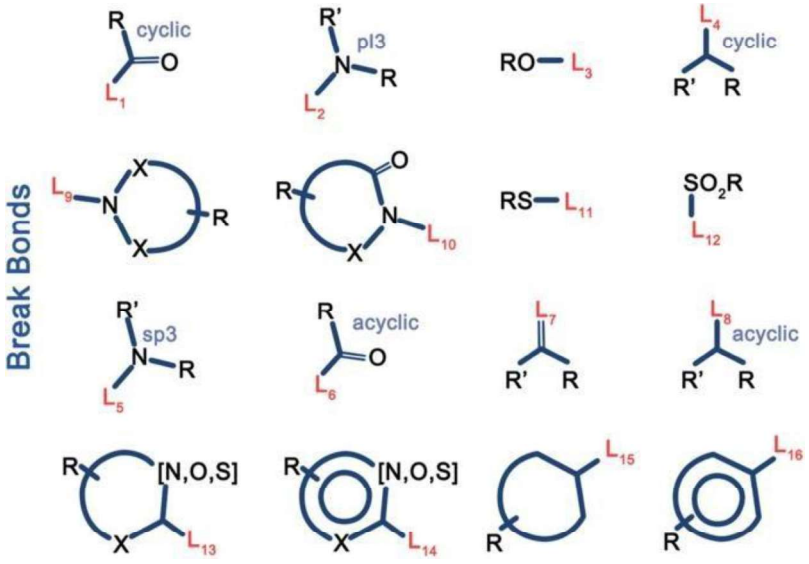
Input: Ligand SDF file or SMILES string

Output: Fragments generated from the input ligand and features are displayed

(a). RECAP elven default bond cleavage type.



(b) BRICS 16 rules for breaking bonds.



(c) Code displayed:

```
from rdkit import Chem
from rdkit.Chem import Descriptors
from rdkit.Chem import BRICS, RECAP
from rdkit.Chem import Draw
from rdkit.Chem import AllChem
import progressbar
def fragment_ligand(molecule):
    mol = Chem.MolFromSmiles(molecule)
    fragments=BRICS.BRICSDecompose(mol)
    return fragments
 def fragment_ligand_recap(molecule):
    mol = Chem.MolFromSmiles(molecule)
    fragment1 = Recap.RecapDecompose(mol)
    fragments = fragment1.GetAllChildren()
    return fragments
```

Fig(3.1) (a). RECAP elven default bond cleavage type, (b) BRICS 16 rules for breaking bonds, (c) Code used for fragmentation

3.2 Fragment Filtering:

We use a variety of criteria in fragment filtering to assess if the fragments that are produced are suitable for further optimization into possible treatment options. The Rule of Three, which establishes particular requirements that fragments should ideally fulfill in order to be deemed promising for drug creation, is one important guideline that we use [3].

The Fragment Likeness, or Rule of Three, is a crucial tool for determining fragment appropriateness and similarity for medication development. It includes multiple essential requirements, such as the molecule weight of the fragment is <300, the cLogP is ≤ 3, then the number of hydrogen bond donors is ≤ 3, and the number of hydrogen bond acceptors is ≤ 3.

Depending on their needs and tastes, users of our platform can choose from these pre-established guidelines. Enhancing the effectiveness and caliber of chemical library design in the drug discovery process requires an understanding of and adherence to certain guidelines and filters. The entire drug development process can be enhanced by researchers by carefully selecting and prioritising segments that have the potential to be effective treatment candidates by following the Rule of Three and related principles [3].

Inputs: Fragments from Module 1
Output: Filtered fragments that meet the specified criteria.

(a) Code used for check rule of three:

```
def check_ruleof3(self,mol):
        Molecular_weight=Descriptors.ExactMolWt(mol)
        logp = Descriptors.MolLogP(mol)
        h_bond_donor = Descriptors.NumHDonors(mol)
        H_bond_acceptors = Descriptors.NumHAcceptors(mol)
         rotatable_bonds = Descriptors.NumRotatableBonds(mol)

            return (molecular_weight <= 300
                and logp <= 3  and
                h_bond_donor <= 3 and
                h_bond_acceptors <= 3 and
                rotatable_bonds <= 3)
```

3.3 Scaffold hopping:

A molecule can be made into fragments by stripping it down to its essential framework and eliminating its side chains and substituents. Because it focuses on optimizing possible therapeutic candidates, scaffold hopping is an essential stage in fragment-based drug design (FBDD).

A key player in this field is the open-source cheminformatics library ScaffoldGraph, which was created with NetworkX and RDKit and focuses on creating and analyzing scaffold networks and scaffold trees.

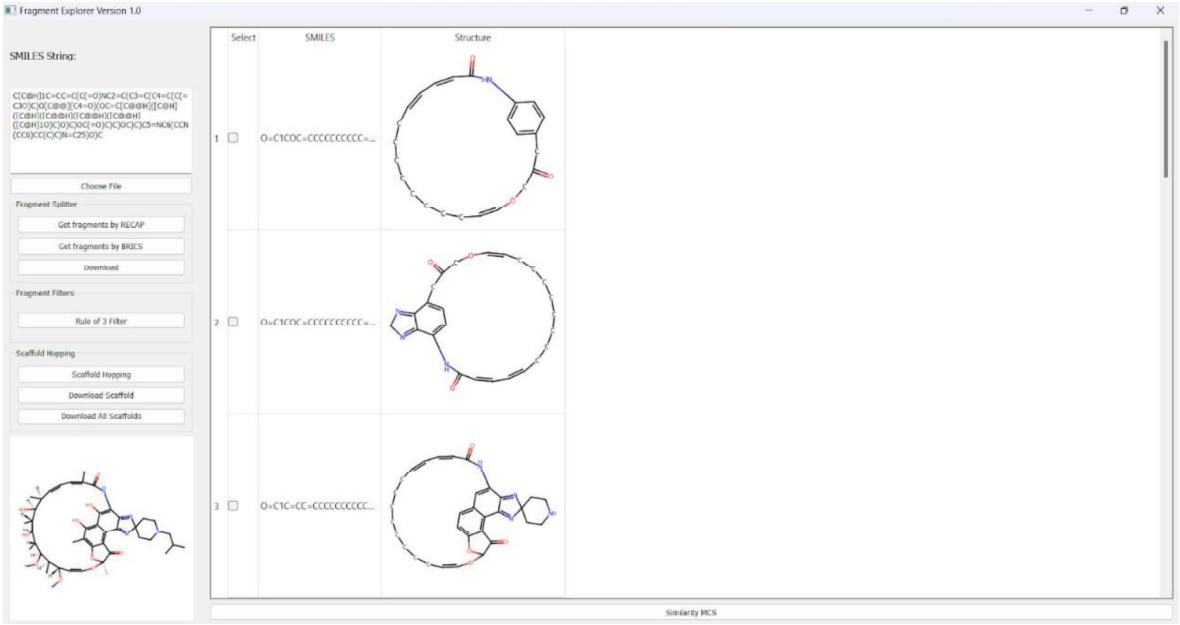
All things considered, ScaffoldGraph provides a thorough toolkit for practitioners and researchers working in the field of fragment-based drug discovery. It is an important tool for speeding up the search for new therapeutic agents because of its abilities in chemical space exploration, scaffold network development, scaffold tree analysis, virtual library construction, and integration with RDKit and NetworkX [4].

Inputs: Fragments generated after filtering, user can select any one.
Output: Scaffold are generated.

(a) Code:

```
# Import scaffoldgraph # Import rdkit
import scaffoldgraph as sg
from rdkit.Chem import Draw
from rdkit import Chem
# Create a molecule from a SMILES string
        mol=Chem.MolFromSmiles('O=C(O)c1ccc2c(c1)nc(-
c1ccccn1)n2C1CCCCCC1')
# ScaffoldGraph can generate all possible Murcko fragments, returning RDKit as
fragments
        frags = sg.get_all_murcko_fragments(mol)
        Draw.MolsToGridImage(frags)
```

(b) Scaffold hopping, the output of the scaffold is shown in the table



(c) Option is given for Scaffolding hopping

Fig(3.3) (a)Code, (b) Scaffold hopping, the output of the scaffold is shown in the table, (c)Option given for Scaffolding hopping.

In order to generate Murcko fragments, side chains and substituents must be eliminated from a molecule in order to reduce it to its basic scaffold or framework. The idea behind creating Murcko pieces is to concentrate on the primary structural motif, which is probably in charge of the biological activity of the molecule. The following are the steps involved in creating Murcko fragments and the guiding ideas behind it:

Process of Generation:

1. Identification of Core Scaffold: Begin by determining the molecule's core scaffold, or fundamental framework. The center rings and the linker atoms that join them are part of this structure. The molecule's action depends on the core scaffold, which also serves as the building block for the production of Murcko fragments [4].

2. Substituent and Side Chain Removal: After determining the core scaffold, eliminate any non-essential functional groups, side chains, and substituents from the molecule. In this stage, the molecule is reduced to its essential structural components while maintaining the core scaffold's connectivity [4].

3. Fragmentation into Murcko Fragments: By methodically eliminating non-scaffold atoms and links, fragment the simpler molecule into Murcko fragments.

Murcko fragments can be individual rings, linker atoms, or mixtures of these, and they represent various components of the core scaffold [4].

Fundamentals of Generating:

1. Focus on Essential Structure: The idea behind creating Murcko fragments is to concentrate on the key structural elements that make a substantial contribution to the biological activity of the molecule. Researchers can identify and prioritize the core scaffold for additional investigation by removing non-essential components [4].

2. Structural Similarity and variety: Murcko pieces are utilized to represent both structural variety and similarities among a group of molecules. Comparable compounds frequently have similar scaffolds, whereas different molecules have different core structures. Understanding structure-activity correlations and creating molecules with desired qualities are made easier by this idea [4].

3. Scaffold Hopping in Drug discovery: To investigate structural variety and find new lead compounds in fragment-based drug discovery, scaffold hopping relies heavily on murcko fragments. The idea of scaffold hopping is to move between distinct scaffolds while preserving crucial pharmacophoric components, which should result in the identification of new therapeutic options[4].

4. Chemical Space Exploration: By illustrating various structural motifs inside a compound library, Murcko fragments aid in the investigation of chemical space. Compound optimization, virtual screening, and the creation of targeted libraries for drug discovery are all aided by this exploratory paradigm[4].

To summarise, the process of creating Murcko fragments relies on the ideas of structural emphasis, variety discovery, scaffold hopping, and simplification. These guidelines help scientists find important structural motifs and create molecules with the best possible pharmacological characteristics.

3.4 Maximum common substructure:

In chemical informatics, the maximum common substructure (MCS) is a crucial term, especially in drug development and molecular similarity research. Finding an obvious shared structure among candidate compounds might give important information about possible patterns of activity when using MCS.

We use the RDKit library in Fragment Explorer to effectively use the maximum common substructure function. A popular open-source cheminformatics tool for molecular modeling, analysis, and drug development is called RDKit. The ability of the RDKit MCS algorithm to take atomic coordinates into account when creating the maximum common substructure is one of its less well-known but extremely helpful capabilities. This property is very useful when working with three-dimensional compounds and taking the atoms' spatial arrangement into account [5].

The following procedures describe how to use RDKit to find the largest common substructure while taking atomic coordinates into account:

1. Input Molecules: Begin by identifying the largest common substructure among a selection of potential molecules or compounds.

2. MCS Calculation: To determine the MCS between the input molecules, use the maximum common substructure function provided by the RDKit library.

3. Atomic Coordinates Consideration: Atomic coordinates can be taken into account while calculating MCS thanks to an underdocumented feature in RDKit. This characteristic is essential for precisely determining the shared substructure while taking into consideration the atoms' spatial arrangement in three dimensions.

4. Matching Atoms and Bonds: Considering the atomic coordinates and connectivity of each input molecule, the MCS algorithm matches atoms and bonds among them.

5. Result Interpretation: The shared substructure between the input molecules is revealed by the MCS calculation's output. The maximum common structure, a shared substructure among the candidate compounds, provides information on shared chemical characteristics and possible patterns of activity.

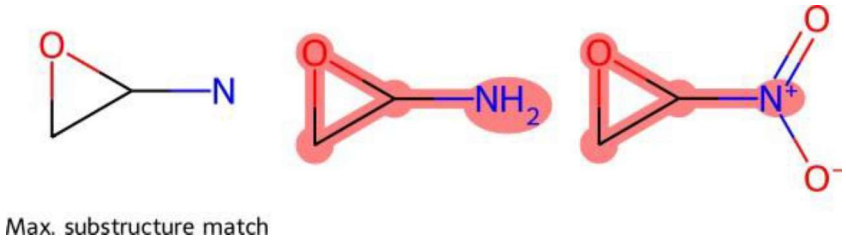
Fragment Explorer uses atomic coordinate consideration in MCS computation, one of RDKit's advanced capabilities, to improve the maximum common substructure analysis's relevance and accuracy. Drug development, molecular similarity analysis, and comprehending the structural underpinnings of chemical activity all benefit greatly from this capacity [5].

Inputs: Select any structures from scaffold.
Output: Common substructures will be viewed.

(a) Code:

```
def find_max_common_substructure(filtered_molecules):
    if len(filtered_molecules) < 2:
        print("Need at least two molecules to find common substructure.")
        return None
    # Convert filtered molecules from SMILES to RDKit Mol objects
    mol_list = [Chem.MolFromSmiles(smiles) for smiles in filtered_molecules
                    if Chem.MolFromSmiles(smiles) is not None]
    # Find MCS among the filtered molecules
    mcs = rdFMCS.FindMCS(mol_list)
    # Get the MCS as a SMARTS pattern
    mcs_smarts = mcs.smartsString
    # Convert the SMARTS pattern back to a Mol object for visualization
    mcs_mol = Chem.MolFromSmarts(mcs_smarts)
    return mcs_smarts, mcs_mol
```

(b) Output for MCS in PNG format



Fig(3.4)(a)Code, (b) Representation identifying the MCS

3.5 2D visualisations and User interface

2D visualizations:

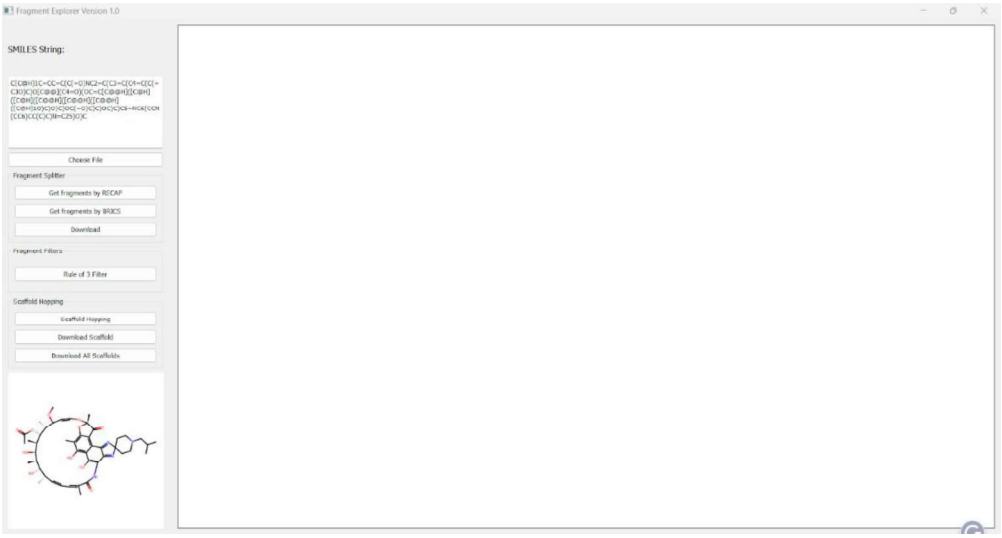
Fragment Explorer offers a range of functionalities crucial for fragment-based drug design. It allows users to generate various sets of molecular fragments based on specified criteria, apply filters to ensure drug-like qualities, prioritize fragments using a scoring mechanism, and visualize fragments in 2D representations for better comprehension.

User Interface:

The user interface of Fragment Explorer is designed with usability in mind. Its graphical interface is intuitive, making it easy for researchers to navigate through the tool's functionalities seamlessly. The GUI enhances user experience, enabling researchers to focus on analysis rather than struggling with complex interfaces.

Impact and Contributions: Fragment Explorer has the potential to significantly impact the field of drug discovery. By providing researchers with a comprehensive suite of tools, it accelerates the identification of potential lead compounds, thereby expediting the drug development process. The tool's intelligent filtering and prioritization capabilities contribute to more informed decision-making in selecting fragments for further investigation.

Inputs: Fragments generated from Modules 1 and 3.
Output: 2D images and 3D structures of the fragments



Fig(3.5) Visualization of the GUI when the user gives the input.

CHAPTER-4

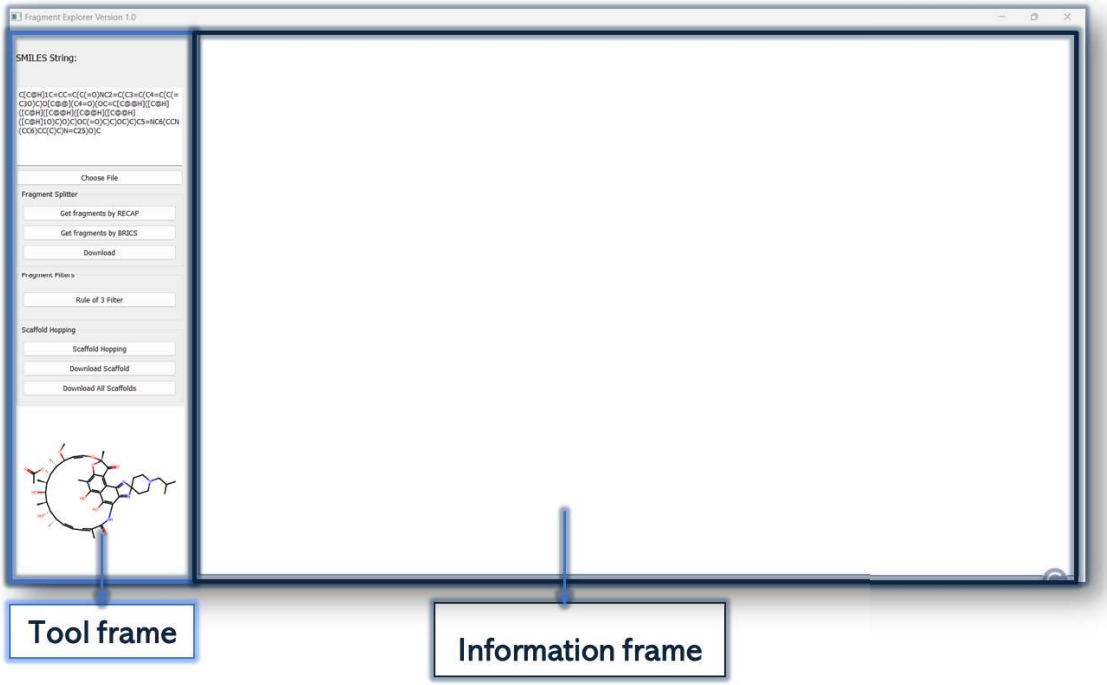
RESULTS AND DISCUSSION

4.1 GUI Frame usage:

Two separate frames comprise the interface:

1. Tool frame:
    a. A text field where SMILES strings can be entered.
    b. BRICS and RECAP are two examples of fragmentation algorithms from which to choose.
    c. Options for filtering, such as the Rule of Three standards.
    d. Extra features to enable scaffold hopping.

2. Information frame:
    a. Specific details on the pieces that were created, including their names, SMILES strings, and the fragmentation technique that was employed. a precise 300 by 300 pixel
    b. 2D representation of the fragments that offers an interactive and understandable depiction of the chemical structures.



Fig(4.1) This is the representation when the user gives SMILE input to this primary display of a 2D molecule in the image frame.

4.2 Fragmentation:

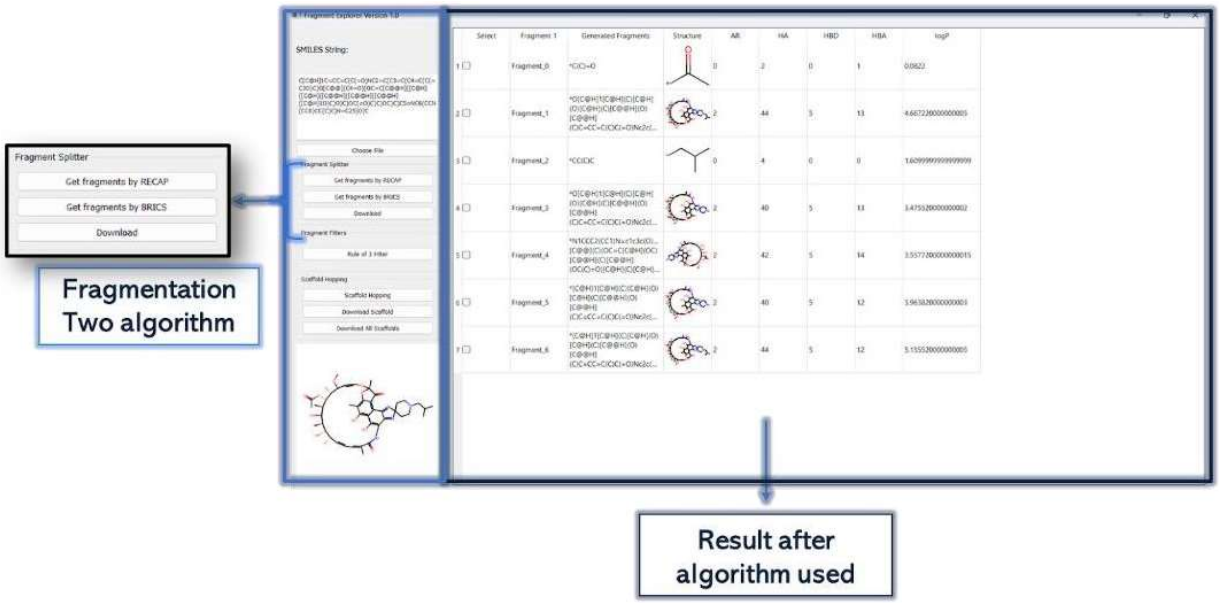
1. Workflow Description: User Interaction: To start, the user inputs a SMILES string into the Tool Frame's input box.

2. Algorithm Selection: From the available alternatives, the user chooses a fragmentation algorithm (e.g., BRICS or RECAP).

3. Filtering: To further hone the results, the user can apply filters like the Rule of Three.

4. Display of Results: The Information Frame presents the generated fragments once an algorithm has been chosen and filters have been applied.

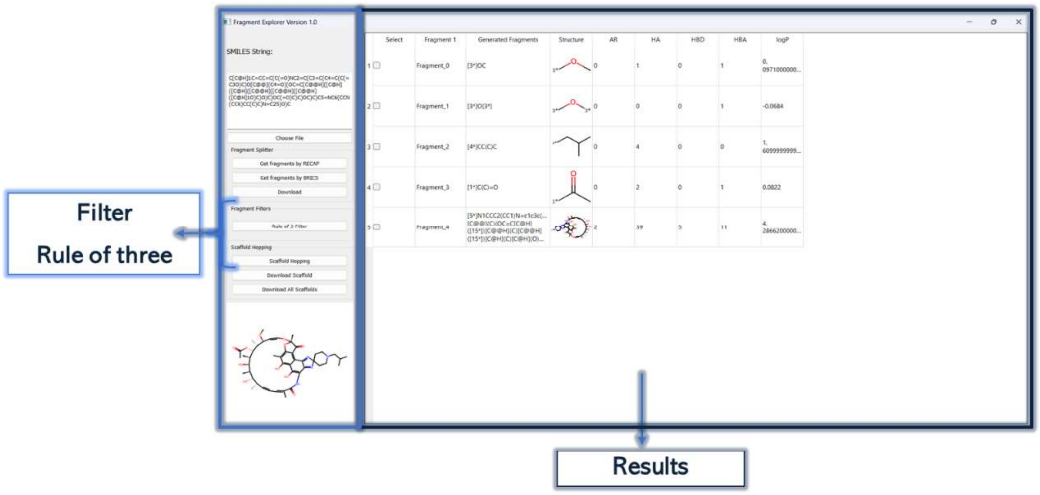
5. 2D Visualization: Each fragment's 2D visualization is included in the Information Frame in addition to the comprehensive fragment information. With its 300 × 300 pixel size, this representation makes it simple for the viewer to examine the chemical structures.



Fig(4.2)Representation after selecting any one of the algorithms either RECAP or BRICS.

4.3 Rule of three - Filter:

Following the use of the rule of three and the utilization of algorithms such as RECAP or BRICS in Fragment Explorer, the outcomes are tastefully shown in an extensive table style. Every row in the table represents a produced fragment and highlights important characteristics that are necessary to assess whether or not they are suitable candidates for prospective drugs. The table displays the result representation as follows:



Fig(4.3) Representation of the result after using the rule of three

The table displays the result representation as follows:

Aromatic Rings: Shows how many aromatic rings are found in a given chemical.

The number of hydrophobic atoms (such as carbon and fluorine) in the molecule is indicated.

Hydrogen Bond Donors: Displays the quantity of donors of hydrogen bonds.

Hydrogen Bond Acceptors: Indicates how many acceptors there are for hydrogen bonds.

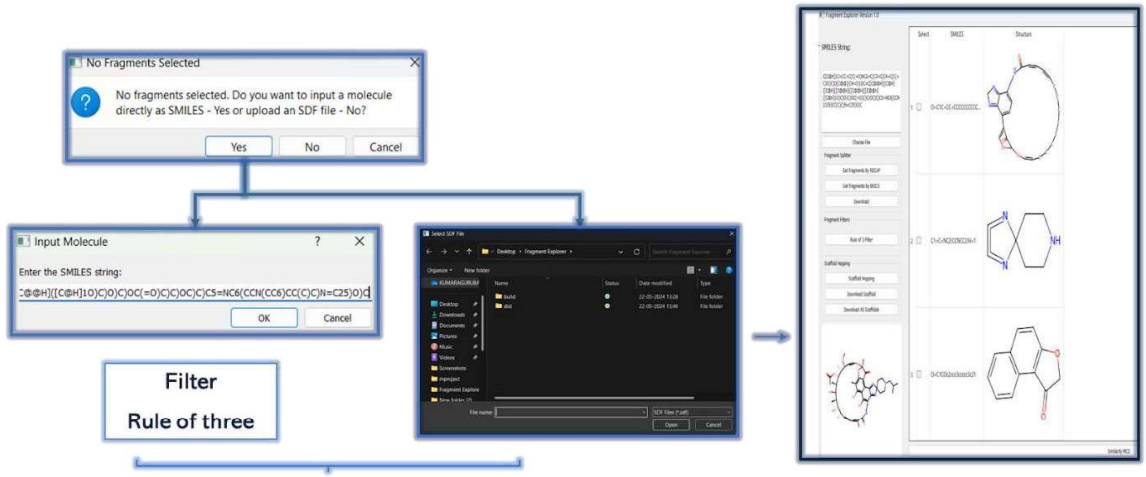
LogP: Gives the LogP value, which represents the lipophilicity and possible cell membrane penetration of the chemical.

Key molecular features are arranged in an orderly table format that facilitates compound comparisons and evaluations of their applicability for many purposes, including chemical production and medication creation.

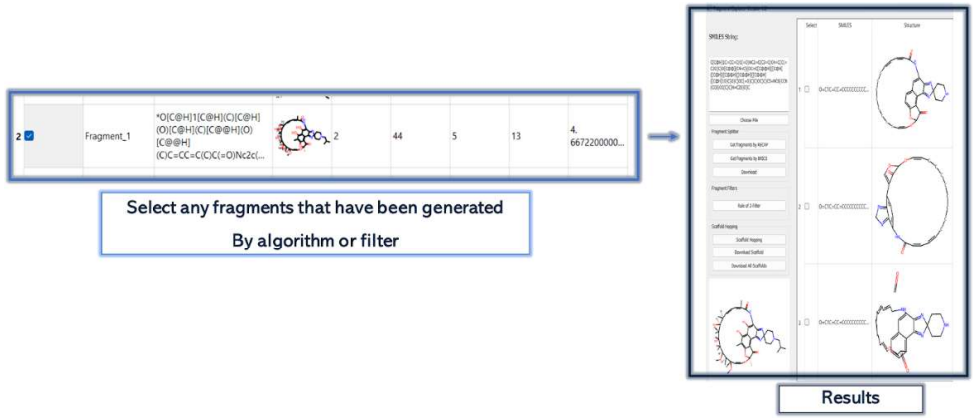
4.4 Scaffold hopping:

There are two major ways to use this option are as follows:

1. Direct import SDF or SMILES as input
2. Select any fragments generated by the algorithm or after using a filter rule of three.



a. Direct import SDF or SMILES as input



b. Select any fragments generated by the algorithm or after using a filter rule of three.

Fig(4.4) (a)Direct import SDF or SMILES as input,(b)Select any fragments generated by the algorithm or after using a filter rule of three.

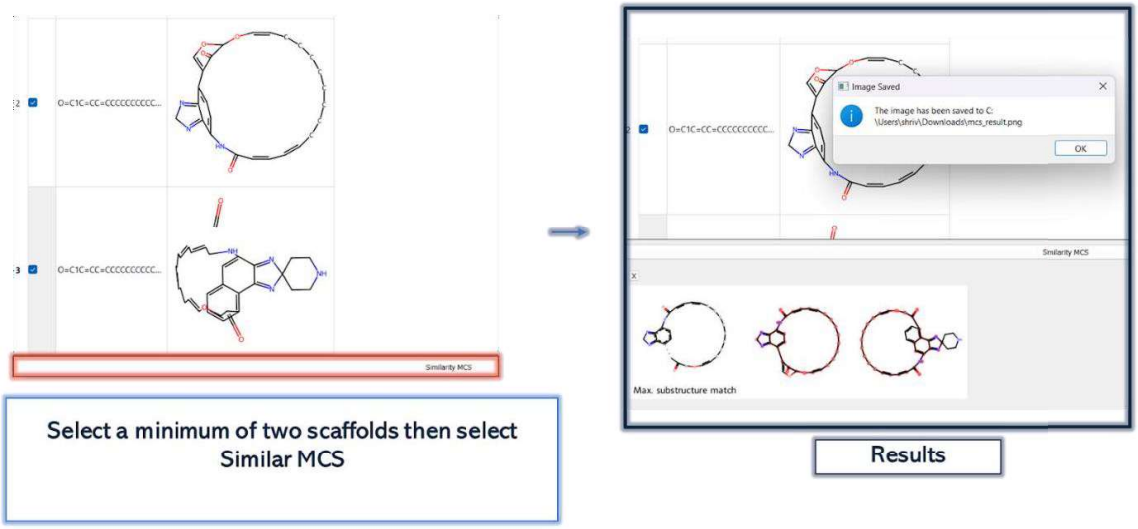
4.5 Maximum common substructure

Scaffold hopping scaffolds are the only scaffolds to which the maximum common substructure (MCS) feature in Fragment Explorer can be applied. It is mandatory for users to choose two scaffolds or more for the MCS analysis. The tool's UI presents the MCS analysis results in a split-screen layout. Furthermore, an automatic PNG image displaying the MCS result is downloaded.

Here's a more detailed breakdown of how the process goes:

1. Selection of Scaffolds: Users access Fragment Explorer's scaffold hopping section and choose two or more scaffolds for MCS analysis.

2. MCS Analysis: Using the scaffolds that have been chosen, Fragment Explorer does the MCS analysis to determine the maximum common substructure that they share.



Fig(4.5) Representation working of Maximum common substructure

3. Split-Screen Display: Within the Fragment Explorer interface, the MCS result is shown in a split-screen format. Users can compare and evaluate the MCS results with the chosen scaffolds using this format.

4. Automatic PNG Image Download: Fragment Explorer automatically creates a PNG image that shows the MCS result in addition to the split-screen display.
The PNG file is downloaded to the user's device, giving the chosen scaffolds' shared common substructure a visual representation.

5. Replacing previous Image: The new PNG image download will take the place of the previous one if users experiment with alternative scaffold combinations for MCS analysis.
Because the program does not store previous photos, users are urged to manually verify and save the PNG image if they are happy with the MCS results.

By streamlining the MCS analysis for many scaffolds, this procedure offers comparative and visual insights into the common substructure that various scaffold combinations share. It is simple for users to evaluate the outcomes and store the PNG image for later use or documentation.

4.6 Properties of Fragment Explorer:
Basic properties of standalone software:
File Description: Fragment Explorer
Type: Application(.exe)
Size: 143 MB

Description of the File Type: Application (.exe)
Benefit: An executable file's standalone format simplifies installation and distribution. Users' setup process is made simpler because they don't require any additional software or dependencies.
Dimensions: 143 MB
Benefit: A modest file size shows that resources and features that are required are included while yet being manageable. This harmony guarantees that the program is extensive without being too huge, facilitating quick downloads and installations.

Principal Benefits

1. Installation simplicity reduces compatibility problems and installation faults.
2. Portability: Easily reusable and transferable between Windows versions.
3. Full Functionality: Packed with functions without being unduly complex.
4. User-Friendly: Easy to use GUI accessible to all technical skill levels.
5. Effective Resource Management: Designed to run smoothly on a range of hardware setups.
6. Security and Stability: Lower chance of conflicts or interference with other software.
7. Performance: Tasks are processed quickly and effectively.
8. Minimal Dependency: Completely self-sufficient, complete with all required resources.

4.7 Comparisons for scaffold hopping and fragmentation between Schrödinger's tools and Fragment Explorer

Aspect	Fragment Explorer	Schrodinger's Tools
Goals and Concentration	Gratitude Concepts for Fragment-based Medication design novel	Advanced molecular modeling and drug discovery
User Base	Researchers, students	Industry workers, and computational chemists in the field of bioinformatics
Functionality	Basics of fragmentation, rule of three filters, basic scaffold hopping, maximum common substructure	Simulations, ligand docking, Molecule modelling
Features	Fragment libraries, scaffold hopping	Virtual screening, complex drug design workflows
Software Architecture	Open–source(RDKit, Scaffoldgraph)	Proprietary algorithms and databases
Cost	Open-source/free	Commercial software with licensing fees
Accessibility	Accessible to students and researchers	Tailored for industry professionals and teams
GUI interface	User-friendly	Advanced user-friendly interface for professional use

Table 1: Table of comparisons between Fragment explorer and Schrodinger's tool

CHAPTER- 5
CONCLUSION

The standalone tool for Fragment Explorer is intended to help researchers and students investigate Fragment-based drug design, with the goal of improving their comprehension of fragmentation, the advantages of filters such as the rule of three, scaffold hopping, and the notion of maximum common substructure. With its excellent performance, this program helps create scaffold libraries and fragment libraries that are saved in the SDF file format. These can be easily included into future studies on fragment-based drug design.

In the future, there will be chances to add more functionality to this tool. Our application provides a user-friendly interface to fill the gap in the lack of a comprehensive GUI interface provided by open-source libraries like RDKit and Scaffoldgraph. By making this interface accessible to people from diverse backgrounds, fragment-based drug design principles can be explored in a clearer and more natural manner.

CHAPTER-6

REFERENCES

[1] Degen J, Wegscheid-Gerlach C, Zaliani A, Rarey M. On the art of compiling and using 'drug-like' chemical fragment spaces. ChemMedChem. 2008 Oct;3(10):1503-7. doi: 10.1002/cmdc.200800178. PMID: 18792903.

[2] Lewell XQ, Judd DB, Watson SP, Hann MM. RECAP--retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. J Chem Inf Comput Sci. 1998 May-Jun;38(3):511-22. doi: 10.1021/ci970429i. PMID: 9611787.

[3] Kralj S, Jukič M, Bren U. Comparative Analyses of Medicinal Chemistry and Cheminformatics Filters with Accessible Implementation in Konstanz Information Miner (KNIME). Int J Mol Sci. 2022 May 20;23(10):5727. doi: 10.3390/ijms23105727. PMID: 35628532; PMCID: PMC9147459.

[4] Schuffenhauer A, Ertl P, Roggo S, Wetzel S, Koch MA, Waldmann H. The scaffold tree--visualization of the scaffold universe by hierarchical scaffold classification. J Chem Inf Model. 2007 Jan-Feb;47(1):47-58. doi: 10.1021/ci600338x. PMID: 17238248.

[5] Cao Y, Jiang T, Girke T. A maximum common substructure-based algorithm for searching and predicting drug-like compounds. Bioinformatics. 2008 Jul 1;24(13):i366-74. doi: 10.1093/bioinformatics/btn186. PMID: 18586736; PMCID: PMC2718661.

[6] Li Q. Application of Fragment-Based Drug Discovery to Versatile Targets. Front Mol Biosci. 2020 Aug 5;7:180. doi: 10.3389/fmolb.2020.00180. PMID: 32850968; PMCID: PMC7419598.

[7] Liu T, Naderi M, Alvin C, Mukhopadhyay S, Brylinski M. Break Down in Order To Build Up: Decomposing Small Molecules for Fragment-Based Drug Design with eMolFrag. J Chem Inf Model. 2017 Apr 24;57(4):627-631. doi: 10.1021/acs.jcim.6b00596. Epub 2017 Apr 4. PMID: 28346786; PMCID: PMC5433162.

[8] Konteatis Z. What makes a good fragment in fragment-based drug discovery? Expert Opin Drug Discov. 2021 Jul;16(7):723-726. doi: 10.1080/17460441.2021.1905629. Epub 2021 Mar 26. PMID: 33769176.

[9] Open source cheminformatic toolkit: https://www.rdkit.org/docs/Overview.html

CHAPTER 7

APPENDIX

7.1 Plagiarism report

REPORT OF THE PLAGIARISM CHECK

THIS REPORT CERTIFIES THAT THE ATTACHED WORK
demo2_report
WAS CHECKED WITH THE PLAGIARISM PREVENTION SERVICE MY.PLAGRAMME.COM
AND HAS:
SIMILARITY

1%

RISK OF THE PLAGIARISM

4%

PARAPHRASE
0%
IMPROPER CITATIONS
0%

File name: demo2.pdf
File checked: 2024-05-24
Report generated: 2024-05-24