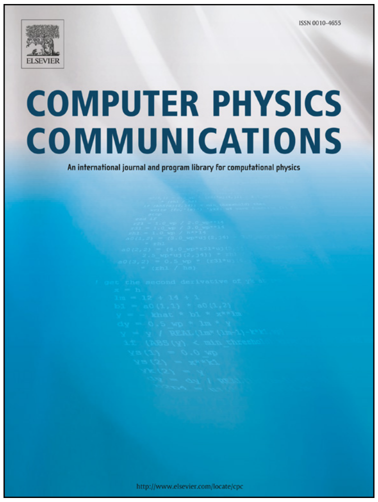

ISSN 0010-4655
ELSEVIER
COMPUTER PHYSICS COMMUNICATIONS
An international journal and program library for computational physics
http://www.elsevier.com/locate/cpc

Molecular Arrangement and Fringe Identification and Analysis from Molecular Dynamics (MAFIA-MD): A Tool for Analyzing the Molecular Structures Formed during Reactive Molecular Dynamics Simulation of Hydrocarbons

Khaled Mosharraf Mukut, Somesh Roy and Eirini Goudeli

PII: S0010-4655(22)00043-1
DOI: https://doi.org/10.1016/j.cpc.2022.108325
Reference: COMPHY 108325

To appear in: *Computer Physics Communications*

Received date: 25 October 2021
Revised date: 27 January 2022
Accepted date: 16 February 2022

Please cite this article as: K.M. Mukut, S. Roy and E. Goudeli, Molecular Arrangement and Fringe Identification and Analysis from Molecular Dynamics (MAFIA-MD): A Tool for Analyzing the Molecular Structures Formed during Reactive Molecular Dynamics Simulation of Hydrocarbons, *Computer Physics Communications*, 108325, doi: https://doi.org/10.1016/j.cpc.2022.108325.

Highlights

- Identifies aromatic and alicyclic ring structures in MD simulations of hydrocarbons.
- Provides chemical characterization from only trajectory files from MD.
- Calculates statistics of fringe spacing in molecular clusters for validation with HRTEM.
- Batch processes a series of trajectory files from reactive molecular dynamics.

Molecular Arrangement and Fringe Identification and Analysis from Molecular Dynamics (MAFIA-MD): A Tool for Analyzing the Molecular Structures Formed during Reactive Molecular Dynamics Simulation of Hydrocarbons

Khaled Mosharraf Mukut[a], Somesh Roy[a,*], Eirini Goudeli[b]

[a]*Marquette University, 1250 W Wisconsin Ave, Milwaukee, 53233, Wisconsin, United States*
[b]*University of Melbourne, Grattan Street, Parkville, 3010, Victoria, Australia*

Abstract

Reactive molecular dynamics (RMD) simulations are becoming popular, with the recent developments in high-performance and efficient computing architecture, for investigating fundamental physicochemical behaviors of reacting species. Due to the complexity of hydrocarbon system, characterization of molecules formed during RMD simulations of hydrocarbons can be very challenging for simulations involving a large number of molecules. The novel post-processing utility "MAFIA-MD" – an acronym for "Molecular Arrangement and Fringe Identification and Analysis from Molecular Dynamics" – presented in this manuscript can analyze and perform molecular characterization of a large number of RMD trajectory files (.XYZ files) simultaneously. The utility will be useful for analysis and chemical characterization of trajectories without the large bond information files. A graphical user interface (GUI) is developed for easy operation. The utility analyzes the existing cyclic structures in the domain and generates statistics of alicyclic and aromatic hydrocarbons based on the number of carbon atoms. Alongside the statistical analysis, the program also separates the constituent molecules and extract their chemical information in terms of Simplified Molecular-Input Line-Entry System (SMILES) and Spatial Data File (.SDF)". A methodology

[*] Corresponding author

for calculating fringe spacing is also implemented in the code for validation of RMD simulation with High-Resolution Transmission Electron Microscopy (HRTEM) images.

Keywords: Molecular Dynamics, Ring Structures, Fringe, Soot Chemistry, Hydrocarbon Chemistry, SMILE

PROGRAM SUMMARY

Program Title: `MAFIA-MD`
CPC Library link to program files: `https://doi.org/10.17632/s7dsk553fh.1`
Developer's repository link: `https://github.com/kmmukut/MAFIA-MD.git`
Licensing provisions: MIT
Programming language: `Python` 3.7
Supplementary material: Bash script for splitting long continous trajectory files with multiple timestep into trajectory files for individual timestep and some example input trajectory files for testing.
Nature of problem: Reactive molecular dynamics (RMD) simulations of hydrocarbons allow chemical reactions between different hydrocarbon molecules through bond breaking and new bond formation. The information about the chemical changes is stored in the bond information files and trajectory of individual atoms are stored in trajectory files. While the bond files can become very large for large simulations, trajectory files remain quite small. `MAFIA-MD` will be useful for practitioners interested in analyzing the chemical structures of emerging molecular clusters from small individual trajectory files without requiring the large, memory-intensive bond information files. Identification of chemical characteristics, particularly aromatic and alicyclic ring structures and molecular fringes, are important for gaining insights from RMD simulations of hydrocarbons, especially in exploration of soot formation during combustion. The capability of isolating the cyclic and non-cyclic molecules from trajectory files is the novel part of `MAFIA-MD`.
Solution method: The coordinates of individual atoms are taken as the input for the program. The coordinates are then analyzed to create an adjacency matrix [1] based on the bond distance between carbon atoms. This adjacency matrix is then converted into a directed graph and the cyclic structures are extracted from the graph by implementing an efficient depth-first-search algorithm [2] developed by Johnson [3]. Apart from the ability to isolate cyclic and non-cyclic components by chemical structural analysis, a methodology to calculate fringe spacing is also implemented by calculating the distances between cyclic structures. The utility uses a `Python` interface for easy management of the input and output parameters. The detailed solution methodology has been discussed in the manuscript.
Additional comments including restrictions and unusual features: In the current

version, the utility can analyze only carbon and hydrogen atoms. The future versions will implement the addition of other common atoms found in aromatic molecules, e.g. nitrogen, oxygen, sulfer, etc. `MAFIA-MD` can be run on Linux, macOS, and Windows operating systems.

1. Introduction

Reactive molecular dynamics (RMD) simulation is a method of analyzing both the physical and chemical changes of atoms and molecules. During the simulation, individual atoms and molecules are allowed to interact with each other based on the chemical and inter-atomic potentials along with the Newton's equations of classical dynamics or in some cases, Schrödinger's equations. With the recent advancement of computational resources, reactive molecular dynamics (RMD) has become more and more practical for performing the first principal analysis of unknown physics, especially in the field of biomolecular chemistry [4].

Several molecular dynamics potentials have been developed in recent years to capture the chemical and structural changes in reactive atomic clusters containing carbon [5]. Among them, empirical models such as Tersoff [6, 7], REBO [8, 9], AIREBO [10], Reactive force-field potential (ReaxFF) [11], Environment Dependent Interaction [12], and Charge Optimized Many-Body [13] potentials have been widely used in recent studies [e.g., 14, 15, 16, 17]. Alongside the empirical potentials, machine learning based neural network potentials [18] are also gaining popularity for being computationally efficient. Quantum molecular dynamics (QMD)[19] developed by Car and Parrinello [20] is another unified scheme for molecular dynamics simulation which utilizes the density functional theory for accurate prediction of electronic structures. Among these, for comprehensive studies focusing on the chemical changes of complex hydrocarbon species (e.g. soot formation), ReaxFF potential [11] has gained popularity among the contemporary researchers [21, 22, 23, 24, 25, 26, 27]

Soot is primarily carbon nano-particles formed during incomplete combustion of hydrocarbon fuels [28]. Under certain conditions (usually in absence of enough oxidizing species) during combustion, mostly gaseous hydrocarbon molecules react with each other to create large molecular clusters which can no longer be treated as gaseous species. These solid or liquid-like clusters are incipient soot and the process of this gas-to-particle transition is called soot inception or nucleation. The exact mechanism of this transformation from gas phase species to solid particles is arguably the least understood phenomenon during soot formation [29]. The widely accepted theories of soot inception indicates that soot inception starts with the formation of polycyclic aromatic hydrocarbons(PAHs) [30, 31, 29], particularly with 5- and 6-membered rings. The initial gas-phase PAH molecules grow and combine together via physical and chemical interactions to form the first soot particles. But the exact physico-chemical processes are yet to be confirmed. The complexity of hydrocarbon systems and the time and length scales of the process makes it very difficult to study the inception process experimentally. This has led to multiple competing and complimentary hypotheses of soot inception [e.g., 30, 31, 29, 25]. The reactive molecular dynamics (RMD) simulation can be very helpful in unraveling the physico-chemical processes of soot inception and can help bridge the gap between theoretical hypotheses and experimental observation at different scales. The use of RMD can also extend to the later stages of soot evolution, where the incipient soot particle grows via physical and chemical interaction with other soot and gas phase species. In order to properly interpret and validate RMD results, it is important to identify, differentiate, and analyze various physical and chemical structures such as rings and fringes that are formed in soot particles. These characteristics often are related to soot reactivity and maturity [32, 33]. The study of temporal evolution of these features will help resolve the mystery of soot inception and evolution. The post-processing utility "Molecular Arrangement and Fringe Identification and Analysis from Molecular Dynamics" or (`MAFIA-MD`) presented in this manuscript provides practitioners an ability to study these features with ease.

In usual practice, RMD simulations of soot related studies start with a set of hydrocarbon molecules in a confined domain [21]. The chemical interactions are then captured using an appropriate molecular dynamics potential as time progresses. The bond-related information (bond orders) is stored in a bond information file (also known as bonds file). The time-resolved snapshot and atomic trajectories in the simulation domain are also available in "`.XYZ`" [34] format. These "`.XYZ`" files (also referred as trajectory files) contain the coordinates of individual atoms at different times. The size of the bonds and trajectory files depends on how frequently the data is saved and the interconnectivity of the atoms. In high-temperature application such as combustion, where the chemical reactions are fast, it is important to save these files frequently (e.g. every few picosecond) in order to capture the fast chemical evolution. Furthermore, the atomic interconnectivity of hydrocarbon systems can grow large quite fast. For example, a 0.1 ns of acetylene combustion simulation consisting of 6000 atoms at 1500 K can lead to a bond information file of approximately 5 GB and a corresponding trajectory file of 300 MB. Because of its smaller size, the capability of extracting key features of atomic redistribution and molecular restructuring only from the trajectory file lead to easier visualization and faster analysis.

The goal of `MAFIA-MD` is to use the trajectory files individually and extract the chemical and structural information, primarily by detecting the presence of 5-, 6-, and 7-membered alicyclic and aromatic ring structures in an atom cluster. The identification of ring structure is important because the stability of such structures, particularly of aromatic rings, is thought to be a key for inception and growth of soot. Therefore, quantitative information on cyclic structures in soot particles can be helpful for identifying the important chemical pathways in soot formation and growth. Additionally, the planarity and curvature of the cyclic structures play important roles in dictating the stability of individual molecules and morphology of an incipient soot nucleus. The identification of these features can thus be utilized to analyze the stability of molecular clusters. The presence of planar and curved surfaces inside molecular clusters creates the optical fringes observed in high-resolution transmission electron microscope (HRTEM) images of soot particles. The quantification of these surfaces provides an accurate and meaningful pathway for direct experimental validation of RMD simulations with HRTEM images.

`MAFIA-MD` extracts this information from RMD simulations using an efficient algorithm implemented using `Python`. For the sake of completeness, the chemical bond information is calculated using an algorithm proposed by Kim and Kim [35]. This enables `MAFIA-MD` to capture the chemical information of the relevant soot forming molecules and export them in chemically understandable and usable formats such as simplified molecular-input line-entry system (SMILES) [36] and spatial data file (SDF) [37]. In short, `MAFIA-MD` captures the number of different alicyclic and aromatic structures present in the simulation domain, the percentage of alicyclic/aromatic and aliphatic carbons, the chemical representations of the constituent molecules existing in the domain, and provide a way to calculate the fringe spacing statistics of soot cluster obtained from RMD simulations. A graphical user interface (GUI) is developed for the easy management of input parameters. The main functions of the utility can be divided into following three segments:

1. **Identification of cyclic structures**
   - C/H ratio
   - Number/percentage of alicyclic and aromatic carbons
   - Number/percentage of aliphatic carbons
   - Statistics of 5-, 6-, and 7-membered ring structures
2. **Chemical characterization of atom clusters**
   - SMILE string for easier vizualization of molecules
   - Export into a molar file format (SDF)
3. **Identification and analysis of molecular fringes in soot for validation with HRTEM images**

It should be noted here that the MAFIA-MD does not strictly check for all four classical aromaticity conditions as per Huckel's rule. It is impossible to extract the exact electronic structure from the trajectory files, therefore conditions such as the presence of $(4n+2)$ π electrons or perpendicularity of p-orbitals are not possible to check. Furthermore, due to the nature of the MD simulations the atoms of an aromatic ring may not always lie in a single plane at an instanteneous timestamp. A strict planarity check will rule out some of these aromatics which may have a single out-of-plane carbon in an instanteneous time. A planarity check criterion is included and implemented in MAFIA-MD for future development in checking the planarity condition for aromaticity. However, in the current version we only focus on bond distance and closed nature of the ring structures. In the chemical characterization segment of MAFIA-MD, the bond order is calculated from the trajectory files. This information can be used to differentiate the aromatics from the alicyclic hydrocarbons.

Finally, some, not all, of the outputs provided by MAFIA-MD can be obtained from the bonds information file. However, as mentioned earlier, the bond information files can be orders of magnitude larger in size than the trajectory (.XYZ files). The capability of post-processing without requiring these large bond information files gives more flexibility to the user during analysis and sharing data. Since the bonds file is not needed to be saved during runtime in order to extract the chemical information, the memory and I/O requirement for the RMD simulation can be made leaner with the help of `MAFIA-MD`. This can lead to faster runtime, easier collaboration and sharing of data (by only sharing small trajectory files). The ability to do batch processing on a number of molecular dynamics trajectories and get a lot of soot-relevant qualitative and quantitative information at the same time also makes `MAFIA-MD` very useful. There are some open-source tools available in the literature that can detect cyclic structures in MD simulation results [38, 39, 40], but these tools were not designed specifically for hydrocarbon systems and cannot analyze the fringe statistics, which is important for soot-relevant physics. To the best of our knowledge, `MAFIA-MD` is the first utility to do these large-scale analyses automatically and directly from the trajectory files. This makes `MAFIA-MD` unique and very useful to practitioners for analyzing the RMD simulations of hydrocarbons.

2. Theory and Algorithm

2.1. Identification of the cyclic structures

2.1.1. Read the Input Trajectory File

Table 1: Example file format for `XYZ` files

testFile.xyz	
Content	Description

N				Number of Atom, N
Timestep:	t			Timestamp , t
C	x	y	z	Atom identifier and coordinates of individual atoms
C	x	y	z	
H	x	y	z	
C	x	y	z	
H	x	y	z	
C	x	y	z	
⋮	⋮	⋮	⋮	⋮
H	x	y	z	
C	x	y	z	

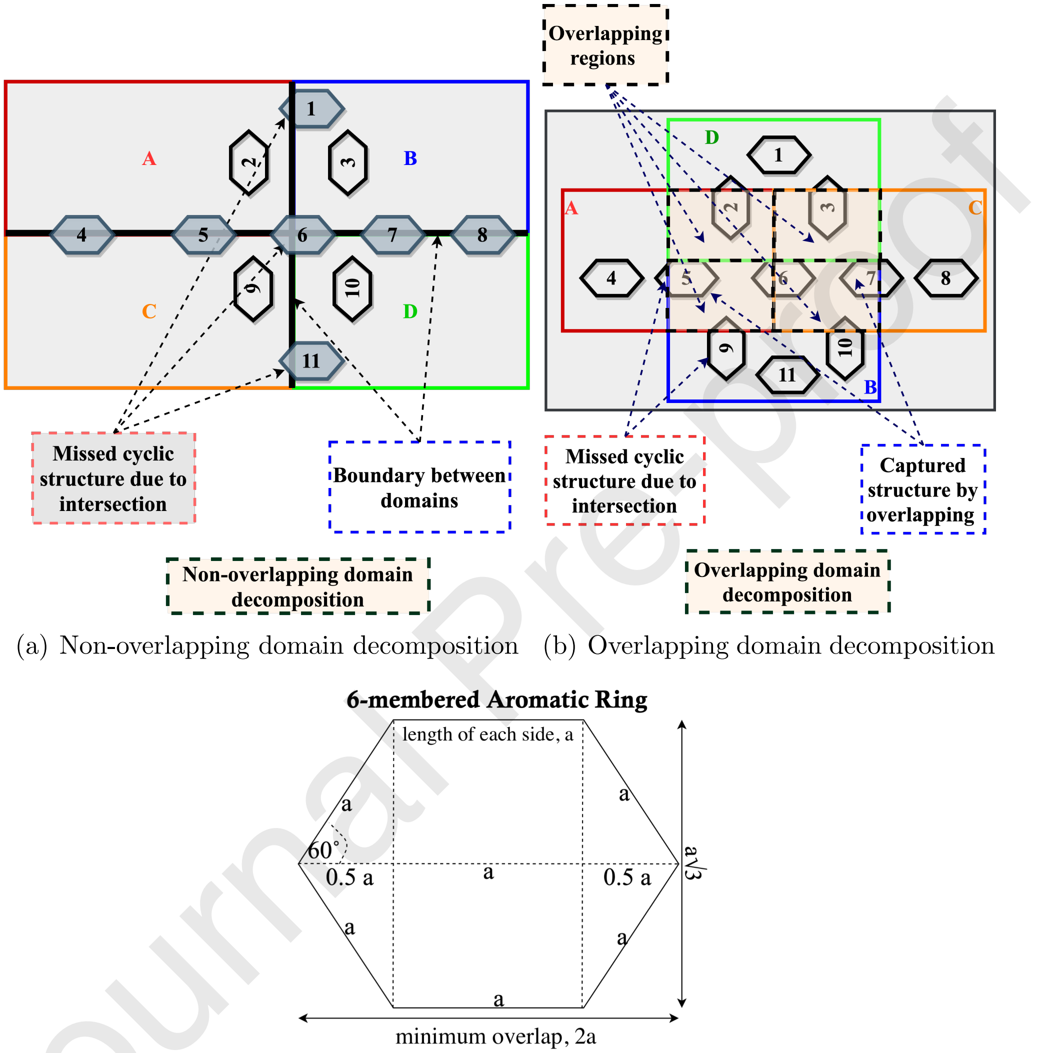
`MAFIA-MD` takes individual snapshot/trajectory files (in "`.XYZ`" format) from molecular dynamics simulation as the input. The "`.XYZ`" files contain coordinates of individual atoms existing in the domain as well as the total number of atoms and the timestamp. An example of the "`.XYZ`" file structure is presented in Table 1. In the current version, the ring-structure detection is carried out solely based on the carbon atoms in the domain. Therefore, once the "`.XYZ`" file is read, the hydrogen atoms are disregarded. For bookkeeping purpose, the timestamp and the total number of atom in the domain is also read.

2.1.2. Pre-Processing of Input Data

The input coordinates of the carbon atoms are sorted according to their distance from the global origin (0,0,0) first. The computational complexity of the overall system depends on the compactness (the degree of interconnectivity between the atoms) of the cluster of carbon atoms. If too many atoms are interconnected with each other (very compact cluster), the execution time will be inconveniently long. To remedy this situation, a divide and conquer approach [41] is implemented to reduce the computational complexity of the problem. In this approach, the computational domain is divided into a number of small spatially overlapping subdomains, each of which has significantly lower computational complexity than the entire domain. These overlapping subdomains are then analyzed sequentially.

The division or splitting of the computational domain in overlapping subdomains can be referred to as *overlapped domain decomposition*. The overlapping is required as some of the ring structures can be shared between multiple subdomains. If not properly accounted for, these shared rings will not be counted. This problem is shown in Fig.1(a) which presents non-overlapping (i.e., conventional) domain decomposition. It is evident from Fig. 1(a) that, even though the sub-domains A,B,C and D encircle all the ring structures among themselves, because of the presence of domain boundaries, they cannot identify all the cyclic structures. Therefore, the non-overlapping domain decomposition scheme misses cyclic structures 1, 4–8 and 11 as seen from Fig. 1(a).

The overlapping domain decomposition scheme implemented in `MAFIA-MD` (shown in Fig. 1(b)) remedies this. In Fig. 1(b), the whole computational domain is divided into five overlapping regions, i.e. A, B, C, D and the intersecting region between A, B, C, and D (referred to as subdomain ABCD). In this way, even though an individual domain misses some part of the ring structures (e.g., subdomain A misses cyclic structures 2, 6 and 9), other domains capture all of them (e.g., cyclic structures 2, 6 and 9 are captured


A
B
C
D
1
2
3
4
5
6
7
8
9
10
11
Missed cyclic structure due to intersection
Boundary between domains
Non-overlapping domain decomposition
Overlapping regions
Missed cyclic structure due to intersection
Captured structure by overlapping
Overlapping domain decomposition
6-membered Aromatic Ring
length of each side, a
a
60°
0.5 a
0.5 a
a√3
minimum overlap, 2a

(a) Non-overlapping domain decomposition (b) Overlapping domain decomposition

(c) Minimum amount of overlap for a 6-carbon ring

Figure 1: Different domain decomposition strategies (1(a) and 1(b)) and minimum amount of overlap required for a 6-carbon ring (1(c)).

by subdomains D, ABCD and B, respectively). It should be noted here, to remove any possible confusion, that the current version of the code is purely serial in nature. However, due to the independent nature of individual subdomains, the code can be easily parallelized to further speed up the analysis.

For the overlapping domain decomposition to work properly, the amount of overlap should be minimized to reduce unnecessary computations. The minimum amount of overlap necessary to capture all cyclic structures is equal to the maximum spatial footprint of the cyclic structures. A typical 6-membered ring structure is shown in Fig. 1(c), where "a" is the bond distance between two carbon atoms. This structure can be considered as a regular hexagon with sides of length a. As shown in the figure, the maximum spatial footprint of the 6-membered ring is twice the bond distance between two carbon atoms ($2a$). This is essentially the circumdiameter of a regular 6-memebered polygon. Similarly, for ensuring the detection of 5- and 7-membered rings, the minimum overlap should be at least 1.7 and 2.3 times the bond distance between two carbon atoms respectively. Accounting for the rectangular shape of the sub-domains, an overlap `span` of 8Åis found to be optimum for ensuring detection of all the cyclic structures of different sizes.

The overlapping domain decomposition scheme shown in Fig. 1(b), has a potential drawback for counting same cyclic structure multiple times as it can reside in multiple subdomains simultaneously. For example, in Fig. 1(b), cyclic structure 6 resides in both subdomains B and ABCD. Therefore, the book-keeping of these cyclic structure requires special handling. An array of global carbon indices (actual indices of carbon atoms in the `XYZ` file) is maintained for keeping track of unique cyclic structures from all the subdomains. By cross-referencing the global indices, the duplicate counting problem introduced by the overlapping domain decomposition scheme is resolved. Every time a ring is detected, the constituent carbon coordinates and their global indices are inserted into an array. This array of carbon indices from each subdomain is then concatenated into a global array which contains all the carbon indices from the identified cyclic structures. When all the subdomains are traversed, the duplicate entries are deleted from the global array retaining only the unique cyclic structures detected throughout the computational domain. The total number of 5-, 6-, and 7-membered rings is calculated using this global array.

For smaller (typically less than 700 carbon atoms, the actual number will depend on the interconnectedness of the atoms) population of atoms, the overlapped domain decomposition may not be necessary since the computational complexity is small enough to perform the ring detection in the entire domain without decomposition. In view of this, as well as to create an inbuilt verification tool, a separate functionality is implemented in `MAFIA-MD` named "`Sanity Check`" which, instead of decomposing the domain in several overlapped subdomains, considers the entire population of atoms as a whole. The user can chose to use this `Sanity Check` feature to either perform analysis on small atom population or to verify the accuracy of the overlapping domain decomposition by comparing the results from `Sanity Check` and overlapping domain decomposition. It should be noted here that, for larger cluster of atoms, the `Sanity Check` feature will take inconveniently long time to finish.

2.1.3. Creation of Directed Graphs and Determining the Elementary Circuits

Each subdomain carries the coordinates of carbon atoms inside the regions. Each point is sorted based on their distances from the global origin. Once the points are all sorted, a distance matrix [42] is calculated. The carbon atoms bonded with each other falls within a finite distances from one another. This distance is called the bond length. Single bond, double bond, triple bond and aromatic bonds between carbon atoms have different stable bond distances. Note that due to the atomic vibrations in the MD simulations, the bond length will not be the same as the exact theoretical value but will lie within a narow range around the theoretical bond length. Therefore, the upper and lower limits of bond length, instead of an exact value, need to be specified for the bond length. To capture the bonded carbon atoms, the element in the distance matrix is converted to "1" if the distance falls between the upper and lower limit of valid bond distances specified in the beginning of the analysis by user. If the distance falls outside this bound, the element in the matrix element is replaced with "0". This simplifies the distance matrix into an adjacency matrix [1], where each element with a value "1" represent a bond between two carbon atoms.

From the adjacency matrix generated in the earlier step, a directed graph is created using the open-source `Python` package "`NetworkX`" [43]. In a directed graph, the edges have direction and can have self-loops (a series of subsequent edges can point back to an earlier node). The main idea is to find the cyclic structures (i.e., ring structures) from the graph generated from the carbon atoms. For achieving this, the algorithm proposed by Johnson [3] is used for finding all elementary cycles from a directed graph. This algorithm is essentially a depth-first-search algorithm [2, 44], optimized to find only elementary circuits (one vertex appear only once) and traversing an edge at most twice. The algorithm identifies the sets of vertices for all the elementary circuits that forms a closed cycle or ring. For soot relevant applications, currently only 5-, 6-, and 7-membered rings are identified in `MAFIA-MD`. However, the framework for identifying larger rings and nested rings or supersets of rings (along with a check for planarity) is already implemented for any future development. Following the identification of elementary circuits, individual 5-, 6-, and 7-membered rings are indexed and counted for each subdomain and added globally to get the total number of rings containing 5-, 6-, and 7-carbons at a given timestep. The algorithm for ring detection is presented in Algorithm 1. It should be noted that, although all Huckel conditions for aromaticity are not checked in the ring detection segment of `MAFIA-MD`, the actual bond order is calculated in the "`Chemical Characterization`" segment (Sec. 2.2) of the code to distinguish the aromatics from the detected alicyclic structure.

2.2. Chemical characterization of atom clusters

The outputs from section 2.1 are used to interpret the chemical characteristics of the given soot cluster obtained from RMD simulation. For this purpose, a universal structural conversion algorithm developed by Kim and Kim [35] is implemented to solve for the atomic connectivity and bond order of the existing molecules. The general chemical rules and valency information are used for this purpose. Initially, each atomic pair is assigned a bond order based on their valencies. After that, the degree of bond saturation is assigned by trial and error until the whole system of atoms is solved. Once the bond order of the atomic connectivity/network is solved, the information is exported in terms of simplified molecular-input line-entry system (SMILES) [36] or into a molar file format (SDF) [37]. An external tool called `xyz2mol` [45] is modified to work with the current implementation of `MAFIA-MD`. `xyz2mol` uses "`RDKIT`" [46], an open source cheminformatics interface, to export the figures of the existing chemical structures inside the domain for visualization. The full functionality of the external tool "`xyz2mol`" is also kept in `MAFIA-MD` for batch processing of multiple `.XYZ` files at the same time by using the "`Chemistry Only`" functionality (see Sec. 3.4). This is particularly useful for the extraction of bond information from `.XYZ` files without requiring the large bond information files generated by RMD simulation.

Algorithm 1: Algorithm for ring identification and quantification.

```
Result: C/H ratio, percentage of alicyclics/aromatics, ring statistics
# Input:
Define the input parameters: upper and lower bond dinstance,
  span/overlapping distance, etc.
Open the trajectory (XYZ) file exported from RMD
Read the total number of atoms and timestamps from the trajectory
  (XYZ) file
# Pre-Processing:
Divide the entire domain into a number of subdomains
while NO subdomains are left do
    Create distance matrix using upper and lower bond limits;
    Simplify the distance matrix into an adjacency matrix;
    # Elementary Cycle Detection:
    Convert the adjacency matrix into a simply connected directed graph;
    Apply Johnson's algorithm [3] to extract all the simple cycles ;
    Separate the cycles containing 5-, 6- or 7- vertices;
    Create an array of sets containing the coordinates of all vertices of
      individual cycles;
    Count and index the carbon atoms present in the ring structure;
    Go to the next subdomain
end
Merge all rings from all subdomains;
Remove duplicate rings using global indices and extract the unique rings;
Pass the array of identified rings to the next segment of the code for
  optional chemical characterization and fringe analysis;
Print out the results;
# Output:
    1. C/H Ratio
    2. Percentage of alicyclics/aromatics
    3. Statistics of different rings
    4. Trajectory of alicyclic/aromatics and aliphatic carbons
```

2.3. Identification and analysis of molecular fringes

High Resolution Transmission Electron Microscopy (HRTEM) is often used to observe and characterize soot particles from different sources [32, 47, 48]. In HRTEM images, nano-structures of soot particles are observed and characterized using the characteristics of the fringes formed. Fringes are formed due to the optical interaction of internal nanostructures within soot clusters during microscopy. During HRTEM imaging, the fringes are usually characterized based on fringe length, fringe spacing, and fringe tortuosity. Figure 2 presents examples of the important characteristics of fringes obtained from HRTEM images of soot. The characteristics of these fringes convey important information about the reactivity and stability of soot clusters [32]. The short fringes (fringes 1, 2, and 4 of Fig. 2) have more free edges per unit length compared to the long fringes (fringe 3 of Fig. 2). The atoms in these free edges are reactive and therefore, soot clusters exhibiting shorter fringe lengths are more reactive. Similarly, the fringes with wider spacing (fringe 2 of Fig. 2) have more spaces between the molecules to diffuse oxygen molecules compared to the fringes with narrow spacing (fringes 1, 3, and 4 of Fig. 2). Therefore, the soot clusters producing narrowly spaced fringes are less prone to oxidation, hence less reactive. The tortuosity of fringes is a representation of the amount of curvature observed in optical fringes. Molecules with higher curvature produce fringes with high tortuosity (fringe 4 of Fig. 2) during HRTEM imaging. Due to the higher curvature, the bond strain is higher in these molecules and they break easily which results in higher reactivity.

The current version of `MAFIA-MD` implements a scheme to calculate the fringe spacing of a cluster of carbon molecules. The calculation for fringe spacing is based on the orientation of different cyclic structure in the domain. Two structure is assumed to form a fringe if the structures are in close proximity (3Å– 6Å) [49] and parallel to each other. With respect to the parallelity constraints, a deviation up to an angle Φ is allowed. The value of Φ is hard-coded in the code as $10°$. Algorithm 2 shows the algorithm implemented to calculate the fringe spacing histogram in `MAFIA-MD`. It is important to note that, for good statistics, a significant number of fringes must form in the soot cluster. Therefore, the analysis is only meaningful when the size of the cluster is large enough to contain multiple fringes.

The fringe statistics is presented in the form of histogram and probability density function (PDF) (shown in Fig. 7(b)). The fringe spacing histogram shows the distribution of fringes based on the distance between the individ-

Algorithm 2: Algorithm for calculating fringe spacing

```
Result: Fringe spacing histogram
# Input:
Coordinates of the carbon atoms inside the simulation domain;
List of all cyclic molecules in the domain from previous code segment;
# Pre-Processing:
Create a list of the coordinates of the centroids of each ring;
Create a list of all the vectors perpendicular to the existing rings through
  the centroids (surface vectors);
# Fringe spacing calculation:
fringeSpacing = [];
iter = 0 ;
while all the points in the centroid array is traversed do
    if $3Å \leq$ distance between two centroids $\leq 6Å$ then
        if ($0 \leq$ angle between the two surface vectors $\leq \Phi$) or
        ($180\text{-}\Phi \leq$ angle between the two surface vectors $\leq 180$) then
            fringeSpacing[iter] = distance between two centroids ;
            iter = iter +1 ;
        end
    end
end
Create histogram from $fingeSpacing$ array;
Estimate probability density function (PDF) of fringe spacing by kernel
density estimation with Gaussian kernels ;
# Output:
Histogram of fringe spacing;
Probability density function (PDF) of fringe spacing;
Fringe spacing vs. Angle;
```

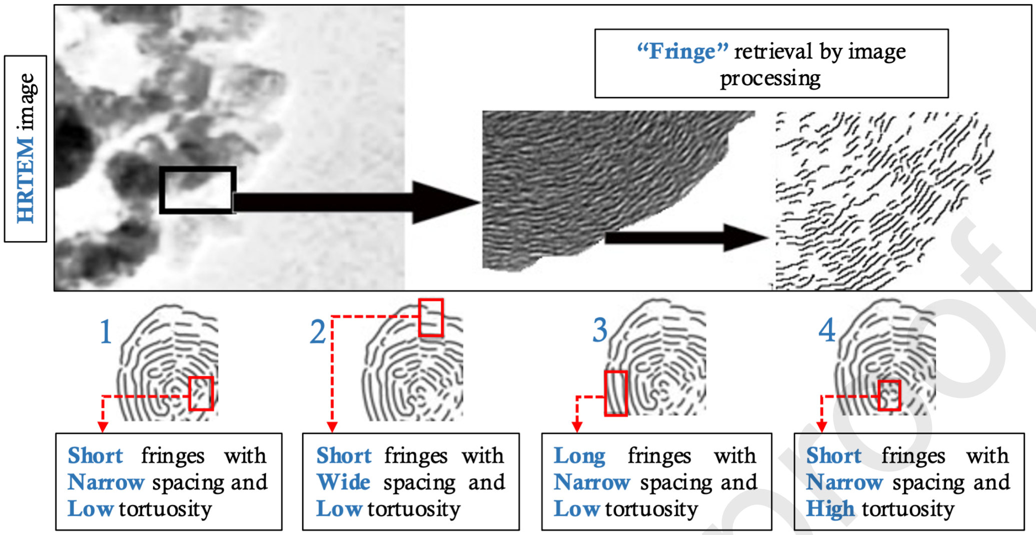

HRTEM image
"Fringe" retrieval by image processing
1
2
3
4
Short fringes with Narrow spacing and Low tortuosity
Short fringes with Wide spacing and Low tortuosity
Long fringes with Narrow spacing and Low tortuosity
Short fringes with Narrow spacing and High tortuosity

Figure 2: Examples and characterization of optical fringes obtained from a hypothetical HRTEM image of a soot particle.

ual fringes. The probability density function (PDF) of the fringe spacing distribution is calculated by kernel density estimation using Gaussian kernels [50] as it is known to work well for both uni-variate and multi-variate distributions [51]. MAFIA-MD utilizes the stats.gaussian_kde [52] function from python package scipy [53] to calculate the probability density function (PDF) of fringe spacing distribution.

3. Workflow

3.1. Code structure

The core functionality of MAFIA-MD is implemented using Python 3. The class FindRing() is created for identification of rings and subsequent calculations as indicated in Algorithm 1. For chemical characterization discussed in Sec. 2.2, it uses an external tool xyz2mol [45] to extract the chemical information (SMILE strings [36] and SDF [37] files). The characterization and analysis of fringes (Sec. 2.3) requires the identification of rings to be performed beforehand and can be thought of as an extension of the ring identification functionality. The external tool xyz2mol requires the open-source cheminformatics interface "RDKIT" [46], which in turn requires "anaconda"[54], an opensource package and environment management system for Python. The

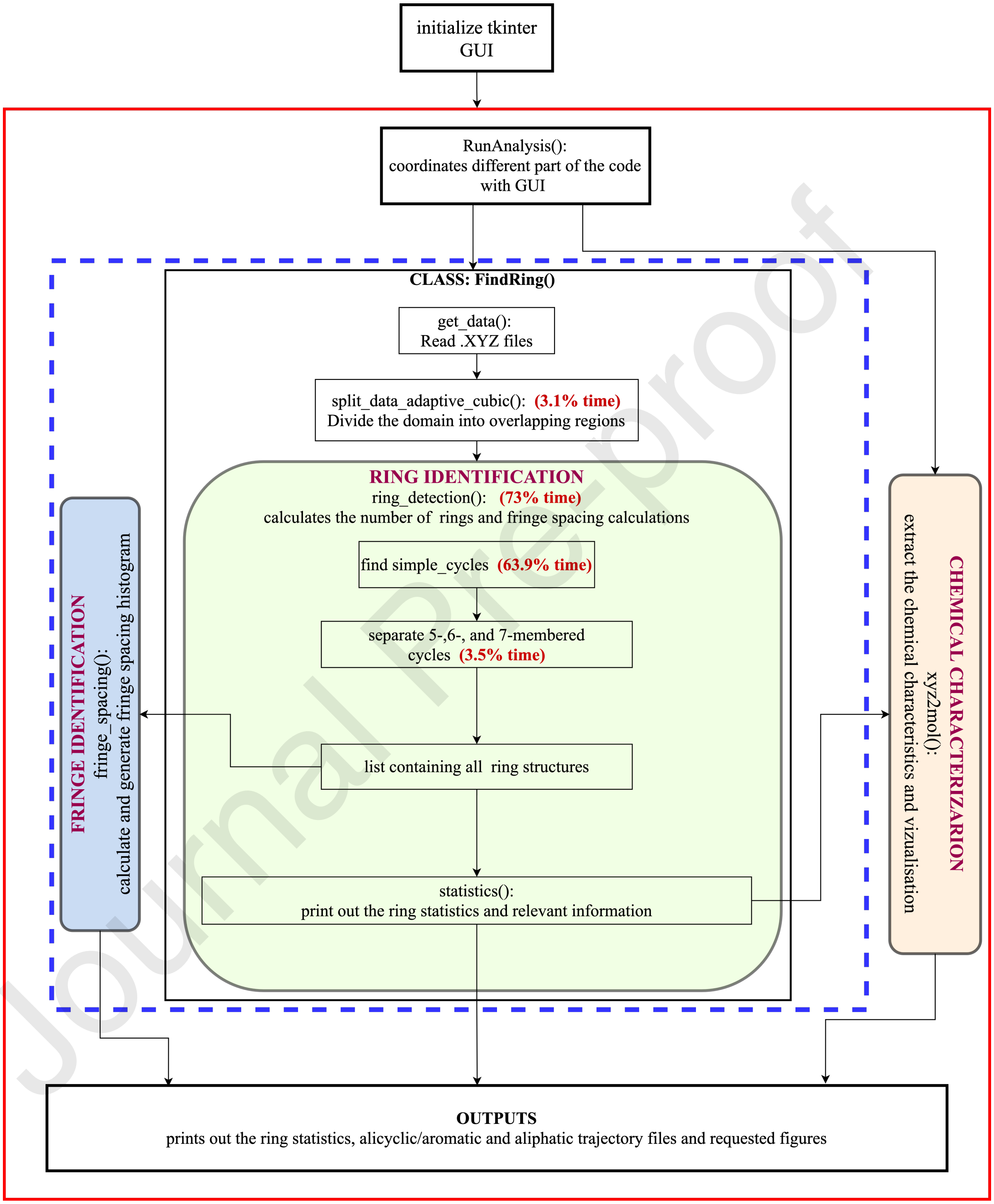

initialize tkinter GUI
RunAnalysis(): coordinates different part of the code with GUI
CLASS: FindRing()
get_data(): Read .XYZ files
split_data_adaptive_cubic(): (3.1% time)
Divide the domain into overlapping regions
RING IDENTIFICATION
ring_detection(): (73% time)
calculates the number of rings and fringe spacing calculations
find simple_cycles (63.9% time)
separate 5-,6-, and 7-membered cycles (3.5% time)
list containing all ring structures
statistics():
print out the ring statistics and relevant information
FRINGE IDENTIFICATION
fringe_spacing():
calculate and generate fringe spacing histogram
CHEMICAL CHARACTERIZARION
xyz2mol():
extract the chemical characteristics and vizualisation
OUTPUTS
prints out the ring statistics, alicyclic/aromatic and aliphatic trajectory files and requested figures

Figure 3: Code structure and relevant computational loads

two segments of the code (i.e., ring identification and chemical characterization) are connected using a graphical user interface (GUI) using tkinter [55].

The main computational complexity of the ring identification portion of the code comes from the detection of simple cycles from the interconnected network of carbon atoms. This portion of the code takes almost 64% time to execute for the given set of trajectory files. This complexity can either increase or decrease based on the complexity of the network of molecules. Figure 3 depicts the schematics of the code structure and computational loads of relevant functions used in the code. The callgraph for MAFIA-MD is shown in the Fig. 4.

3.2. Program deployment

Any operating system containing anaconda and Python 3 can run MAFIA-MD. The present code is tested on popular operating systems like Windows 10, Ubuntu (16.04, 18.04 and 20.04), Fedora 34, CentOS 8, Debian 10, MacOS (Catalina and Big Sur). The deployment procedure is described below:

1. Install conda or miniconda (installation instructions can be found in their respective websites [56, 57])

2. Download the MAFIA-MD repository. The installation files contain following files and directories

   - The parent directory:
     - mafiamd.py: The main Python code
     - Makefile: Makefile for Linux-like systems
     - requirements.yml: Build-requirements list
     - LICENSE: License file
     - README.md: Readme file
     - callgraph.png: The callgraph of the code
   - The external_tool directory: contains the xyz2mol.py package and its license
   - The input directory:
     - The set1_validation_demo directory: contains example files for validation (Sec. 3.5.2)

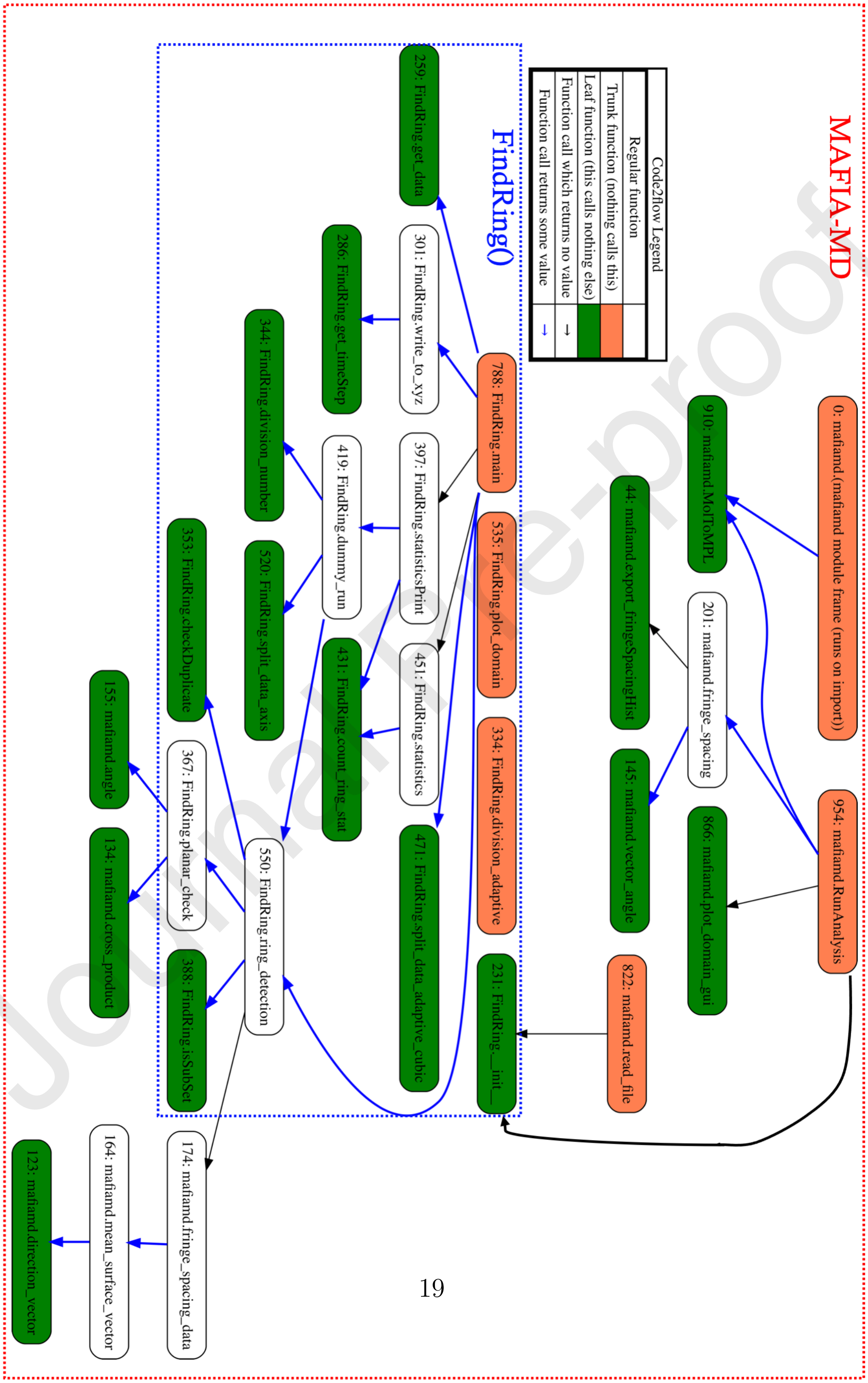

MAFIA-MD
Code2flow Legend
Regular function
Trunk function (nothing calls this)
Leaf function (this calls nothing else)
Function call which returns no value
Function call returns some value
0: mafiamd.(mafiamd module frame (runs on import))
954: mafiamd.RunAnalysis
910: mafiamd.MolToMPL
201: mafiamd.fringe_spacing
866: mafiamd.plot_domain_gui
44: mafiamd.export_fringeSpacingHist
145: mafiamd.vector_angle
822: mafiamd.read_file
FindRing()
788: FindRing.main
535: FindRing.plot_domain
334: FindRing.division_adaptive
231: FindRing.__init__
259: FindRing.get_data
301: FindRing.write_to_xyz
397: FindRing.statisticsPrint
451: FindRing.statistics
471: FindRing.split_data_adaptive_cubic
286: FindRing.get_timeStep
419: FindRing.dummy_run
431: FindRing.count_ring_stat
344: FindRing.division_number
520: FindRing.split_data_axis
353: FindRing.checkDuplicate
550: FindRing.ring_detection
367: FindRing.planar_check
388: FindRing.isSubSet
155: mafiamd.angle
134: mafiamd.cross_product
174: mafiamd.fringe_spacing_data
164: mafiamd.mean_surface_vector
123: mafiamd.direction_vector

Figure 4: Callgraph of MAFIA-MD

    - ○ The `set2_fringe_analysis_demo` directory: contains example files for fringe analysis (Sec. 3.5.3)
  - The `output` directory: a blank demo directory for storing output
  - The `ancillary_script` directory: contains an ancillary script for splitting continous trajectories files in discrete individual timesteps and a demo example trajectory file

3. Go to the code's parent directory and create a virtual environment

   - Linux and MacOS: Use the Makefile

     ```
     $ make
     ```

   - Windows: Use the `Anaconda Prompt` installed during the installation of `conda` or `miniconda` and execute

     ```
     $ conda env create -f requirements.yml -p
       MAFIAMD
     ```

4. Activate the `conda` environment

   ```
   $ conda activate ./MAFIAMD
   ```

5. Execute the code:

   ```
   $ python mafiamd.py
   ```

   This step (Step 5) will bring out the GUI (shown in Fig. 5) for specifying the analysis parameters discussed in Sec. 3.4. The execution time of the program can range from a few seconds to a few hour for each trajectory file depending on the size and complexity (i.e., interconnectivity of the carbon atoms) of the network. Once the execution is completed, close the program by using the `Quit` button on the GUI.

6. to deactivate the conda enviroment after execution

   ```
   $ conda deactivate
   ```

3.3. Analysis modes

The user can perform the following types of analyses in `MAFIA-MD`:

1. Detection of cyclic structures and analysis: This is the main operational mode of `MAFIA-MD`. This mode will analyze the provided trajectory files and generate relevant information like ring statistics, C/H ratio, percentage of cyclic carbon atoms, etc. The result can be visualized in GUI by selecting the `Plot` option.

2. Sanity check: This mode is triggered by selecting the `Sanity Check` option. In this mode, the non-overlapping domain decomposition is performed *along with* the overlapping domain decomposition scheme from step 1.

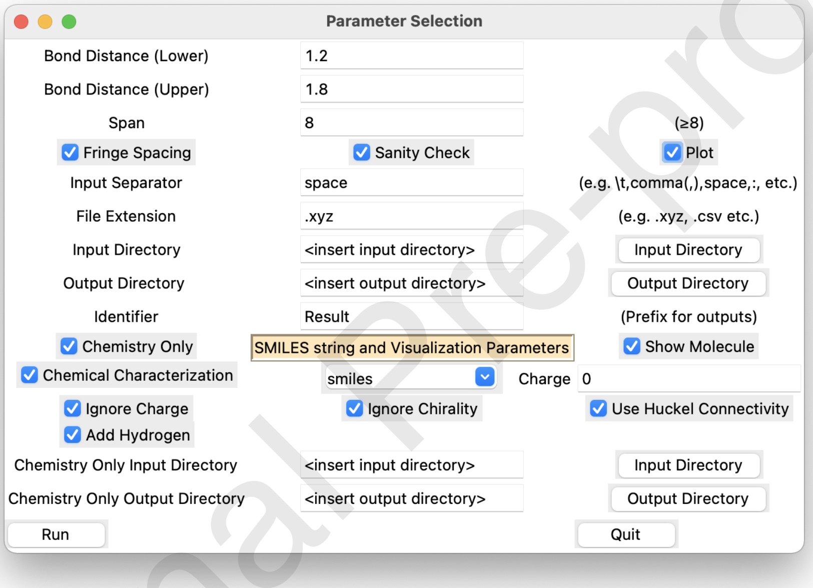
3. Chemical characterization: This mode is triggered by selecting the `CHEM Calculation` option. This will perform step 1 and generate chemical information from the results in terms of SMILES or SDF files. The user can also visualize the existing alicyclic and aromatic rings by selecting the `Show Molecule` option.

4. Chemistry only: This mode is performed when only the chemical characterization of trajectory files are required and can be enabled by selecting the `Chemistry Only` option in the GUI. This will analyze the entire input trajectory and perform chemical characterization (step 3) without performing the ring detection analysis from step 1.

5. Fringe analysis: The user can use this functionality by selecting the `Fringe Spacing` option in GUI. This will perform step 1 and generate fringe spacing histogram for the input trajectory files along with the probability density function (PDF) calculated using the Gaussian kernel density estimation [50].

3.4. Program input

- **Bond Distance (Lower)**: minimum (lower limit) carbon-carbon bond distance (in Å). The default value is 1.2. This is an optional argument. While specifying an unrealistically low value does not change the outcome of `MAFIA-MD`, doing so often increases the runtime of `MAFIA-MD` by a small amount. If the text-box is left empty, MAFIA-MD will assume a zero value for the lower limit of bond distance.



Parameter Selection

Bond Distance (Lower) 1.2

Bond Distance (Upper) 1.8

Span 8 (≥8)

☑ Fringe Spacing ☑ Sanity Check ☑ Plot

Input Separator space (e.g. \t,comma(,),space,:, etc.)

File Extension .xyz (e.g. .xyz, .csv etc.)

Input Directory <insert input directory> Input Directory

Output Directory <insert output directory> Output Directory

Identifier Result (Prefix for outputs)

☑ Chemistry Only SMILES string and Visualization Parameters ☑ Show Molecule

☑ Chemical Characterization smiles Charge 0

☑ Ignore Charge ☑ Ignore Chirality ☑ Use Huckel Connectivity

☑ Add Hydrogen

Chemistry Only Input Directory <insert input directory> Input Directory

Chemistry Only Output Directory <insert output directory> Output Directory

Run Quit

Figure 5: Program graphical user interface

- **Bond Distance (Upper); required**: maximum (upper limit) carbon-carbon bond distance (in Å). The default value is 1.8.

- **Span; required**: twice the amount of overlap between two consecutive subdomains (in Å). A span of 8Åis found to be optimum for finding upto 7-membered rings. Therefore, the default value is 8.

- **Input Separator; required**: separator used between (X,Y,Z) coordinates in the input `XYZ` files. The default value is `space` (whitespace). The output trajectory files generated by the code is always tab separated.

- **File Extension; required**: extension of the input trajectory files. This option is kept for the cases when the input trajectory files are provided in comma separated format (.csv). The default value is `.xyz` (with the dot).

- **Input Directory; required**: directory containing the input trajectory files. The code will analyze *all* files kept in the specified input directory.

- **Output Directory; required**: directory where the output files (e.g., logs and plots) will be saved.

- **Identifier; required**: a text string prepended to each output file-name to differentiate between the separate instances of `MAFIA-MD` runs. The default value is `Result`.

- **Charge**: useful for `Chemistry Only` runs where a trajectory file contains a net charge.

- **Chemistry Only Input Directory**: directory containing the input trajectory files for `Chemistry Only` analysis.

- **Chemistry Only Output Directory**: directory where the output files will be saved for `Chemistry Only` analysis.

- **Analysis mode options:**

  - *Sanity Check:* when selected, the sanity check segment of the code will be executed along with the overlapping domain decomposition scheme.

- *Plot:* when selected, the existing rings in the domain will be plotted in a separate window
- *Fringe Spacing:* when selected, fringe spacing histogram is calculated and printed out.
- *Chemical Characterization:* when selected, the chemical characterization will be performed along with the ring analysis. Select "`smiles`" to get a SMILE string for the existing molecule or "`sdf`" to get individual SDF files representing the intput trajectories. The following sub-options are available for chemical analysis.
  - *Show Molecule:* when selected, the ring structures found in the domain will be drawn on a separate window using `RDKIT`.
  - *Ignore Charge:* when selected, the chemical calculation will not consider the effect of net charge during calculation
  - *Ignore Chirality:* when selected, the chemical calculation will not consider the effect of chirality during calculation. This is a functionality of the external tool `xyz2mol`.
  - *Use Huckel Connectivity:* when selected, the chemical calculation will use extended Huckel bond orders to locate bonds during calculation. Otherwise, van der Waals radii will be used. This is a functionality of the external tool `xyz2mol`.
  - *Add Hydrogen:* when selected, the output images will also contain hydrogen atoms. The default is to plot only carbon atoms.
- *Chemistry Only:* when selected, only the chemical analysis will be performed on the trajectory files considering both hydrogen and carbon atoms and it will not perform any ring identification. It is helpful for visualizing the general chemical structures inside the domain.

3.5. Numerical example

3.5.1. Example cases

Two sets of input trajectories, included with the code, are used to demonstrate the capabilities of `MAFIA-MD`. The first set contains three relatively small trajectories for the purpose of validation of the ring detection method. These three test trajectories are `fabricated.xyz`, `fabricated2.xyz`, and real_MD.xyz and are kept in one single directory. These three input trajectory files are whitespace separated. Two of these are fabricated (fabricated.xyz and fabricated2.xyz) and the other one (real_MD.xyz) is selected from an actual molecular dynamics study performed by Sharma et al. [21] where an earlier version of MAFIA-MD [58] was used. The actual molecular dynamics trajectory file real_MD.xyz has 760 atoms. The fabricated trajectories are prepared using openbabel [59]. The details of these trajectories are

1. fabricated.xyz: 58 carbon atoms and 96 hydrogen atoms

   - Designed ring count:
     5-membered: 4, 6-membered: 8, 7-membered: 1

2. fabricated2.xyz: 110 carbon atoms and 190 hydrogen atoms

   - Designed ring count:
     5-membered: 5, 6-membered: 7, 7-membered: 2

3. real_MD.xyz: 559 carbon atoms and 201 hydrogen atoms

   - Manually counted ring count:
     5-membered: 14, 6-membered: 20, 7-membered: 6

In a second directory the second set of test case is kept. This set consists of a single large trajectory file named 1769atoms.xyz (1256 carbon atoms and 513 hydrogen atoms), which is also selected from [21]. This file is tab separated. Due to the size of this trajectory, it was not possible to manually count the rings in this trajectory for validation. This trajectory is used to show the capabilities of fringe analysis.

3.5.2. Validation tests

In the first set of tests, the directory containing the first set of trajectories (fabricated.xyz, fabricated2.xyz, and real_MD.xyz) was selected as input directory and MAFIA-MD performed analysis of all three trajectories simultaneously. The parameter selection and output for this set of trajectory files are shown in Fig. 6 and the output text file containing the result is shown in Listing 1. The output file contains all the requested information and it correctly determines the exact number of rings present in the supplied trajectories, thereby validating the code.

Listing 1: Output file

```
--------------------------------------------------------------------------------------------------
1:      fabricated.xyz
--------------------------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:       0.6041666666666666
Total Alicyclic/Aromatic Carbon:         52
Total Aliphatic Carbon Number:   6
Total Existing Rings     {7: 1, 5: 4, 6: 8}
Percentage of Alicyclic/Aromatic components     : 0.896551724137931
Percentage of Aliphatic components      : 0.10344827586206895
--------------------------------------------------------------------------------------------------
1:      Sanity Check for:       fabricated.xyz
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
Starting code for Axis:          0
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:       0.6041666666666666
Total Alicyclic/Aromatic Carbon:         52
Total Aliphatic Carbon Number:   6
Total Existing Rings     {5: 4, 6: 8, 7: 1}
Percentage of Alicyclic/Aromatic components     : 0.896551724137931
Percentage of Aliphatic components      : 0.10344827586206895
--------------------------------------------------------------------------------------------------
1:      smiles_fabricated       : START
--------------------------------------------------------------------------------------------------
[C]1=C=C=C=C=C2[C]=C=C=C12.[C]1=c2c([c]c3[c]c4c(c5[c]c#cc2c35)=C=C=C=4)C#C1.[C]1C#CC2=c3c(c4[c]c#
     cc5c4c4c(c#c[c]c34)=C=C=5)=C=C=C12
--------------------------------------------------------------------------------------------------
1:      smiles_fabricated       : END
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
2:      fabricated2.xyz
--------------------------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:       0.5789473684210527
Total Alicyclic/Aromatic Carbon:         69
Total Aliphatic Carbon Number:   41
Total Existing Rings     {5: 5, 6: 7, 7: 2}
Percentage of Alicyclic/Aromatic components     : 0.6272727272727273
Percentage of Aliphatic components      : 0.3727272727272727
--------------------------------------------------------------------------------------------------
2:      Sanity Check for:       fabricated2.xyz
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
Starting code for Axis:          0
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:       0.5789473684210527
Total Alicyclic/Aromatic Carbon:         69
Total Aliphatic Carbon Number:   41
Total Existing Rings     {6: 7, 5: 5, 7: 2}
Percentage of Alicyclic/Aromatic components     : 0.6272727272727273
Percentage of Aliphatic components      : 0.3727272727272727
--------------------------------------------------------------------------------------------------
2:      smiles_fabricated2      : START
--------------------------------------------------------------------------------------------------
C1=C=C=c2c#cc#cc2=C=1.C1=C=C=c2c3[c]c#cc-3[c]c#cc2=C=1.[C]1=C2C(=C=C=c3c2[c]c(=C2C#CC#C2)c2c3=C=C=C
     =2)C#CC#C1.[C]1=C=C=C1C1=C=C=[C]C#C1.[C]1C#CC#C1.[C]1C#CC#C1
--------------------------------------------------------------------------------------------------
2:      smiles_fabricated2      : END
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
3:      real_MD.xyz
--------------------------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:       2.7810945273631837
Total Alicyclic/Aromatic Carbon:         158
Total Aliphatic Carbon Number:   401
Total Existing Rings     {5: 14, 6: 20, 7: 6}
```

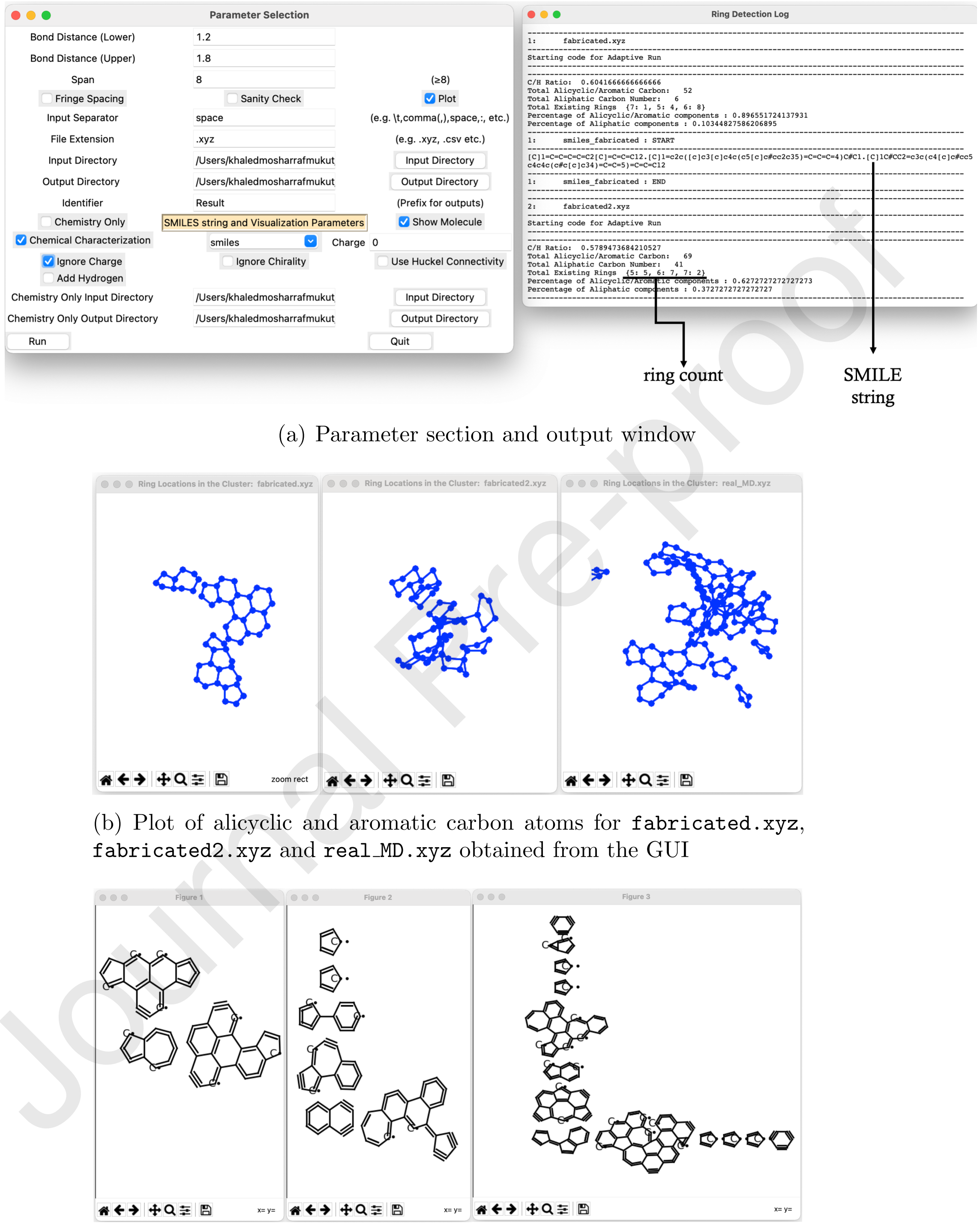
```
Percentage of Alicyclic/Aromatic components     : 0.2826475849731664
Percentage of Aliphatic components      : 0.7173524150268336
----------------------------------------------------------------------------------------------
3:      Sanity Check for:       real_MD.xyz
----------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
Starting code for Axis:          0
----------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
C/H Ratio:       2.7810945273631837
Total Alicyclic/Aromatic Carbon:         158
Total Aliphatic Carbon Number:   401
Total Existing Rings     {6: 20, 5: 14, 7: 6}
Percentage of Alicyclic/Aromatic components     : 0.2826475849731664
Percentage of Aliphatic components      : 0.7173524150268336
----------------------------------------------------------------------------------------------
3:      smiles_real_MD  : START
----------------------------------------------------------------------------------------------
C1#CC(=C2C#Cc3c#cc#cc32)C#C1.C1=C=c2[c]c3c#cc4c#cc5[c]c6c#cc=1c6c2c3c45.[C]1=C=C=C2C#CC3=[C][C@@]34C3
     =C=C=c5c#cc6c7c5c3c(c3[c]c5[c]c#cc8[c]c9c#c[c]c%10c(c(c37)-c(c85)c9%10)=C=C=6)C1=C24.[C]1=C=C=
     C2C#C[C]=C=C12.[C]1=C=C=c2[c]c3[c]c4c#cc#cc4[c]c4c5c6c(c#cc#cc6c(c21)c34)=C=C=C=5.[C]1C#CC#C1.[C
     ]1C#CC#C1.[C]1C#CC#C1.[C]1C#CC#C1.[C]1C#CC#C1.c1[c]c2[c]c-2c#1.c1c#cc#cc#1.c1c#cc#cc#1
----------------------------------------------------------------------------------------------
3:      smiles_real_MD  : END
----------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
```

3.5.3. Demonstration of fringe analysis

The second test is done on the input directory containing only `1769atoms.xyz`. The fringe analysis capability is demonstrated in this test. The GUI for fringe spacing calculation mode and the output from `MAFIA-MD` is presented in Fig. 7(a). The output contains the fringe histogram and a list of fringe spacing vs. angle between the surface normal of two rings (to show the parallelity of the planes). The fringe spacing histogram generated by `MAFIA-MD` is shown in Fig. 7(b). The figure for the fringe spacing histogram with probability density function (PDF) and a complete output log is stored in the user specified output directory along with the graphical output shown in Fig. 7. The output log for fringe spacing calculation is shown in Listing 2.

Listing 2: Fringe spacing calculation mode output log

```
----------------------------------------------------------------------------------------------
1:      1769atoms.xyz
----------------------------------------------------------------------------------------------
Starting code for Adaptive Run
----------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
C/H Ratio:       2.448343079922027
Total Alicyclic/Aromatic Carbon:         747
Total Aliphatic Carbon Number:   509
Total Existing Rings     {6: 103, 5: 42, 7: 54}
Percentage of Alicyclic/Aromatic components     : 0.5947452229299363
Percentage of Aliphatic components      : 0.40525477707006374
----------------------------------------------------------------------------------------------
1:      Fringe Spacing_1769atoms        : START
----------------------------------------------------------------------------------------------


Fringe spacing: Fringe Spacing vs Angle

5.950012676836169        3.9242557263703115
5.290388731629016        0.0
4.629775482026949        5.429560409174904
4.419991063508438        9.246261423942366
```



Parameter Selection

Bond Distance (Lower) 1.2

Bond Distance (Upper) 1.8

Span 8 (≥8)

□ Fringe Spacing □ Sanity Check ☑ Plot

Input Separator space (e.g. \t,comma(,),space,:, etc.)

File Extension .xyz (e.g. .xyz, .csv etc.)

Input Directory /Users/khaledmosharrafmukut

Output Directory /Users/khaledmosharrafmukut

Identifier Result (Prefix for outputs)

□ Chemistry Only SMILES string and Visualization Parameters ☑ Show Molecule

☑ Chemical Characterization smiles Charge 0

☑ Ignore Charge □ Ignore Chirality □ Use Huckel Connectivity

□ Add Hydrogen

Chemistry Only Input Directory /Users/khaledmosharrafmukut

Chemistry Only Output Directory /Users/khaledmosharrafmukut

Ring Detection Log

```
--------------------------------------------------------------------------------------------------
1:      fabricated.xyz
--------------------------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:  0.6041666666666666
Total Alicyclic/Aromatic Carbon:   52
Total Aliphatic Carbon Number:   6
Total Existing Rings  {7: 1, 5: 4, 6: 8}
Percentage of Alicyclic/Aromatic components : 0.896551724137931
Percentage of Aliphatic components : 0.10344827586206895
--------------------------------------------------------------------------------------------------
1:      smiles_fabricated : START
--------------------------------------------------------------------------------------------------
[C]1=C=C=C=C=C2[C]=C=C=C12.[C]1=c2c([c]c3[c]c4c(c5[c]c#cc2c35)=C=C=C=4)C#C1.[C]1C#CC2=c3c(c4[c]c#cc5
c4c4c(c#c[c]c34)=C=C=5)=C=C=C12
--------------------------------------------------------------------------------------------------
1:      smiles_fabricated : END
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
2:      fabricated2.xyz
--------------------------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
C/H Ratio:  0.5789473684210527
Total Alicyclic/Aromatic Carbon:   69
Total Aliphatic Carbon Number:   41
Total Existing Rings  {5: 5, 6: 7, 7: 2}
Percentage of Alicyclic/Aromatic components : 0.6272727272727273
Percentage of Aliphatic components : 0.3727272727272727
--------------------------------------------------------------------------------------------------
```

(a) Parameter section and output window

Ring Locations in the Cluster: fabricated.xyz
Ring Locations in the Cluster: fabricated2.xyz
Ring Locations in the Cluster: real_MD.xyz
zoom rect

(b) Plot of alicyclic and aromatic carbon atoms for `fabricated.xyz`, `fabricated2.xyz` and `real_MD.xyz` obtained from the GUI

(c) Molecules extracted from `fabricated.xyz`, `fabricated2.xyz` and `real_MD.xyz` respectively from left to right

Figure 6: Execution and output of `MAFIA-MD`

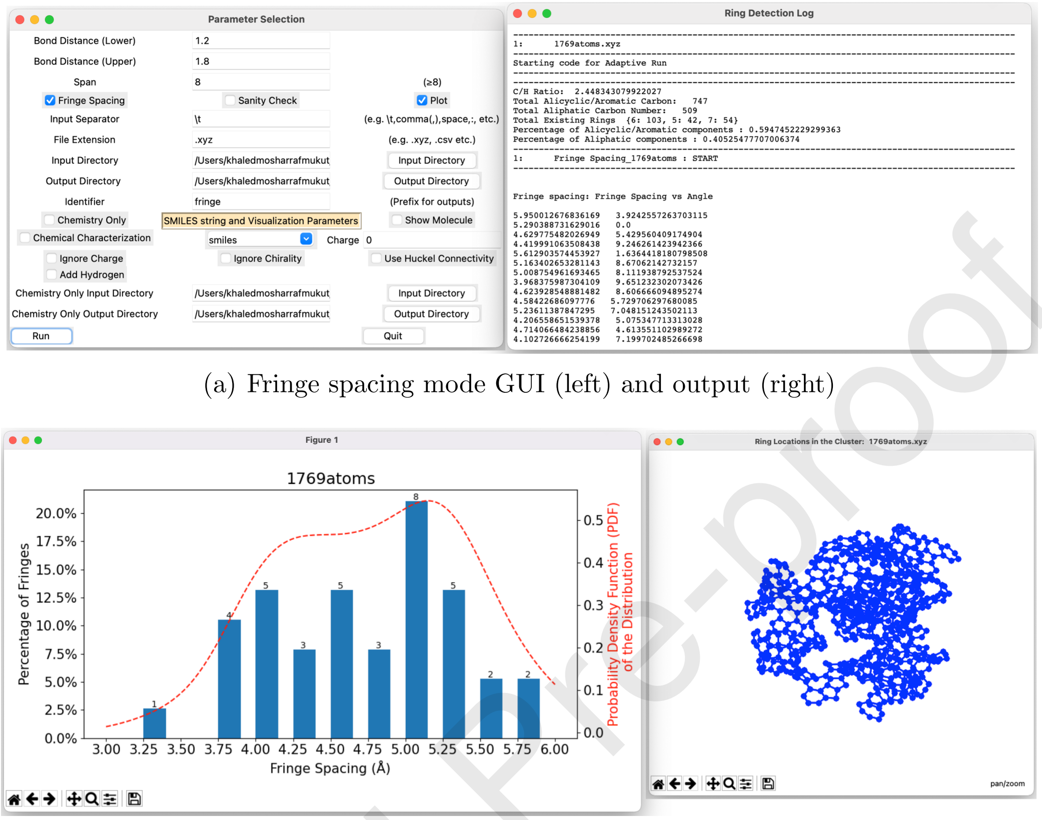
```
5.612903574453927        1.6364418180798508
5.163402653281143        8.67062142732157
5.008754961693465        8.111938792537524
3.968375987304109        9.651232302073426
4.623928548881482        8.606666094895274
4.58422686097776         5.729706297680085
5.23611387847295         7.048151243502113
4.206558651539378        5.075347713313028
4.714066484238856        4.613551102989272
4.102726666254199        7.199702485266698
4.185954445820429        8.694128830794705
5.288683680451116        8.082443661468403
5.5246824885358565      9.121954527071836
3.9753826457241774      5.869899451671865
4.574647618386823        8.868308926350613
5.368046285546225        6.273100098502783
5.228879733893831        4.206006987785507
4.792224005753124        6.96472208665477
5.173526348729457        175.53685851260337
4.857647329007471        176.0514868131242
4.035298892998304        173.28483345021596
4.3219796323389685      170.746229983982
5.409863373609561        9.259381236595647
4.175217901573789        180.0
3.988760975726621        178.47288283324156
5.333240479100813        171.310888466995
5.126083238875523        8.463148316690887
3.979080838899854        8.951229787825891
5.055383120720362        5.3254081727125575
4.3724567234374465      5.943245793999107
4.783263574854577        5.033070521661905
5.869554203719395        5.342639370384733
5.239332298840521        9.795728302807293
3.4251790290727895      9.152348900213486

Fringe Spacing Histogram

Bins:    [3.   3.25 3.5  3.75 4.   4.25 4.5  4.75 5.   5.25 5.5  5.75 6.  ]
Hist:    [0 1 0 4 5 3 5 3 8 5 2 2]

Fringe Spacing Histogram

End of Fringe spacing


---------------------------------------------------------------------------------------------------
1:      Fringe Spacing_1769atoms        : END
---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------
```

3.5.4. Computational cost

The main computational cost of MAFIA-MD is in the ring detection part of the code as indicated in Fig. 3. The total computational cost of MAFIA-MD for the ring detection, chemical analysis and fringe spacing calculation for the example trajectory files when used on a system with intel 2.3GHz Core i5-8259U CPU is presented in Table 2.

4. Future works

The post-processing tool MAFIA-MD presented in this manuscript is capable of analyzing molecular dynamics trajectory files for hydrocarbon reactions. The current implementation is not parallelized. From Fig. 3, we see



Parameter Selection

Bond Distance (Lower) 1.2

Bond Distance (Upper) 1.8

Span 8 (≥8)

☑ Fringe Spacing ☐ Sanity Check ☑ Plot

Input Separator \t (e.g. \t,comma(,),space,:, etc.)

File Extension .xyz (e.g. .xyz, .csv etc.)

Input Directory /Users/khaledmosharrafmukut

Output Directory /Users/khaledmosharrafmukut

Identifier fringe (Prefix for outputs)

☐ Chemistry Only ☐ Show Molecule

☐ Chemical Characterization smiles Charge 0

☐ Ignore Charge ☐ Ignore Chirality ☐ Use Huckel Connectivity

☐ Add Hydrogen

Chemistry Only Input Directory /Users/khaledmosharrafmukut

Chemistry Only Output Directory /Users/khaledmosharrafmukut

Ring Detection Log

```
--------------------------------------------------------------------------------
1:      1769atoms.xyz
--------------------------------------------------------------------------------
Starting code for Adaptive Run
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
C/H Ratio:  2.448343079922027
Total Alicyclic/Aromatic Carbon:   747
Total Aliphatic Carbon Number:   509
Total Existing Rings  {6: 103, 5: 42, 7: 54}
Percentage of Alicyclic/Aromatic components : 0.5947452229299363
Percentage of Aliphatic components : 0.40525477707006374
--------------------------------------------------------------------------------
1:      Fringe Spacing_1769atoms : START
--------------------------------------------------------------------------------


Fringe spacing: Fringe Spacing vs Angle

5.950012676836169   3.9242557263703115
5.290388731629016   0.0
4.629775482026949   5.429560409174904
4.419991063508438   9.246261423942366
5.612903574453927   1.6364418180798508
5.163402653281143   8.67062142732157
5.008754961693465   8.111938792537524
3.968375987304109   9.651232302073426
4.623928548881482   8.606666094895274
4.58422686097776   5.729706297680085
5.23611387847295   7.048151243502113
4.206558651539378   5.075347713313028
4.714066484238856   4.613551102989272
4.102726666254199   7.199702485266698
```

(a) Fringe spacing mode GUI (left) and output (right)

(b) Fringe spacing histogram (left) and rings inside domain (right)

Figure 7: Fringe spacing calculation mode outputs.

Table 2: Computational cost breakdown of MAFIA-MD

Trajectory Files		Computational Cost (s)		
Name	Composition	Ring Detection	Chemical Analysis	Fringe Spacing
`fabricated.xyz`	58 Carbon 96 Hydrogen	0.2109	0.0201	NA
`fabricated2.xyz`	110 Carbon 190 Hydrogen	0.2475	0.0313	NA
`real_md.xyz`	559 Carbon 201 Hydrogen	1.6093	0.0944	NA
`1769atoms.xyz`	1256 Carbon 513 Hydrogen	5.0159	1.5683	1.0928

that the most computationally intensive (64%) part of the code is the part where it identifies the rings from the network of carbon atoms. The cycle-finding segment of the code is parallelizable and doing so will increase the functionality of `MAFIA-MD` in analyzing very large atomic clusters and reduce the runtime. The overlapping regions do not have any dependency on each other and therefore can take the advantage of many core systems like GPUs. This kind of optimization can enable real-time analysis of large scale reactive molecular dynamics simulation. The authors will continue working toward this goal in the future.

5. Summary

A useful post-processing utility called `MAFIA-MD` is presented in this manuscript for analyzing reactive molecular dynamics trajectory of hydrocarbon molecules. `MAFIA-MD` is written in `Python-3.7` and has been tested successfully in Windows, Linux, and MacOS. `MAFIA-MD` extracts the alicyclic and aromatic hydrocarbons by identifying cyclic structures which are important in understanding complex physico-chemical phenomena in soot formation and growth. It also extracts the relevant soot related chemical information like C/H ratio and aliphatic to alicyclic/aromatic carbon ratio, etc. A methodology of calculating fringe spacing is implemented as well for diagnostic studies. `MAFIA-MD` have the ability to work with large number of simple trajectory files generated by reactive molecular dynamics simulation simultaneously. This capability of batch processing large number of files makes `MAFIA-MD` very useful for researchers working on large scale reactive molecular dynamics simulation.

Acknowledgements

The authors would like thank Md. Hasanul Aziz for his guidance toward the development of MAFIA-MD.

References

[1] G. Turán, On the succinct representation of graphs, Discrete Applied Mathematics 8 (1984) 289–294.

[2] D. Kozen, The Design and Analysis of Algorithms, Springer-Verlag, New York, NY, USA, 1992.

[3] D. B. Johnson, Finding All the Elementary Circuits of a Directed Graph, SIAM Journal on Computing (2006).

[4] S. A. Hollingsworth, R. O. Dror, Molecular dynamics simulation for all, Neuron 99 (2018) 1129.

[5] X. Li, A. Wang, K.-R. Lee, Comparison of empirical potentials for calculating structural properties of amorphous carbon films by molecular dynamics simulation, Computational Materials Science 151 (2018) 246–254.

[6] P. Erhart, K. Albe, Analytical potential for atomistic simulations of silicon, carbon, and silicon carbide, Physical Review B 71 (2005) 035211.

[7] J. Tersoff, New empirical approach for the structure and energy of covalent systems, Physical Review B 37 (1988) 6991–7000.

[8] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, S. B. Sinnott, A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons, Journal of Physics: Condensed Matter 14 (2002) 783–802.

[9] D. W. Brenner, Erratum: Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films, Physical Review B 46 (1992) 1948.

[10] T. C. O'Connor, J. Andzelm, M. O. Robbins, AIREBO-M: A reactive model for hydrocarbons at extreme pressures, Journal of Chemical Physics 142 (2015) 024903.

[11] A. C. T. van Duin, S. Dasgupta, F. Lorant, W. A. Goddard, ReaxFF: A Reactive Force Field for Hydrocarbons, Journal of Physical Chemistry A 105 (2001) 9396–9409.

[12] N. A. Marks, Thin film deposition of tetrahedral amorphous carbon: a molecular dynamics study, Diamond & Related Materials 8 (2005) 1223–1231.

[13] D. Zhang, M. R. Dutzer, T. Liang, A. F. Fonseca, Y. Wu, K. S. Walton, D. S. Sholl, A. H. Farmahini, S. K. Bhatia, S. B. Sinnott, Computational investigation on CO2 adsorption in titanium carbide-derived carbons with residual titanium, Carbon 111 (2017) 741–751.

[14] X. Li, P. Ke, H. Zheng, A. Wang, Structural properties and growth evolution of diamond-like carbon films with different incident energies: A molecular dynamics study, Applied Surface Science Complete (2013) 670–675.

[15] V. S. Dozhdikov, A. Y. Basharin, P. R. Levashov, Structure of amorphous carbon quenched from liquid in the pressure range 1–40 GPa: Molecular dynamic modeling, Journal of Physics: Conference Series 946 (2018) 012086.

[16] T.-B. Ma, L.-F. Wang, Y.-Z. Hu, X. Li, H. Wang, A shear localization mechanism for lubricity of amorphous carbon materials - Scientific Reports, Scientific Reports 4 (2014) 1–6.

[17] M. Joe, M.-W. Moon, J. Oh, K.-H. Lee, K.-R. Lee, Molecular dynamics simulation study of the growth of a rough amorphous carbon film by the grazing incidence of energetic carbon atoms, Carbon 50 (2012) 404–410.

[18] M. Gastegger, P. Marquetand, Molecular Dynamics with Neural Network Potentials, in: Machine Learning Meets Quantum Physics, Springer, Cham, Switzerland, 2020, pp. 233–252.

[19] B. Curchod, T. J. Martínez, Ab Initio Nonadiabatic Quantum Molecular Dynamics, Chemical Reviews 118 (2018) 3305–3336.

[20] R. Car, M. Parrinello, Unified Approach for Molecular Dynamics and Density-Functional Theory, Physical Review Letters 55 (1985) 2471–2474.

[21] A. Sharma, K. M. Mukut, S. P. Roy, E. Goudeli, The coalescence of incipient soot clusters, Carbon 180 (2021) 215–225.

[22] A. M. Kamat, A. C. T. van Duin, A. Yakovlev, Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field, Journal of Physical Chemistry A 114 (2010) 12561–12572.

[23] S. Shabnam, ReaxFF reactive force field investigations in combustion and nascent soot formation, 2020. [Online; accessed 21. Aug. 2021].

[24] Q. Mao, A. C. T. van Duin, K. H. Luo, Formation of incipient soot particles from polycyclic aromatic hydrocarbons: A ReaxFF molecular dynamics study, Carbon 121 (2017) 380–388.

[25] C. A. Schuetz, M. Frenklach, Nucleation of soot: Molecular dynamics simulations of pyrene dimerization, Proceedings of the Combustion Institute 29 (2002) 2307–2314.

[26] S. Han, X. Li, F. Nie, M. Zheng, X. Liu, L. Guo, Revealing the Initial Chemistry of Soot Nanoparticle Formation by ReaxFF Molecular Dynamics Simulations, Energy & Fuels 31 (2017) 8434–8444.

[27] C. Chen, X. Jiang, Molecular dynamics simulation of soot formation during diesel combustion with oxygenated fuel addition, Physical Chemistry Chemical Physics 22 (2020) 20829–20836.

[28] H. A. Michelsen, M. B. Colket, P.-E. Bengtsson, A. D'Anna, P. Desgroux, B. S. Haynes, J. H. Miller, G. J. Nathan, H. Pitsch, H. Wang, A Review of Terminology Used to Describe Soot Formation and Evolution under Combustion and Pyrolytic Conditions, ACS Nano 14 (2020) 12470–12490.

[29] H. Wang, Formation of nascent soot and other condensed-phase materials in flames, Proceedings of the Combustion Institute 33 (2011) 41–67.

[30] K. O. Johansson, M. P. Head-Gordon, P. E. Schrader, K. R. Wilson, H. A. Michelsen, Resonance-stabilized hydrocarbon-radical chain reactions may explain soot inception and growth, Science 361 (2018) 997–1000.

[31] M. Frenklach, A. M. Mebel, On the mechanism of soot nucleation, Physical Chemistry Chemical Physics 22 (2020) 5314–5331.

[32] J. Hwang, F. S. Hirner, C. Bae, C. Patel, T. Gupta, A. K. Agarwal, HRTEM evaluation of primary soot particles originated in a small-bore biofuel compression-ignition engine, Applied Thermal Engineering 159 (2019) 113899.

[33] K. O. Johansson, F. El Gabaly, P. E. Schrader, M. F. Campbell, H. A. Michelsen, Evolution of maturity levels of the particle surface and bulk

during soot growth and oxidation in a flame, Aerosol Science and Technology 51 (2017) 1333–1344.

[34] XYZ trajectory — MDAnalysis User Guide documentation, 2021. [Online; accessed 5. Jun. 2021].

[35] Y. Kim, W. Y. Kim, Universal Structure Conversion Method for Organic Molecules: From Atomic Connectivity to Three-Dimensional Geometry, Bulletin of the Korean Chemical Society 36 (2015) 1769–1777.

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

[37] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland, J. Laufer, Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited, Journal of Chemical Information and Computer Sciences 32 (1992) 244–255.

[38] S. Le Roux, P. Jund, Ring statistics analysis of topological networks: New approach and application to amorphous ges2 and sio2 systems, Computational Materials Science 49 (2010) 70–83.

[39] M. Brehm, B. Kirchner, TRAVIS - a free analyzer and visualizer for Monte Carlo and molecular dynamics trajectories, Journal of Chemical Information and Modeling 51 (2011) 2007–2023.

[40] M. Brehm, M. Thomas, S. Gehrke, B. Kirchner, TRAVIS – a free analyzer for trajectories from molecular simulation, The Journal of Chemical Physics 152 (2020) 164105.

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Third Edition, The MIT Press, Cambridge, MA, USA, 2009.

[42] H. Flávio, 1.2) Creating a distances matrix, 2021. [Online; accessed 6. Jun. 2021].

[43] NetworkX — NetworkX documentation, 2021. [Online; accessed 6. Jun. 2021].

[44] R. Tarjan, Depth-First Search and Linear Graph Algorithms, undefined (1972).

[45] jensengroup, xyz2mol, 2021. [Online; accessed 23. Jun. 2021].

[46] G. Landrum, RDKit, 2019. [Online; accessed 23. Jun. 2021].

[47] R. L. Vander Wal, A. Yezerets, N. W. Currier, D. H. Kim, C. M. Wang, HRTEM Study of diesel soot collected from diesel particulate filters, Carbon 45 (2007) 70–77.

[48] C. K. Gaddam, C.-H. Huang, R. L. Vander Wal, Quantification of nanoscale carbon structure by HRTEM and lattice fringe analysis, Pattern Recognition Letters 76 (2016) 90–97.

[49] R. L. V. Wal, Soot Nanostructure: Definition, Quantification and Implications, SAE Transactions 114 (2005) 429–436.

[50] Y.-C. Chen, A tutorial on kernel density estimation and recent advances, 2017.

[51] D. W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, 2nd Edition, Wiley, Hoboken, NJ, USA, 2015.

[52] scipy.stats.gaussian_kde — SciPy v1.7.1 Manual, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html`, 2021. [Online; accessed 21. Jan. 2022].

[53] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272.

[54] Anaconda software distribution, 2020.

[55] F. Lundh, An introduction to tkinter, URL: www. pythonware. com/library/tkinter/introduction/index. htm (1999).

[56] Installation guide for conda, 2021. [Online; accessed 5. Jul. 2021].

[57] Miniconda documentation, 2021. [Online; accessed 5. Jul. 2021].

[58] K. M. Mukut, S. Roy, kmmukut/RingDetection: RingDetection, `https://doi.org/10.5281/zenodo.4283067`, 2020. v1.01.

[59] Open Babel development team, Open babel, 2021. [Online; accessed 5. Jul. 2021].

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: