AI

MDPI

Article

HiFrAMes: A Framework for Hierarchical Fragmentation and Abstraction of Molecular Graphs

Yuncheng Yu, Max A. Smith, Haidong Wang and Jyh-Charn Liu *

Department of Computer Science & Engineering, Texas A&M University, College Station, TX 77843, USA; yyc@tamu.edu (Y.Y.); maxsmith271346@tamu.edu (M.A.S.)
* Correspondence: liu@cse.tamu.edu

Abstract

Recent advances in computational chemistry, machine learning, and large-scale virtual screening have rapidly expanded the accessible chemical space, increasing the need for interpretable molecular representations that capture the hierarchical topological structure of molecules. Existing formats, such as Simplified Molecular Input Line Entry System (SMILES) strings and MOL files, effectively encode molecular graphs but provide limited support for representing the multi-level structural information needed for complex downstream tasks. To address these challenges, we introduce HiFrAMes, a novel graph–theoretic hierarchical molecular fragmentation framework that decomposes molecular graphs into chemically meaningful substructures and organizes them into hierarchical scaffold representations. HiFrAMes is implemented as a four-stage pipeline consisting of leaf and ring chain extraction, ring mesh reduction, ring enumeration, and linker detection, which iteratively transforms raw molecular graphs into interpretable abstract objects. The framework decomposes molecules into chains, rings, linkers, and scaffolds while retaining global topological relationships. We apply HiFrAMes to both complex and drug-like molecules to generate molecular fragments and scaffold representations that capture structural motifs at multiple levels of abstraction. The resulting fragments are evaluated using selection criteria established in the fragment-based drug discovery literature and qualitative case studies to demonstrate their suitability for downstream computational tasks.

Keywords: cheminformatics; AI drug design; drug discovery; data curation; virtual screening; fragment library

Check for updates

Academic Editor: Pier Luigi Gentili

Received: 29 December 2025
Revised: 5 February 2026
Accepted: 10 February 2026
Published: 13 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The recent release of Google DeepMind's AlphaFold3 has accelerated interest in AI-driven drug discovery by expanding protein structure availability at scale [1]. However, progress in this area fundamentally depends on how molecular structures and physicochemical properties are represented for computation. Complex molecules must be transformed into data formats, such as Simplified Molecular Input Line Entry System (SMILES) strings or fingerprints, that machine learning models can use for prediction or design [2]. Just as sentences are made up of words and words from letters, complex chemical structures are composed of basic chemical components. Therefore, the central challenge is not merely fragmenting molecules but designing fragmentation methods that expose informative substructure information while preserving chemical interpretability.

Molecules, especially those that are large and complex, consist of many structural components, each contributing to the molecule's overall function and behavior. Take synthetic

cannabinoids as an example; they are often fragmented into four primary groups during mass spectrometry analysis, the head, linker, core, and tail, reflecting their distinct chemical roles in biological interactions and analytical behavior [3]. Although such decomposition is typically applied in experimental settings, the underlying modular organization is a general characteristic of complex molecules across both natural and synthetic compounds. This compositional nature motivates the development of systematic methods for decomposing sophisticated chemical structures into elementary components. Through the decomposition of complex molecules into chemically meaningful substructures and the identification of relationships among these substructures, researchers can build a robust foundation for future advancements in AI-driven drug discovery and chemical analysis.

Among the widely adopted approaches, molecular fragmentation has traditionally relied on predefined fragment libraries and experimental screening techniques, such as high-resolution mass spectrometry (HRMS), X-ray crystallography, and nuclear magnetic resonance (NMR), to elucidate molecular structure and generate fragments with low molecular weight (MW) [4,5]. While these techniques are effective and chemically grounded, they are limited by high experimental cost, time-consuming analysis, and the inability of predefined fragment libraries to encompass all potential molecular fragments. Therefore, such approaches are often used as preprocessing tools rather than as scalable, standalone solutions, particularly when analyzing large and diverse molecular datasets.

To improve scalability, computational fragmentation methods have increasingly drawn inspiration from natural language processing (NLP), facilitated by the widespread adoption of SMILES strings. By treating SMILES strings as sentences, subword-based segmentation techniques aim to decompose molecules into statistically informative substructures. Byte-pair encoding (BPE), introduced by Sennrich [6], exemplifies this approach by iteratively merging frequently occurring token pairs to construct data-driven vocabularies of molecular fragments. Despite their scalability, these methods rely on statistical frequency and operate on linear representations, which often fail to preserve chemically meaningful motifs and obscure underlying molecular topology. Graph-based fragmentation methods address these limitations by directly leveraging molecular graphs to preserve chemically meaningful substructures.

A fundamental concept in drug discovery is the molecular scaffold, the invariant core of a molecule that is retained while other periphery structures are varied to alter properties [7]. Bemis and Murcko formalized this concept by defining molecular frameworks obtained through the removal of terminal side chains [8,9]. Beyond scaffold extraction, rule-based methods such as the Retrosynthetic Combinatorial Analysis Procedure (RECAP) [10] and the Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) [11,12] introduced predefined cleavage rules based on retrosynthetically relevant bonds to generate chemically valid building blocks. While interpretable and chemically intuitive, these approaches remain constrained by fixed cleavage rules and limited flexibility.

Recent work has sought to address constraints through graph-based fragmentation methods that preserve structural connectivity without relying on predefined rules. Matched molecular pairs (MMPs) model local structural transformations through dual cleavages and synthetic accessibility filtering [13,14], while methods such as ReLMole [15] and Tree Decomposition [16] further abstract molecular graphs into functional-group-level or hierarchical representations. Although these approaches demonstrate meaningful progress, many lack a principled mechanism for constructing interpretable, hierarchical abstractions that systematically organize molecular topology across different levels of granularity.

Although there have been extensive advances in molecular fragmentation, existing methods continue to face fundamental trade-offs between chemical interpretability, topological fidelity, and representational flexibility. Motivated by these limitations, we introduce

HiFrAMes, a framework for **Hi**erarchical **Fr**agmentation and **A**bstraction of **M**ol**e**cular graph**s**. HiFrAMes transforms raw chemical structure graphs into interpretable, abstracted topological scaffold representations to enable structured and chemically informed analysis of complex molecules. In this work, we (1) present a four-stage, topology-driven pipeline for hierarchical fragmentation of molecular graphs; (2) define a set of abstract substructures (chains, rings, linkers, and scaffold representations) that are derived from input molecular graphs; and (3) evaluate the resulting fragment library extracted from ZINC-250K (a widely used benchmark set of 250,000 drug-like molecules) using multiple fragment selection criteria established in the literature, as well as qualitative case studies to demonstrate that HiFrAMes produces quality, reusable fragments aligned with common medicinal chemistry motifs.

2. Materials and Methods

2.1. Framework Overview

HiFrAMes is designed to transform raw chemical structure graphs into interpretable, abstract topological scaffolds through multiple stages of reduction. Given a molecular graph as input, the system extracts fundamental topological objects and organizes them into a hierarchical representation for downstream cheminformatics analysis. The overall workflow, as shown in Figure 1, is divided into two levels: topological object detection and scaffold construction.

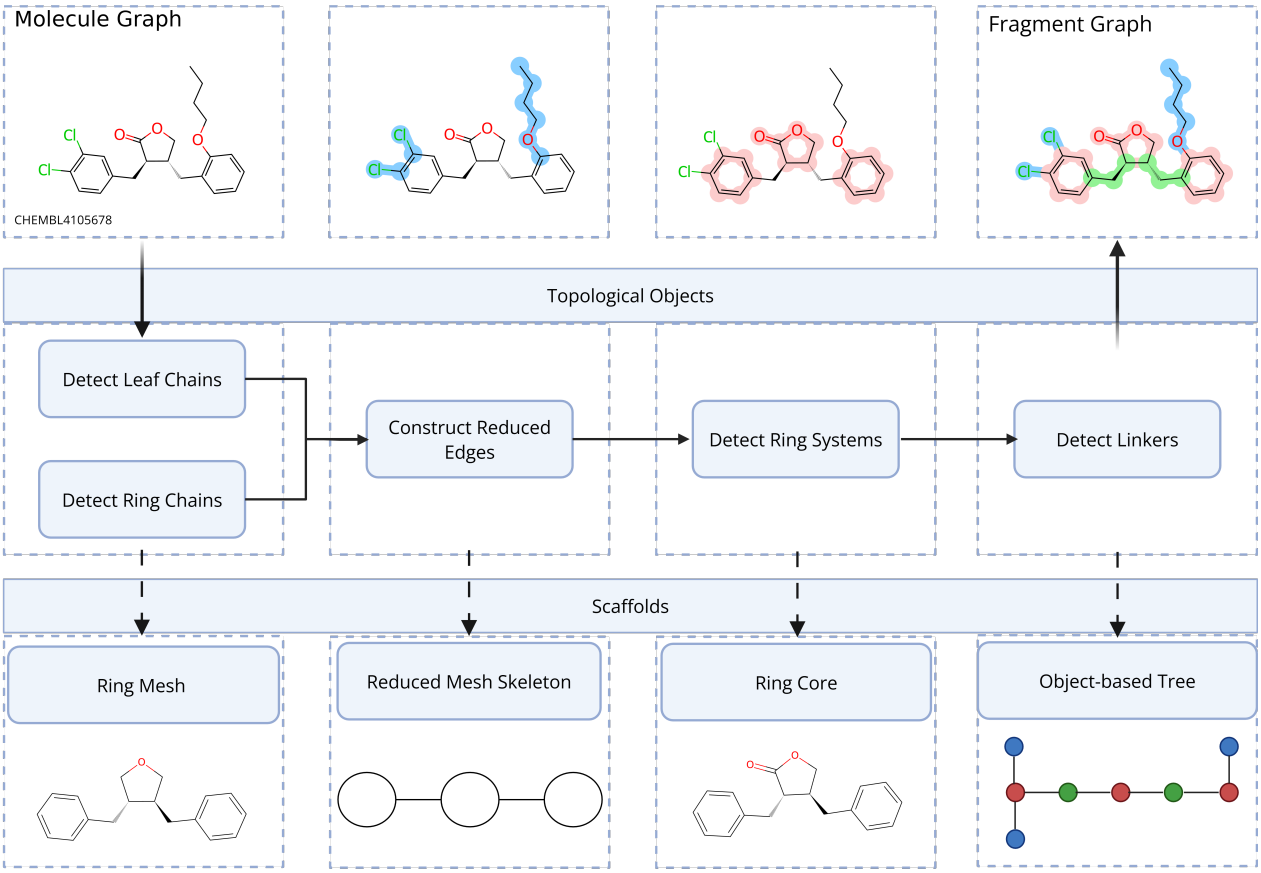

Figure 1. Overviewof the HiFrAMes framework. An input molecular graph (CHEMBL4105678) [17] is decomposed into fundamental topological objects and assembled into hierarchical scaffold representations through a two-level pipeline comprising topological object detection and scaffold construction [18].

At the object detection level, the framework decomposes the input molecular graph into atom-level topological substructures. A sequence of core algorithms operates to extract a set of fundamental structural objects, including chains, linkers, and rings, which together capture the core connectivity patterns of input molecules. These objects represent chemically meaningful substructures that serve as the building blocks for higher-level abstraction.

The extracted objects are transformed into four abstract scaffold representations constructed at different stages of the pipeline: ring meshes, reduced mesh skeletons, ring cores, and object-based trees. A ring mesh is the molecular graph obtained by removing all leaf chains, leaving all interconnected rings, including adjacent linkers. The ring core is a richer form of the ring mesh that emphasizes rings and their chemically integral exocyclic attachments. The reduced mesh skeleton simplifies this structure by collapsing intermediate nodes into reduced edges, yielding a simplified graph that retains connectivity between major structural regions. Lastly, the object-based tree organizes all of the extracted topological objects into a hierarchical tree structure based on their connectivity to support structured traversal and compact representation of molecular topology.

HiFrAMes operates on undirected molecular graphs and focuses on preserving topological connectivity rather than stereochemical or conformational detail. Atomic element types, bond orders, ring membership, and formal charges are retained within extracted fragments, and aromaticity is preserved since rings are not fragmented; however, HiFrAMes does not perform stereochemical reassignment or inference during fragmentation. As a result, stereochemistry is preserved only when chiral centers are fully contained within an extracted subgraph and may be lost when stereocenters span fragmentation boundaries, consistent with the behavior of established fragmentation methods such as RECAP and BRICS. This design choice prioritizes canonical, topology-driven abstraction and fragment reuse at scale, while stereochemistry can be reintroduced at later stages for applications that require stereospecific modeling.

2.2. Topological Decomposition Algorithms

2.2.1. Algorithm A1: Leaf Chain and Ring Chain Extraction

Our iterative leaf chain detection algorithm identifies and labels all leaf chains in a chemical structure graph, as depicted in Figure 2. A leaf chain is a maximal, connected, acyclic induced subgraph of the molecular graph that attaches to the remaining structure at exactly one vertex, whereas a leaf node is defined as any vertex of degree one in the molecular graph. The input to the algorithm is a molecular graph represented as an adjacency list. The algorithm iteratively removes leaf nodes and traces their connectivity to construct an induced subgraph C that contains all extracted leaf chains, while preserving the remaining cyclic components in a residual subgraph R.

At each iteration, a leaf node v is selected and removed together with its incident edge. The adjacency list of its neighboring node u is then updated by removing v. If the degree of u drops to one, it becomes a new leaf node and is processed in a subsequent iteration. All removed nodes and edges are assembled into C. This pruning process continues until no nodes of degree one remain in the graph. After all leaf chains have been removed, the remaining nodes should all have degree two or greater, leaving only the cyclic core.

Our pipeline enforces chemical constraints by separately handling exocyclic bonds that are topologically similar to those in leaf chains, but are chemically integral to ring systems. An exocyclic bond is a covalent bond connecting a ring atom to a non-ring atom, such as oxygen, sulfur, carbon, or nitrogen. Such bonds are often chemically and electronically integral to the ring system, and removing them would fundamentally change the ring environment. Therefore, our algorithm does not handle exocyclic double bonds as leaf chains. Instead, exocyclic attachments that are chemically integral to adjacent rings are

treated as ring-associated chains, referred to as ring chains, and are preserved within the residual subgraph R rather than being included in the leaf chain subgraph C. The overall time and space complexity of the algorithm is linear, $O(n + m)$, where n is the number of vertices and m is the number of edges. Detailed pseudocode for Algorithm A1 is provided in Appendix A.

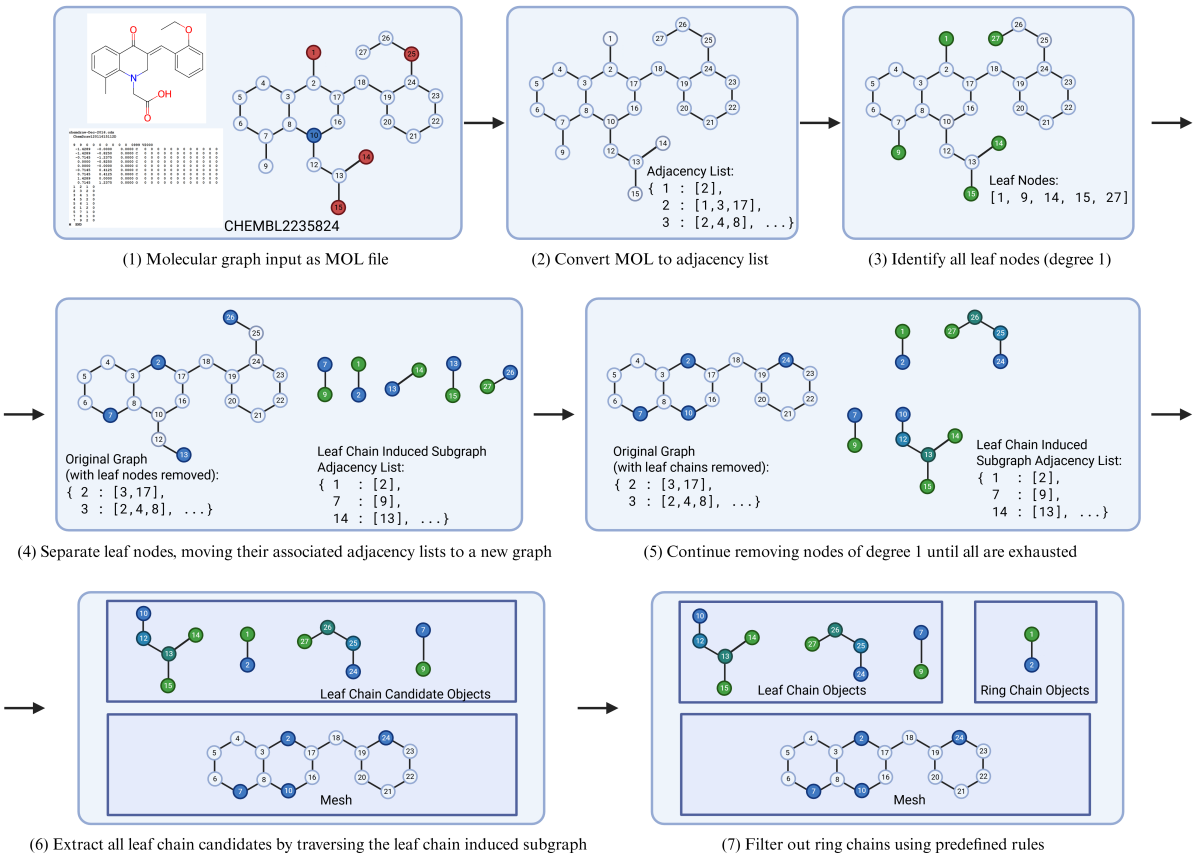

Figure 2. Algorithm A1: Leaf chain and ring chain extraction. An input molecular graph (CHEMBL2235824) [17] is converted to an adjacency list, leaf nodes are iteratively removed to form induced leaf chains, and the remaining vertices constitute a residual cyclic subgraph (ring mesh). Leaf chain objects are extracted while preserving exocyclic attachments as ring-associated chains [18].

2.2.2. Algorithm A2: Ring Mesh Reduction

In this algorithm, we construct a reduced mesh skeleton from the ring mesh result from Algorithm A1 by forming reduced edges. A reduced edge is an abstraction of an acyclic, connected induced path that links two furcation nodes (vertices with degree greater than or equal to three from the ring mesh) through intermediate nodes of degree two. The two furcation nodes may coincide, allowing for self-loops. Furcation nodes from the ring mesh are preserved in the reduced graph. The input to the algorithm consists of the set of furcation nodes and the residual cyclic subgraph (ring mesh) obtained from Algorithm A1.

From each furcation node, all adjacent nodes are examined and reduced edges are constructed from the furcation node through each of its neighbors. Starting from a furcation node f, the algorithm traverses all neighboring nodes, following paths composed exclusively of vertices with degree two, until another furcation node is encountered. The resulting path, together with its two endpoint furcation nodes, forms a reduced edge and is added to the set of reduced edges in the reduced graph.

By processing each furcation node independently and limiting traversal to nodes with degree two, the algorithm guarantees that all valid reduced edges are identified while preserving the underlying cyclic topology of the molecular graph. The reduced edges

provide a simplified representation of the cyclic connectivity that is subsequently used for ring enumeration. The overall time and space complexity of the algorithm is $O(n + m)$, where n is the number of nodes and m is the number of edges, since each node and edge is visited at most once during traversal. Figure 3 illustrates the implementation details and data flow of Algorithm A2. Detailed pseudocode for this algorithm is provided in Appendix A.

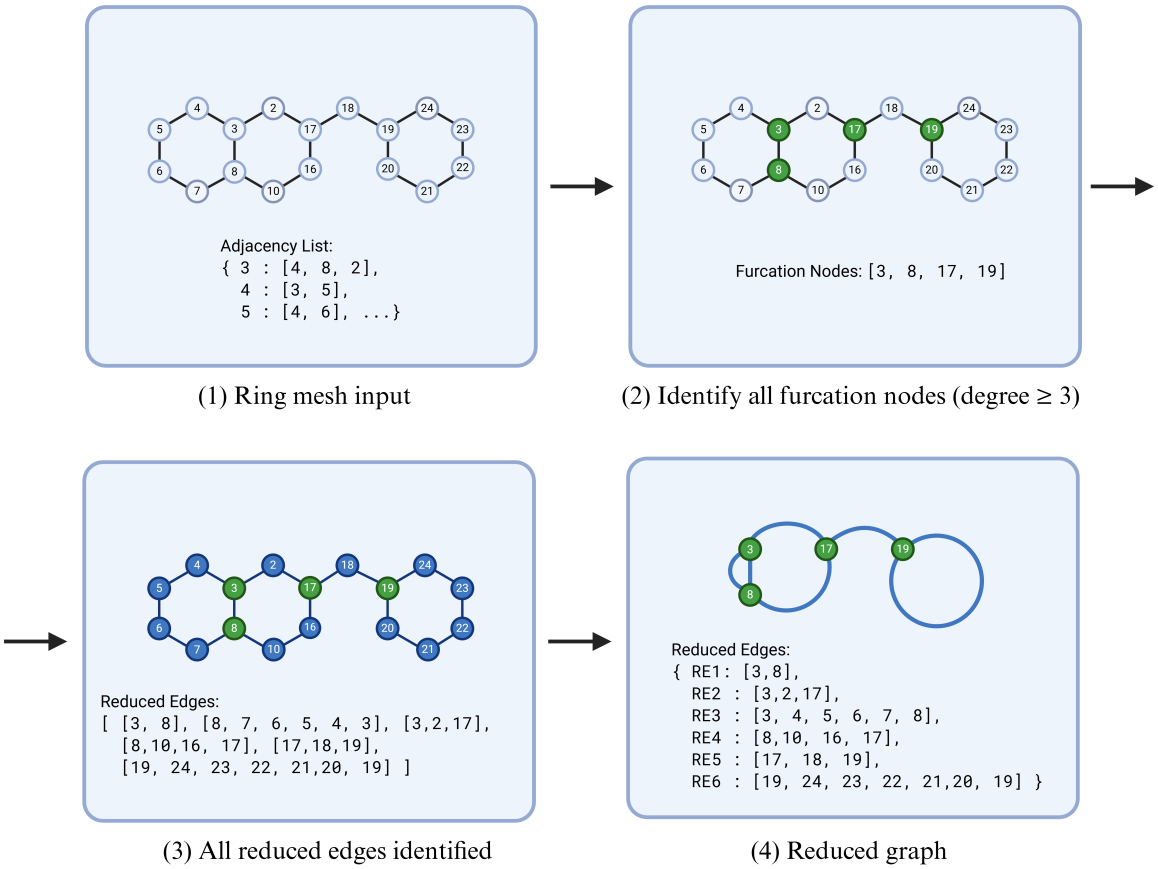

Figure 3. Algorithm A2: Ring mesh reduction. From the residual cyclic subgraph (ring mesh), furcation nodes are identified and used to yield reduced edges. The resulting reduced edges yield a simplified representation of cyclic connectivity for ring enumeration [18].

2.2.3. Algorithm A3: Ring Enumeration

From the reduced molecular graph, we can enumerate each ring. More precisely, we aim to enumerate all edge-unique cycles in graph G. A cycle is considered edge-unique if there does not exist another cycle with the same unordered set of edges. A simple cycle refers to a closed path in which no node, other than the endpoints, is repeated. We employ the `simple_cycles()` method from the NetworkX library [19] as the foundation of our algorithm, which leverages a modified variant of Johnson's cycle enumeration algorithm and supports undirected multigraphs containing self-loops and parallel edges [20].

The cycles enumerated by the `simple_cycles()` method ignore possible multiple edges between the same pair of nodes. Therefore, we designed a post-processing algorithm to find all edge-unique cycles in the given graph. Each simple cycle enumerated by the NetworkX is treated as a skeleton, upon which edge-unique cycles can be generated using different combinations of parallel edges between the same pair of nodes. Possible edge-unique cycles are permuted by enumerating all the combinations of parallel edges between node pairs in the simple cycle, simulating a digital mixed-radix counter. We treat each node pair as a step, such that an array of consecutive steps covers all the nodes and edges in a

simple cycle. Then we start with an initial combination of edges and select other edges at each step. Each time we make an update to edge selection at step `i`, edge selection at other steps are fixed. In this way, when we iterate over all the edges of selections at every step of a simple cycle, we have gone over all the possible edge combinations. Cycles with only two nodes are handled separately by enumerating all the possible combinations of different edges between this node pair. The complete process is illustrated in Figure 4, and the detailed pseudocode of Algorithm A3 is shown in Appendix A.

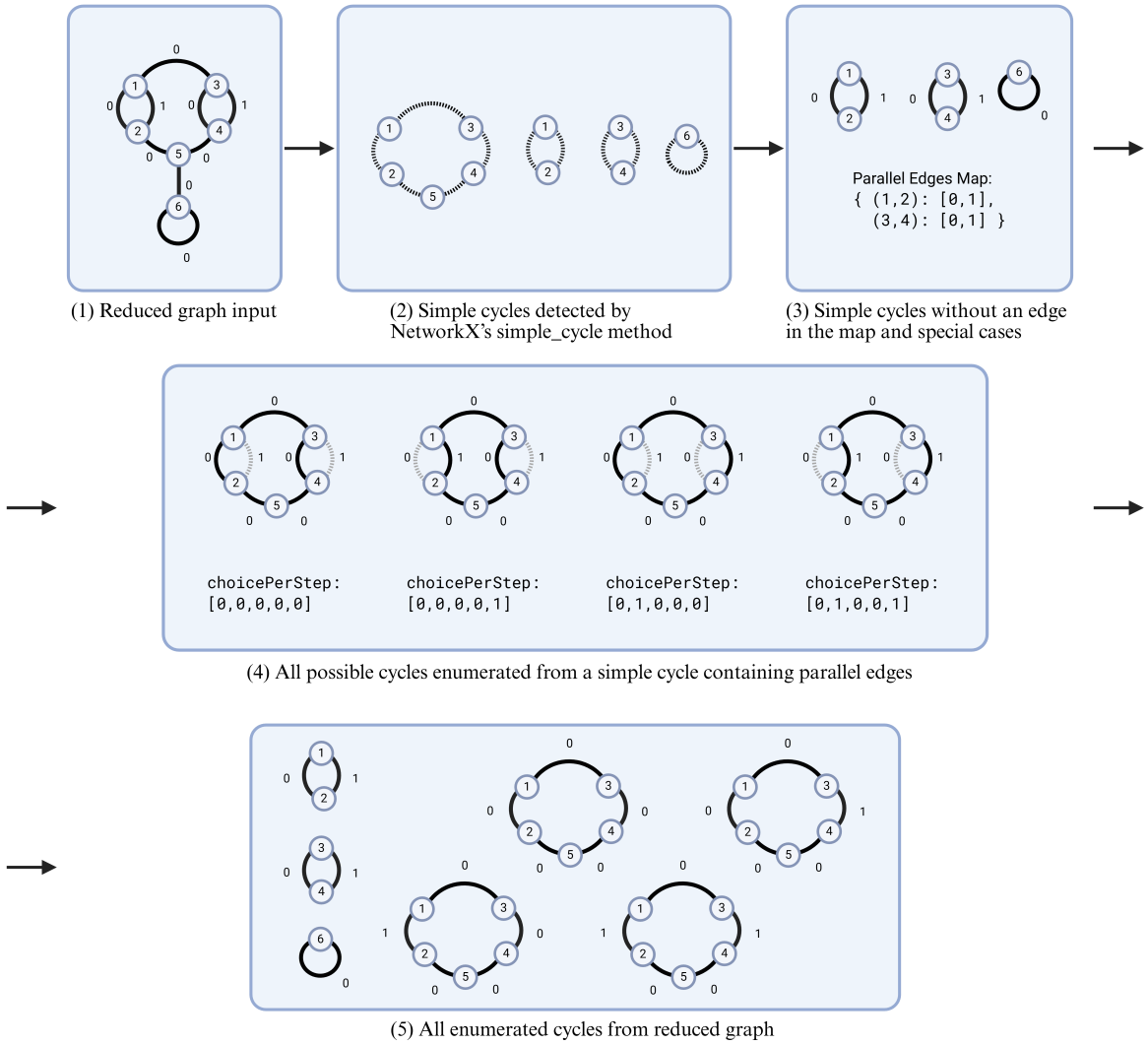

Figure 4. Algorithm A3: Ring enumeration. Node-simple cycles are first identified using NetworkX's implementation of Johnson's cycle enumeration algorithm. For cycles containing parallel edges, edge-unique cycles are generated by systematically enumerating all valid combinations of parallel edges for each node pair, yielding all distinct ring structures present in the multigraph [18].

The `simple_cycles()` method is based on Johnson's cycle enumeration algorithm, whose time complexity is $O((m+n)(C_{node}+1))$ [21], where m is the number of edges, n is the number of nodes, and C_{node} is the number of simple cycles. For each simple cycle c of length K_c, our post-processing enumerates all combinations of parallel edges between consecutive node pairs. Let r_k denote the number of parallel edges between the k-th node pair in the simple cycle, then the total number of edge-unique cycles is $R_c = \prod_{k=0}^{K_c-1} r_k$, it takes $O(K \cdot R)$ time to enumerate these combinations in a simple cycle using our algorithm, and $O(\sum_c K_c \cdot R_c)$ to enumerate over all simple cycles. Therefore, the total time complexity of the proposed algorithm is $O((m+n)(C_{node}+1) + \sum_c K_c \cdot R_c)$.

In the theoretical worst case, if K_{max} is the maximum cycle length and r_{max} is the maximum number of parallel edges per node pair, then for one cycle we have $R_c = \prod_{e \in c} r_e$, where $r_e \leq r_{max}^{K_{max}}$, and the total time complexity would be $O(C_{node} \cdot K_{max} \cdot r_{max}^{K_{max}})$. So in the theoretical worst case, the time complexity of this algorithm is exponential, which is unavoidable because the number of answers itself is exponential. After leaf pruning and chain collapsing in Algorithms A1 and A2, the input graph for Algorithm A3 is a small skeleton with constrained structure. Regarding the `simple_cycles()` method, n is relatively small and m is comparable to n. In Figure 5, we plot the distributions of six metrics of reduced graphs extracted from the full ZINC-250K dataset, including parallel edge counts between furcation node pairs, edge count in simple cycles, edge-unique ring count, simple cycle count, node count, and edge count. Based on Figure 5a–c, the number of reduced edges between node pairs is within 2 in most cases ($\leq$4 in all cases), the number of reduced edges found in all simple cycles is typically within 6 ($\leq$18 in all cases), and the number of edge-unique cycles, C_{edge}, is typically no more than 11 ($\leq$68 in all cases). Additionally, we know from Figure 5d–f, that the number of simple cycles in reduced graphs is typically within 20 ($\leq$160 in all cases), and there are typically no more than 8 nodes ($\leq$18 in all cases) and 12 edges ($\leq$21 in all cases) in the reduced graphs of small drug-like molecules. These statistics yield an estimated workload for Algorithm A3 on the order of $(8 + 12) \cdot (20 + 1) + 20 \cdot (6 \cdot 2^6) \approx 8.1 \times 10^3$ operations per molecule.

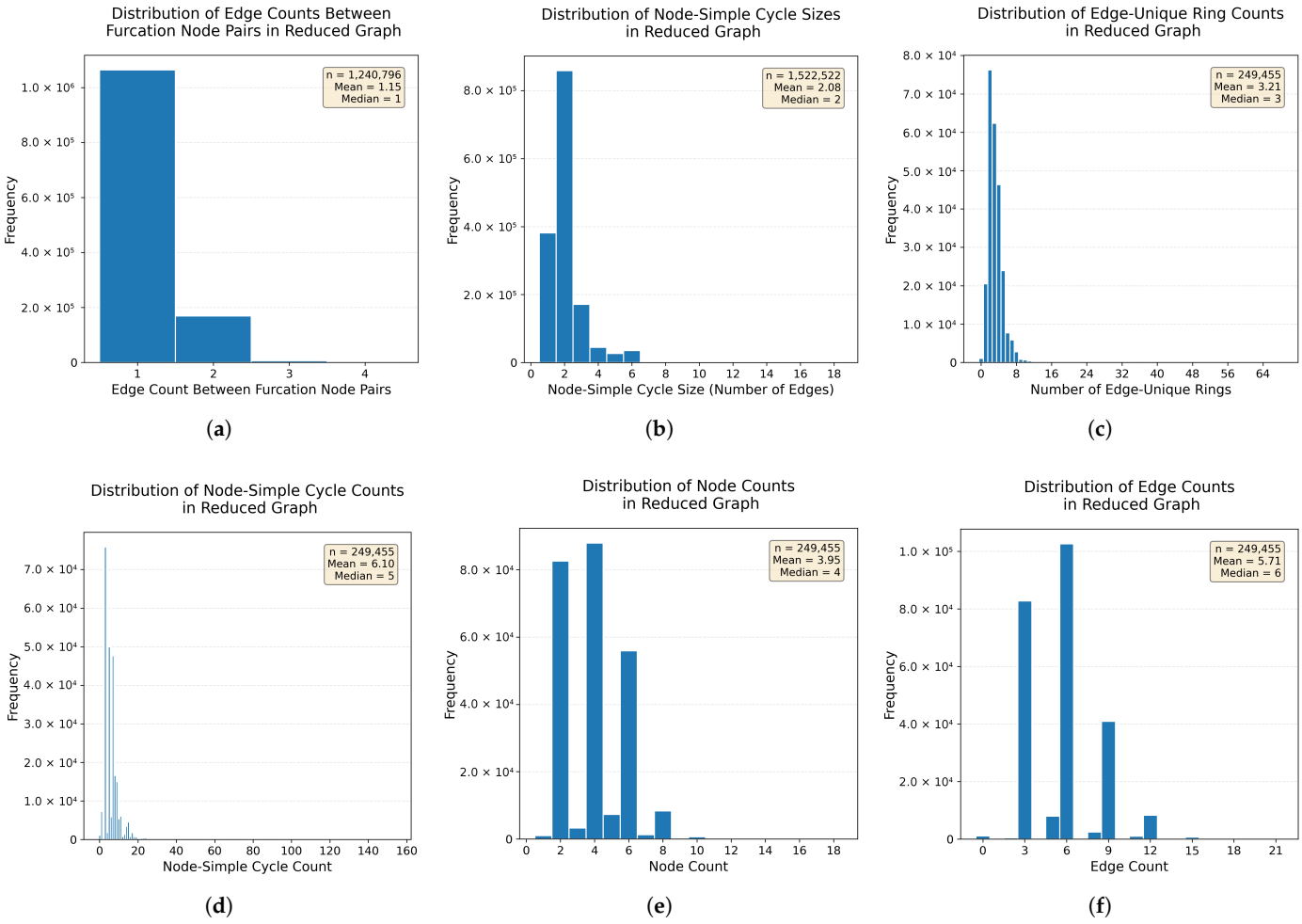

Figure 5. Statistics of reduced graphs of the ZINC-250K dataset. (**a**) Parallel edge counts between furcation node pairs, (**b**) Edge count in simple cycles, (**c**) Edge-unique ring count, (**d**) Simple cycle count, (**e**) Node count, and (**f**) Edge count.

2.2.4. Algorithm A4: Linker Extraction

In a ring mesh, different rings may or may not share nodes or edges. When two rings do not share any vertices or edges, they must be connected through an acyclic path containing one or more edges. Such paths are referred to as linkers, and the edges along these paths are termed linker edges. Linkers do not themselves form rings but serve as connectors between distinct ring systems.

Since all ring-associated edges have been exhaustively enumerated in Algorithm A3, the remaining edges in the graph correspond to linker edges. Algorithm A4 extracts these linker edges by comparing the full edge set of the ring mesh with the union of all edges participating in edge-unique cycles. This process iterates over all edge-unique cycles and all edges in the graph. The overall time complexity of the algorithm is $O(E + R)$, where E is the number of edges in the graph, and R is the number of edge-unique cycles in the graph. Detailed implementation of the algorithm is provided in Figure 6 and Appendix A.

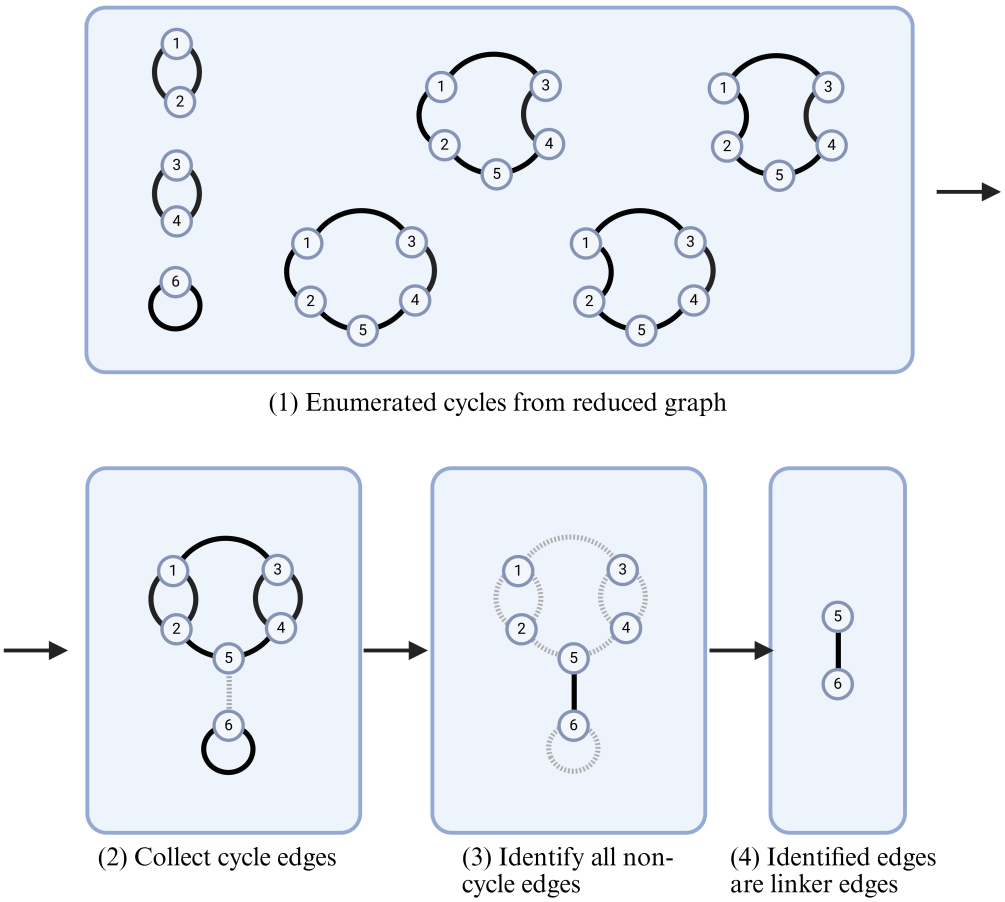

Figure 6. Algorithm A4: Linker extraction. Edges not participating in any edge-unique ring cycle are identified as linker edges, forming acyclic paths that connect distinct ring systems without contributing to ring closure [18].

3. Results

To demonstrate the functionality of our topological fragmentation framework, we apply HiFrAMes to several structurally diverse molecules as illustrative examples. Each molecule is decomposed into chemically meaningful topological objects, and from these objects, we generate scaffold representations that capture each molecule's global organization.

3.1. Vitamin B-12

Vitamin B-12 (cobalamin) is a large and structurally complex biomolecule that presents a challenge for automated fragmentation due to its polycyclic corrin ring system, multiple pendant side chains, and central cobalt coordination. Figure 7 shows the original molecular structure of vitamin B-12 together with the fragments recognized by the proposed framework.

The leaf chain extraction stage isolates a variety of small pedant side chains, including glycinamide and aminoamide type fragments, phosphate groups, primary alcohol chains, nitrile and carbonyl units, cycloalkyl segments, and hydroxyl-bearing chains. These fragments, highlighted in Figure 7b, correspond to terminal substituents distributed around the corrin core.

Ring enumeration identifies the major cyclic components, including the corrin macrocyclic core, the benzimidazole moiety, and the five-carbon sugar unit, highlighted in Figure 7c. These ring structures form the cyclic backbone of the molecule and are preserved during the fragmentation process. Linker extraction further isolates connecting regions that bridge distinct ring systems or connect rings to other structural units, such as the aminoalkyl phosphoester linker, which links the corrin ring to the nucleotide component of the molecule, as shown in Figure 7d.

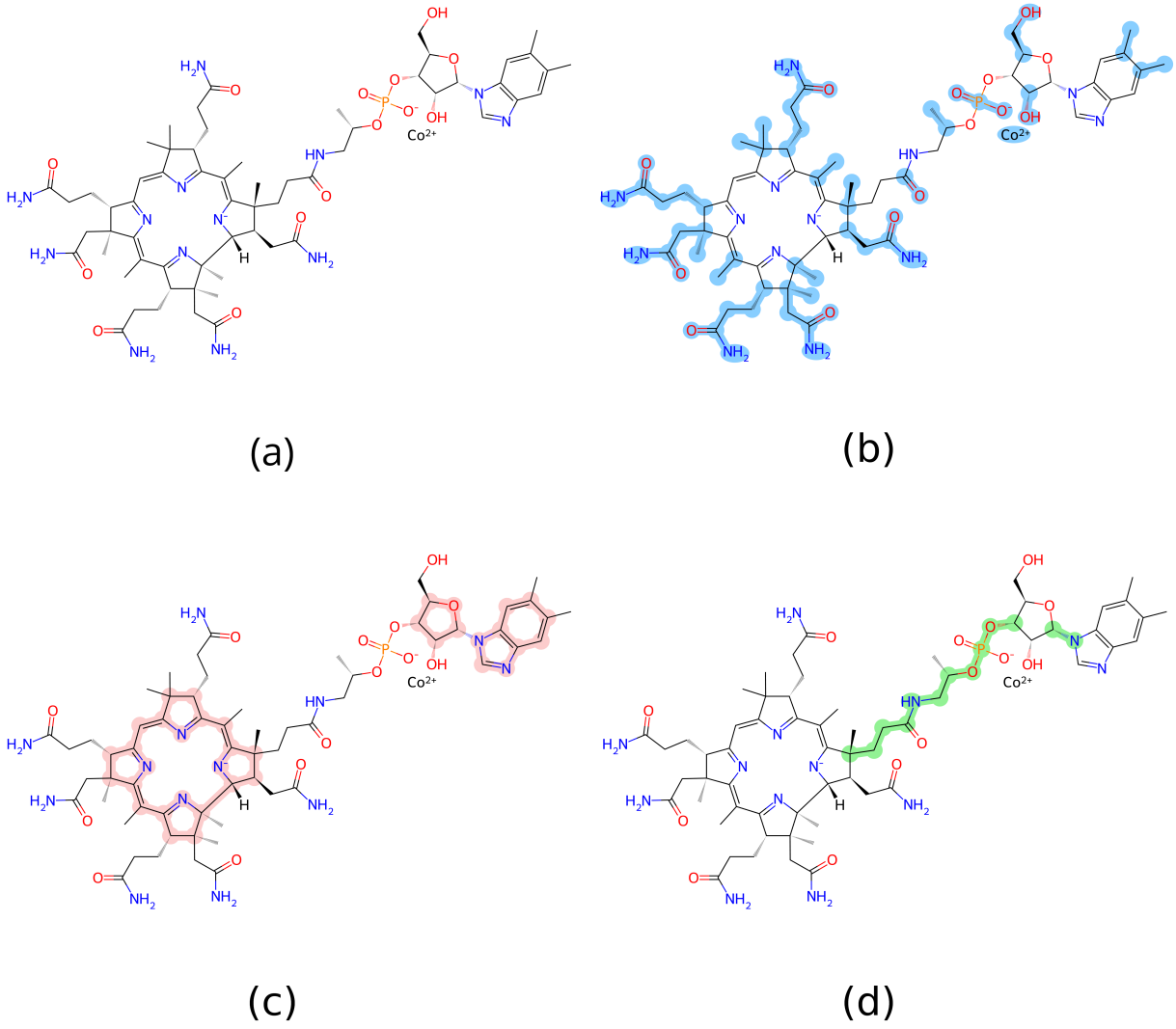

Figure 7. Fragments extracted from vitamin B-12. (**a**) Original molecular structure [22], (**b**) Extracted chains, (**c**) Extracted rings, (**d**) and Extracted linkers.

Beyond individual fragments, our framework produces a hierarchy of scaffold representations that capture molecular organization at multiple levels of abstraction. Figure 8a shows the original molecular structure, while Figure 8b presents the corresponding ring mesh scaffold obtained after removing all leaf chains. Figure 8c illustrates the reduced mesh skeleton, which encodes the connectivity between cyclic regions via reduced edges. Finally, Figure 8d shows the object-level topological hierarchical representation, in which leaf chains, rings, and linkers are represented as nodes in a tree-structured graph, with edges denoting their connectivity.

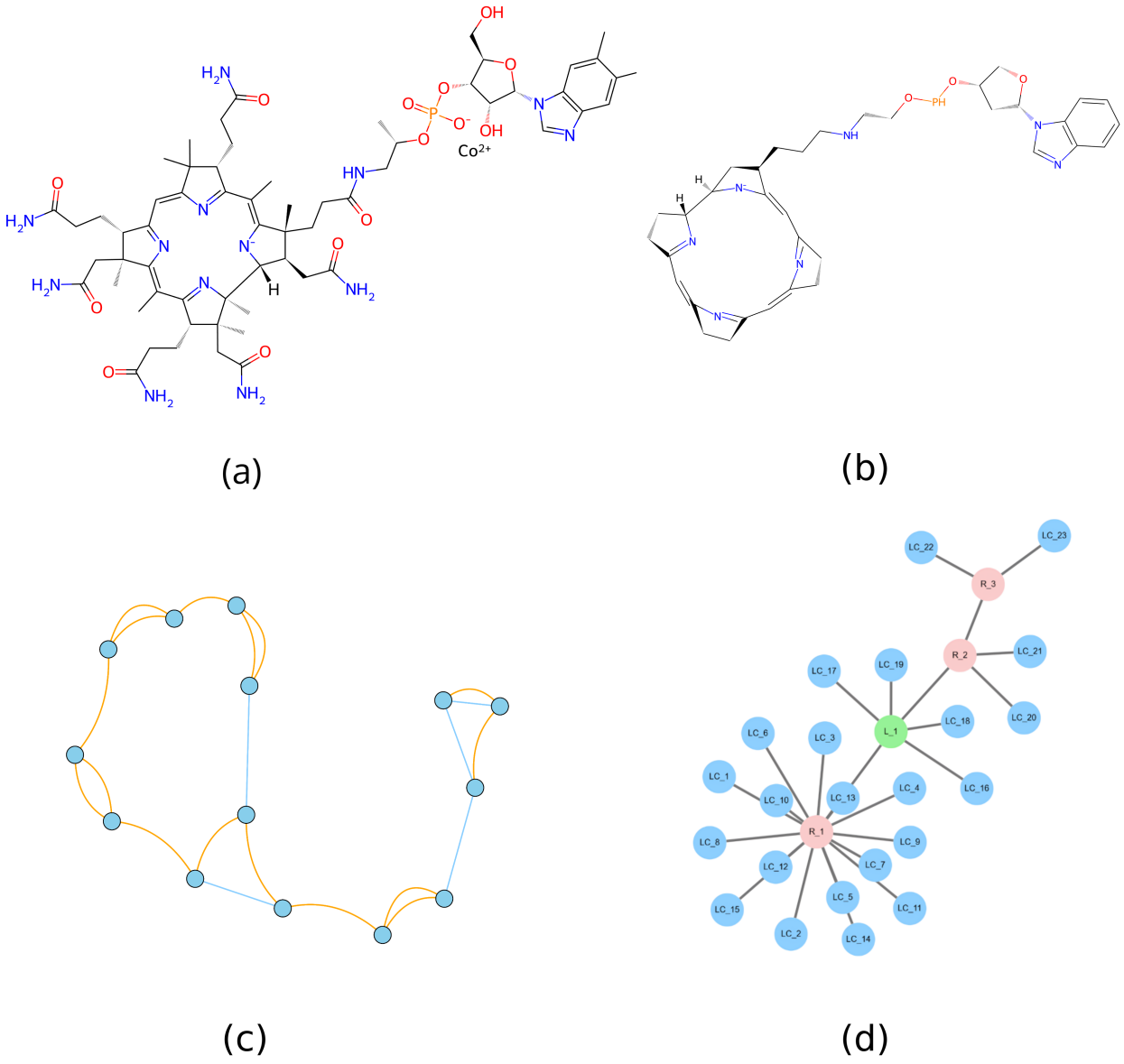

Figure 8. Scaffolds extracted from vitamin B-12. (**a**) Original molecular structure [22], (**b**) Ring mesh scaffold, (**c**) Reduced mesh skeleton (blue edges: single-edge segments, orange edges: multi-edge segments), and (**d**) Object-level topological hierarchical representation (blue: leaf chains, red: rings, and green: linkers) [18].

Together, Figures 7 and 8 demonstrate how the proposed framework systematically decomposes a highly complex biomolecule into interpretable topological components while preserving its overall connectivity structure.

3.2. Synthetic Cannabinoids

Synthetic cannabinoids are lab-designed compounds that mimic the psychoactive effects of plant-derived cannabinoids found in *Cannabis sativa*. When applied to the synthetic cannabinoid JWH-018, our pipeline produces a chemically interpretable decomposition that aligns closely with known pharmacophoric roles. The leaf chain detection stage isolates the pentyl side chain, shown in blue in Figure 9b, while subsequent ring enumeration identifies two distinct cyclic systems: the indole core and the naphthyl moiety, each highlighted red in Figure 9b. The remaining carbonyl group is then classified as a linker, shown in green in Figure 9b. According to the literature [23], the naphthyl unit serves as the receptor-binding head; the carbonyl linker modulates binding orientation, the indole ring functions as the selective cyclic core, and the pentyl chain affects potency. This case highlights how topological decomposition can support pharmacophore-based analysis and structure–property relationship studies.

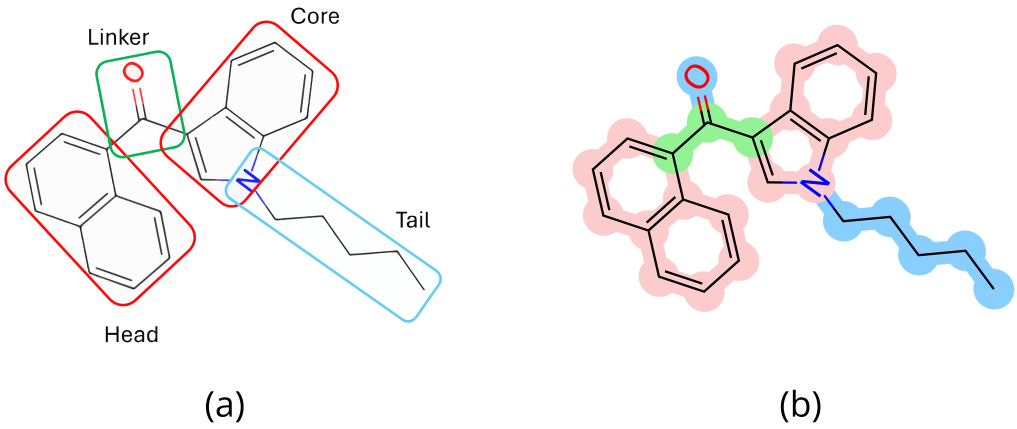

Figure 9. Decomposition illustration of synthetic cannabanoid (JWH-018). (**a**) Synthetic cannabanoid labeled with structural subunits defined in the literature [23] and (**b**) Chemical fragments recognized HiFrAMes.

3.3. Erythromycin

Erythromycin [24] is a clinically important macrolide antibiotic characterized by a large 14-membered lactone ring (macrocyclic scaffold) decorated with multiple sugars and functional substituents. Leaf chain extraction then highlights terminal groups such as hydroxyl-terminated chains, dimethylamino sugar branches, and short alkyl segments as shown in Figure 10a in blue. Through ring enumeration, our algorithm isolates the macrolactone core, highlighted red in Figure 10b. Linker detection identifies glycosidic bonds connecting the core to desosamine and cladinose units, both highlighted green in Figure 10c. The decomposition highlights how the rigid macrocyclic core provides a structural framework for binding to the bacterial ribosome, while flexible linkers and polar terminal groups enhance solubility and modulate interactions with the target site. The graph-based representation of erythromycin reveals a radial connectivity pattern, with the macrolactone scaffold as the central hub from which multiple functional chains and sugar moieties project outward.

These case studies demonstrate the ability of HiFrAMes to systematically generate standardized fragment libraries, in which unique substructures are extracted across molecules. Each fragment is represented as a graph and labeled according to its classification (chain, ring, linker). The generated libraries can enable rapid cross-molecule comparison, facilitating the identification of recurring pharmacophoric patterns and potential bioisosteres. We also translate the decomposed structures into topological graphs, where nodes correspond to fragments and edges denote covalent connectivity. This representation preserves higher-

level molecular connectivity while providing meaningful abstraction. Graph analysis reveals distinct connectivity motifs for each molecule. Vitamin B-12 exhibits a hub-and-spoke topology centered around the corrin ring. Synthetic cannabinoids display a clear head-linker-core-tail structure. Erythromycin demonstrates a radial multi-attachment pattern, in which a central macrolactone scaffold anchors multiple sugar moieties and peripheral substituents. These results confirm that our topological fragmentation framework is capable of accurately isolating chemically meaningful substructures across molecules of varying size and complexity to generate standardized fragment libraries and produce graph-based molecular abstractions.

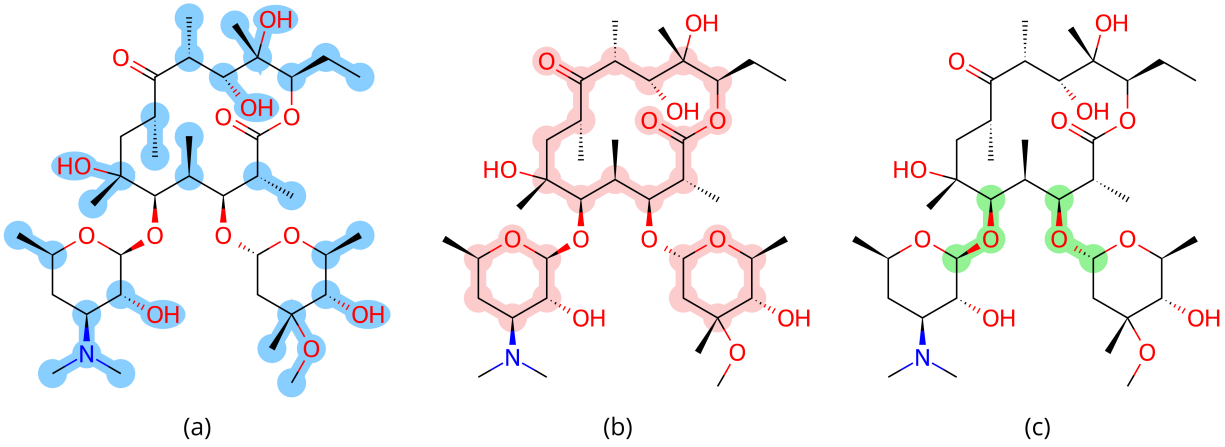

Figure 10. Decomposition illustration of erythromycin. (**a**) Identified terminal functional groups; (**b**) Identified macrolactone core, desosamine and cladinose; and (**c**) Identified ester linkers.

4. Discussion

To evaluate the quality of fragments extracted by HiFrAMes, we applied HiFrAMes to ZINC-250K, a 250,000 molecule subset of ZINC, one of the most widely used public databases of commercially available drug-like compounds [25]. On the ZINC-250K benchmark, our framework generated a rich vocabulary of chemically-valid fragments, obtaining 29,219 unique fragments from 1,638,977 fragment instances, enabling significant abstraction of the original molecular graphs. The resulting fragment library contains 19,803 leaf chain fragments, 7153 ring fragments, and 2468 linker fragments.

In fragment-based drug discovery (FBDD), fragments are commonly described as low complexity molecules with molecular weight below 300 Da and typically no more than 20 heavy atoms (non-hydrogen atoms) [26–28]. These size ranges are widely used to keep fragments interpretable, efficiently searchable, and to enable growing, linking, and merging [29–31]. As shown in Figure 11, the molecular weight and heavy atom count (HAC) distributions of HiFrAMes fragments exhibit tightly bounded and unimodal profiles across all fragment categories. Over 99% of fragments fall below the 300 Da threshold and contain no more than 20 heavy atoms, with median sizes well within these ranges. Moreover, leaf chain, ring, and linker fragments each span distinct, yet overlapping size ranges, consistent with their different roles within molecules.

Having established that HiFrAMes fragments conform to established fragment size and complexity criteria, we next examine the chemical composition and structural motifs represented by each fragment category. Common structural motifs extracted by HiFrAMes are shown in Figures 12–14, organized by fragment type. From the analysis presented in the figures, three key findings stand out.

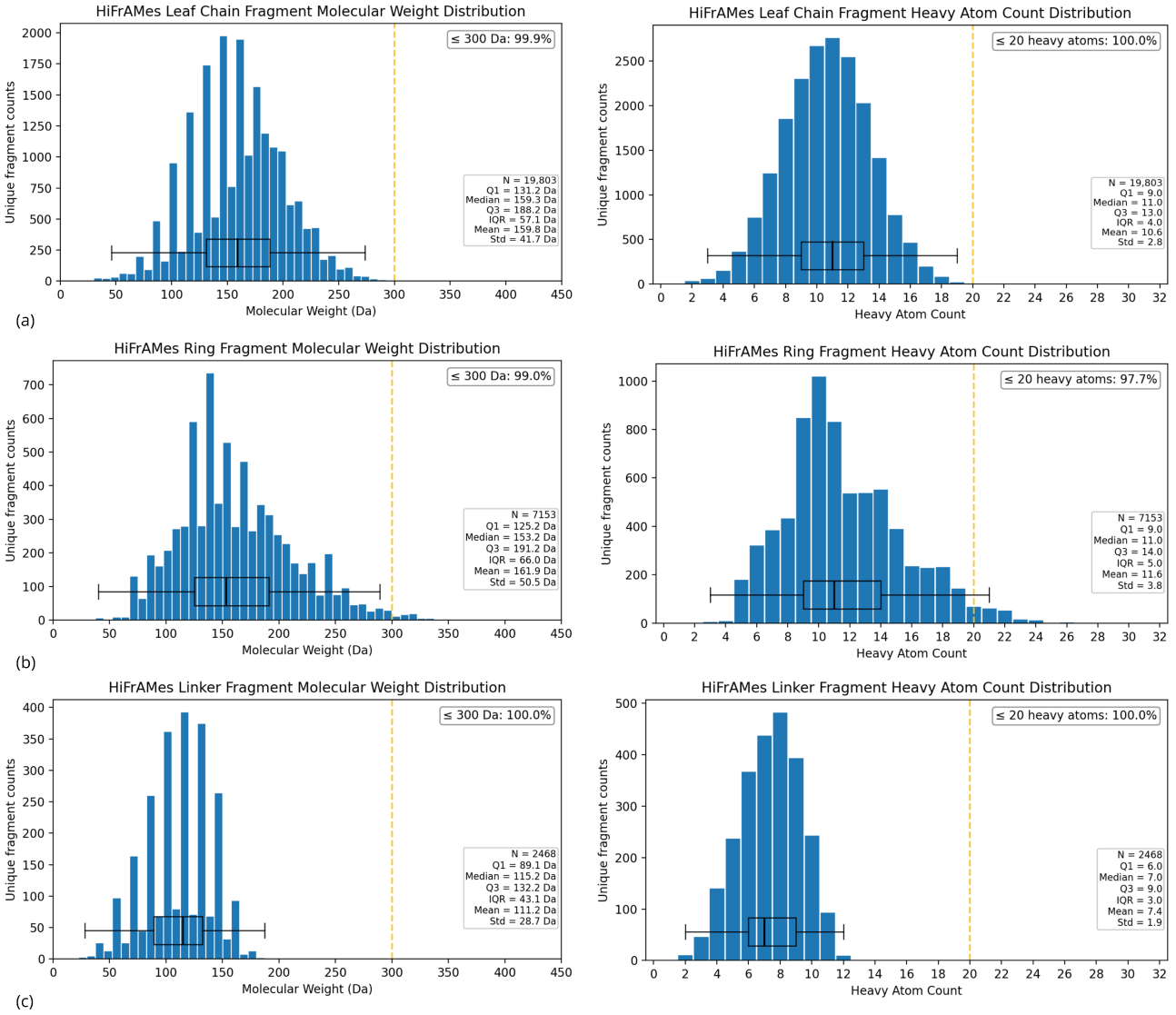

Figure 11. Molecular weight (**left**) and heavy atom count (**right**) distributions of HiFrAMes fragments extracted from ZINC-250K with box plots. (**a**) Leaf chain fragments, (**b**) Ring fragments, and (**c**) Linker fragments. Vertical dashed lines indicate common fragment-based drug discovery (FBDD) thresholds (300 Da and 20 heavy atoms).

First, leaf chains dominate by count, suggesting that most molecules feature multiple peripheral substituents decorating core scaffolds composed of a comparatively smaller number of ring and linker fragments. As shown in the leaf chain fragment profile in Figure 12, the most frequent leaf chains contain common functional groups, such as carbonyls (C=O), hydroxyls (-OH), amines (-NH_2), and halogenated substituents, consistent with what is typically observed in drug-like molecules. These fragments tend to be reactive or polar moieties appended to ring systems, a pattern well aligned with functional group driven optimization strategies in medicinal chemistry, where peripheral substituents are systematically varied to tune molecular properties while preserving core scaffolds [32]. Carbonyl and amino groups frequently act as H-bond donors or acceptors or reactive centers, whereas hydroxyls and protonated amines (-NH_3^+) increase polarity and solubility, thereby shaping Absorption, Distribution, Metabolism, and Excretion (ADME) profiles [33].

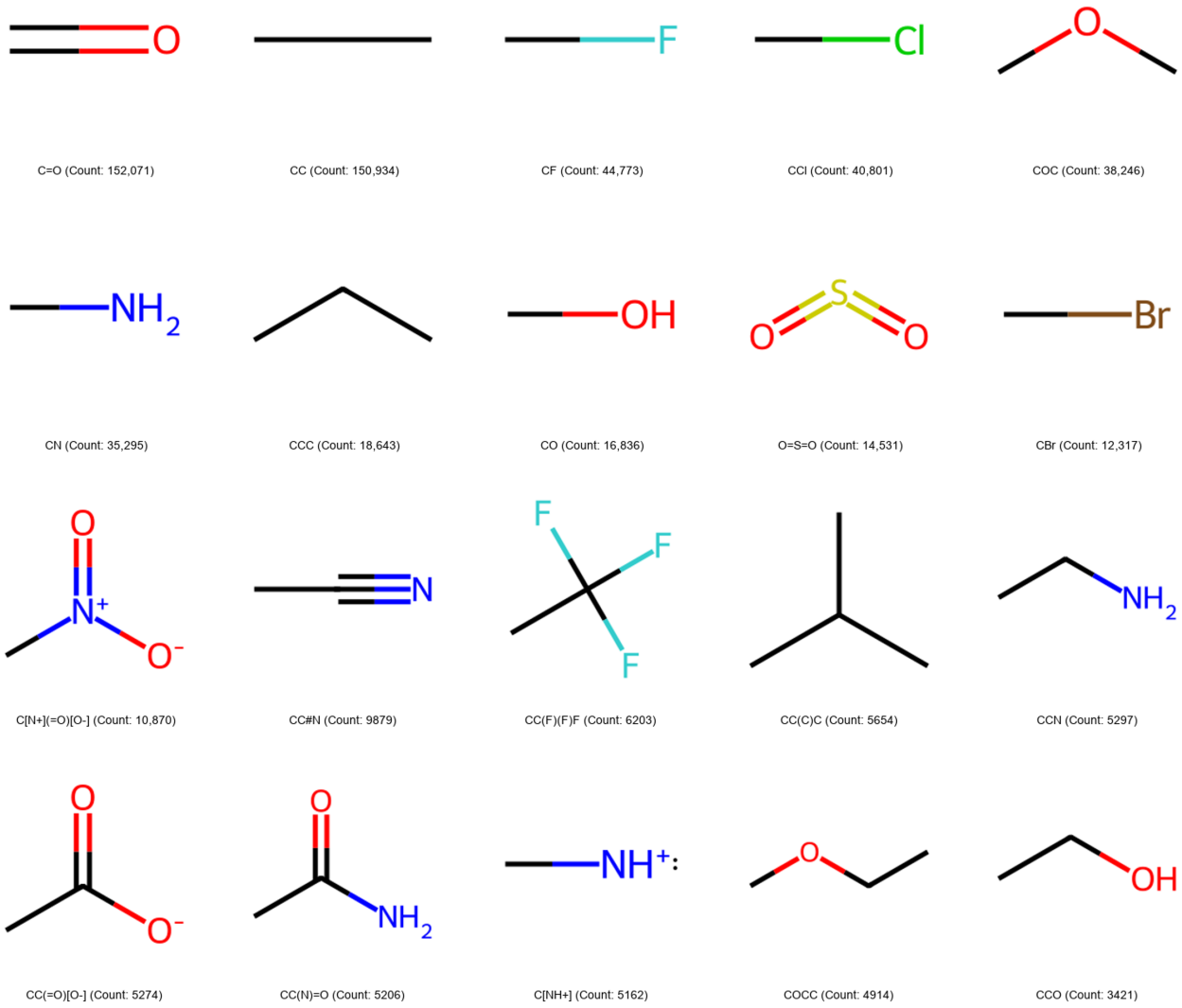

Figure 12. Top 20 most frequent leaf chain fragments from ZINC-250K.

Second, the number of unique ring systems is relatively small. Many molecules reuse the same ring families shown in Figure 13, such as aromatic benzene rings and heterocycles, indicating that HiFrAMes extracts frequently occurring core ring structures.

Third, simple aliphatic linkers shown in Figure 14, such as propyl (-CCC-), ethyl ether (-CCO-), and ethylamine (-CCN-), bridge larger substructures and are frequently observed as short connectors. These flexible linkers help space functional domains, modulate conformations, and often increase lipophilicity by boosting hydrocarbon content. Overall, the most frequent leaf chain, ring, and linker fragments correspond to established medicinal chemistry motifs [34,35], suggesting that HiFrAMes extracts a compact set of reusable, chemically meaningful fragments.

To assess the behavior of HiFrAMes relative to established fragmentation approaches, we conducted qualitative and quantitative comparisons against two widely used fragmentation methods: RECAP and BRICS. Although all three methods aim to decompose molecules into reusable substructures, they differ fundamentally in how fragments are defined, organized, and contextualized. Figure 15 presents a side-by-side decomposition of vitamin B-12 using RECAP, BRICS, and HiFrAMes. RECAP produces a highly conservative decomposition, producing only a small number of large fragments in which most peripheral substituents remain attached to the corrin macrocycle or are grouped into single nucleotide-like fragments. BRICS applies more aggressive retrosynthetic bond cleavage,

separating many peripheral functional groups into small, disconnected fragments while still preserving a limited number of large cores. In contrast, HiFrAMes fragments the same molecule by separating major ring systems, such as the corrin core, benzimidazole, and sugar ring, into distinct subgraphs and isolates the phosphate-containing connector as its own fragment, while extracting numerous small chain fragments. Although HiFrAMes yields a larger number of fragments for this example than both RECAP and BRICS, the decomposition reflects the underlying roles of structural components, rather than simply cleaving predefined bond types.

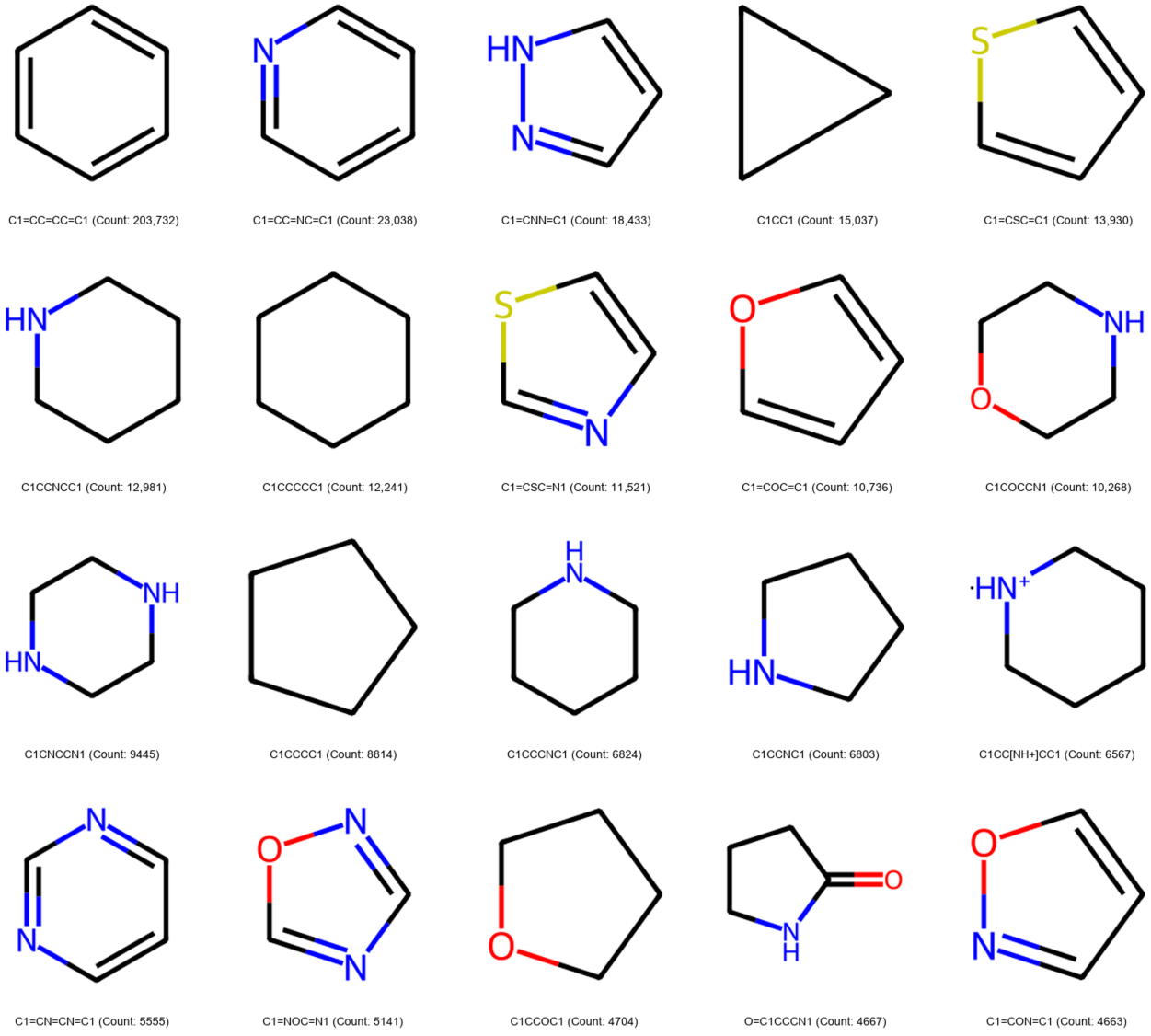

Figure 13. Top 20 most frequent ring fragments from ZINC-250K.

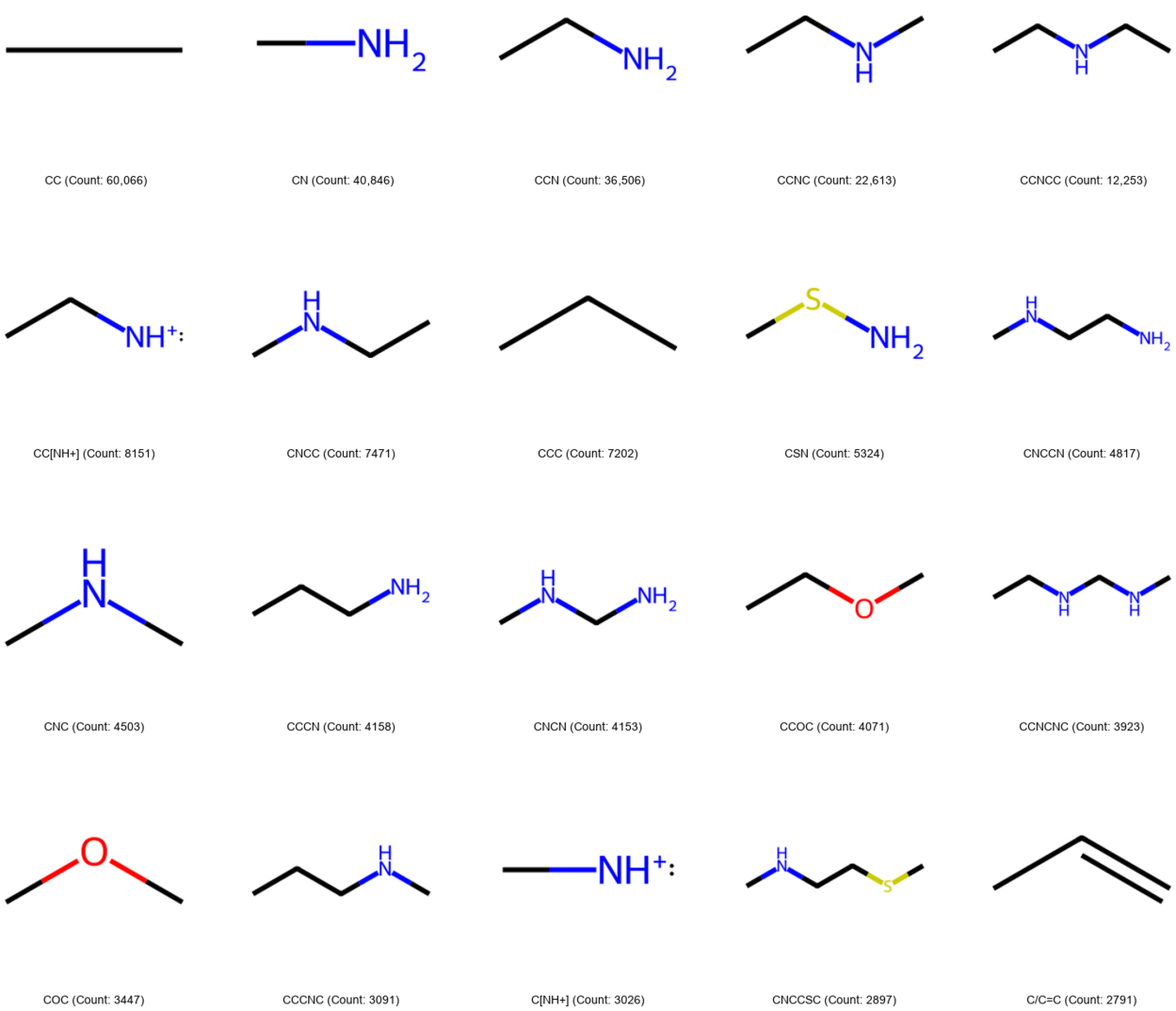

Figure 14. Top 20 most frequent linker fragments from ZINC-250K.

In contrast to the vitamin B-12 example, where molecular complexity exposes clear differences between fragmentation schemes, structurally simple molecules exhibit largely convergent behavior across methods. Figure 16 illustrates this for tiabendazole, a simple ZINC-250K molecule composed of a compact heteroaromatic core with no peripheral substituents and only a single cleavable linker. In this case, HiFrAMes, RECAP, and BRICS yield a nearly identical fragmentation (BRICS and RECAP produce the same fragments) that preserve the fused cyclic core as a single dominant fragment, with only a minor difference arising from the retention of a small linker moiety in HiFrAMes, as shown in the figure. This behavior reflects the topology of the molecular graph rather than algorithmic limitations, and indicates that HiFrAMes is most informative for sets of molecules with pronounced structural heterogeneity.

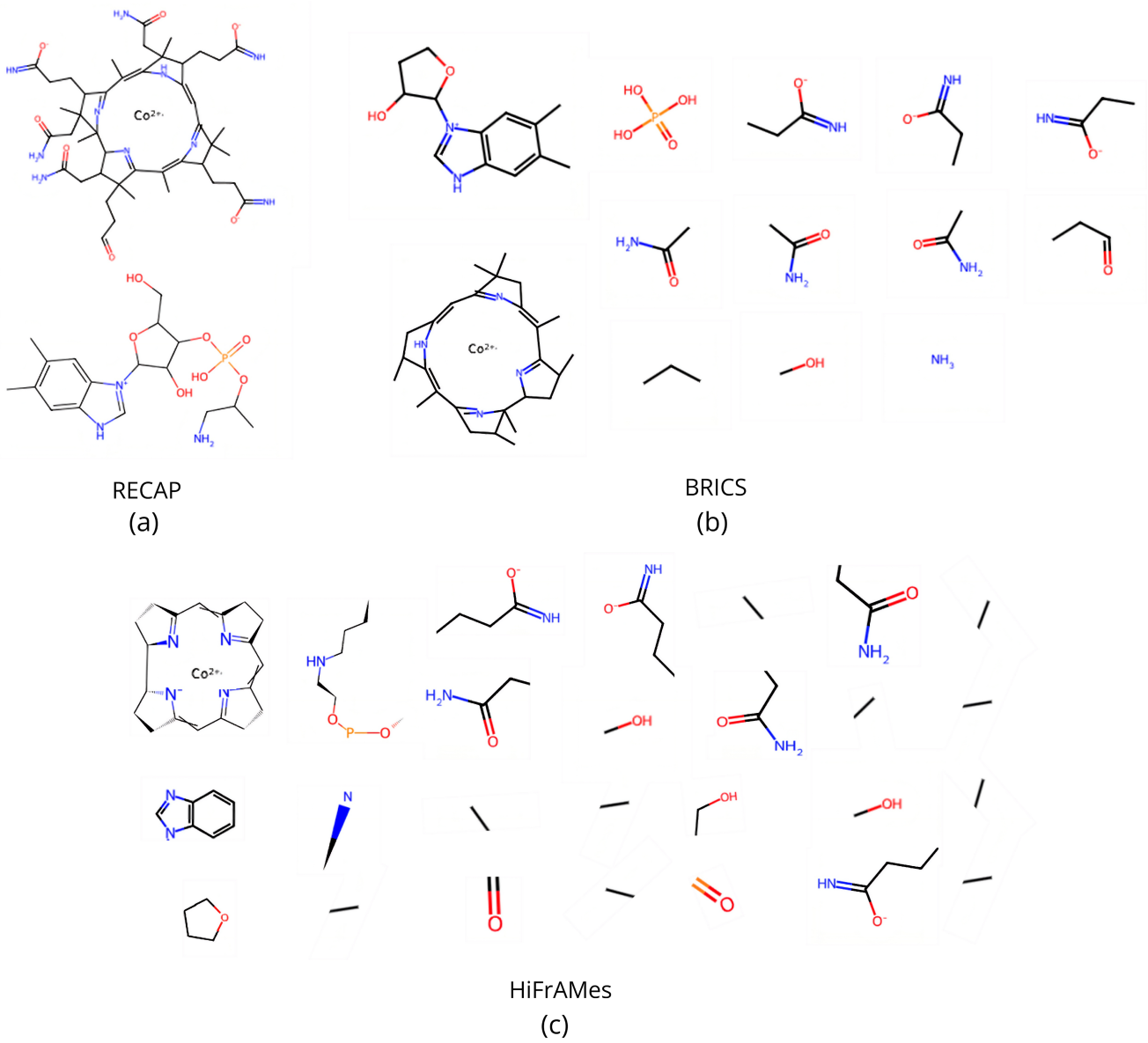

Figure 15. Fragmentation comparison for vitamin B-12. (**a**) Retrosynthetic Combinatorial Analysis Procedure (RECAP) fragments, (**b**) Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) fragments, and (**c**) HiFrAMes fragments.

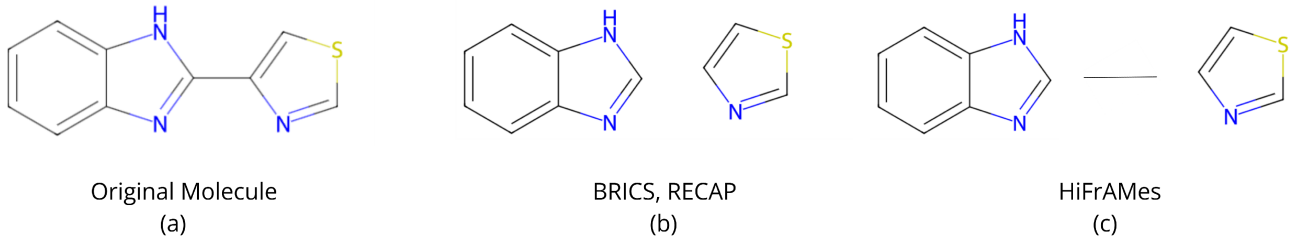

Figure 16. Fragmentation comparison for tiabendazole. (**a**) Original molecule, (**b**) BRICS and RECAP fragments, and (**c**) HiFrAMes fragments.

When HiFrAMes is applied to the entire ZINC-250K dataset, these observed differences are reflected in the fragment size statistics summarized in Table 1 and Figure 17. These results demonstrate that HiFrAMes generates a substantially more compact fragment vocabulary compared to RECAP and BRICS. This indicates markedly higher fragment reuse across molecules for HiFrAMes, suggesting that it captures recurring topological motifs

rather than producing a large number of idiosyncratic fragments. Consistent with FBDD criteria, 99.7 % of HiFrAMes fragments fall below 300 Da and 99.4% contain no more than 20 heavy atoms, exceeding the corresponding proportions for RECAP (93.8% and 92.6%) and BRICS (73.0% and 69.9%). These trends are clearly visible in the molecular weight and heavy atom count histograms, where HiFrAMes exhibits tightly bounded unimodal distributions with limited tails, whereas RECAP shows broader distributions, and BRICS produces a pronounced shift toward larger and more complex fragments. Notably, when both size criteria are applied jointly (MW $\leq$ 300 Da and HAC $\leq$ 20), 99.3% of HiFrAMes fragments satisfy fragment-like constraints, compared to 91.9% for RECAP and 68.3% for BRICS. These results demonstrate that HiFrAMes avoids the under-fragmentation observed in conservative rule-based methods, preventing the explosion of large or overly complex fragments characteristic of more aggressive retrosynthetic cleavage strategies, enabling greater fragment reuse.

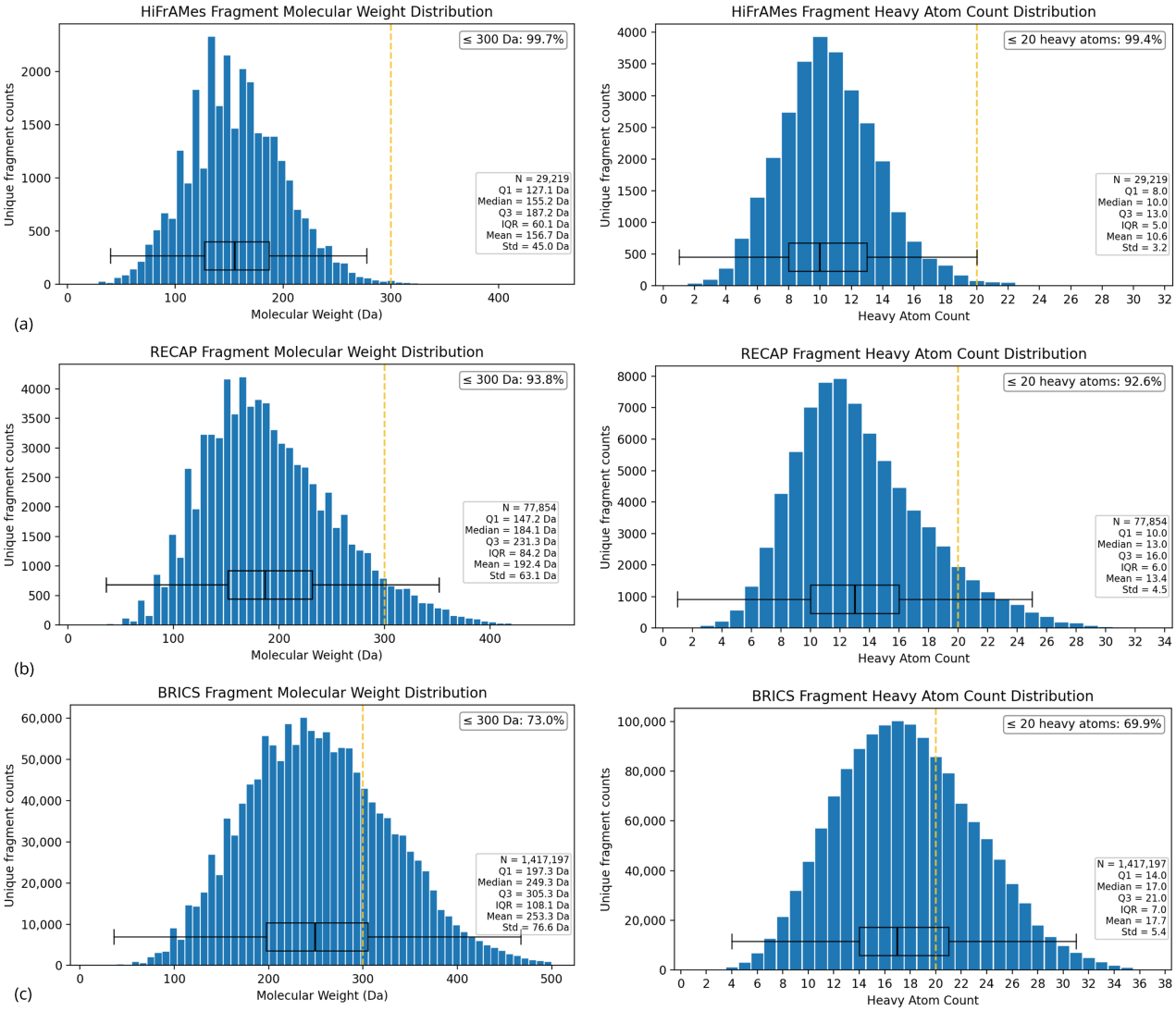

Figure 17. Fragment size distributions with box plots for (**a**) HiFrAMes, (**b**) RECAP, and (**c**) BRICS on ZINC-250K.

Table 1. Fragment library size and size-based statistics for HiFrAMes, RECAP, and BRICS.

Fragmentation Method	Total Fragment Instances	Unique Fragment Count	MW $\leq$ 300 (%)	HAC $\leq$ 20 (%)	MW $\leq$ 300 & HAC $\leq$ 20 (%)
HiFrAMes (All)	1,638,977	29,219	99.7	99.4	99.3
— Leaf Chains	718,427	19,803	99.9	100.0	99.9
— Rings	580,693	7153	99.0	97.7	97.6
— Linkers	339,857	2468	100.0	100.0	100.0
RECAP	635,020	77,854	93.8	92.6	91.9
BRICS	4,719,267	1,417,197	73.0	69.9	68.3

MW: molecular weight (Da). HAC: heavy atom count. HiFrAMes (All) is the union of leaf chains, rings, and linkers after deduplication. Counts for individual HiFrAMes categories do not sum to the total because a fragment may belong to both the leaf chain and linker categories.

In addition to fragment size, we evaluated other physicochemical characteristics of fragments produced by each method, including the commonly used Rule of Three (Ro3) selection criteria [36–38]. As summarized in Table 2, the HiFrAMes fragments exhibit consistently high compliance for individual constraints when compared with those of BRICS and RECAP, indicating that the fragments produced are not only appropriately sized but also chemically well-balanced.

Table 2. Common FBDD fragment selection criteria.

Fragmentation Method	MW $\leq$ 300 (%)	HAC $\leq$ 20 (%)	ClogP $\leq$ 3 (%)	HBD $\leq$ 3 (%)	HBA $\leq$ 3 (%)	NROT $\leq$ 3 (%)	TPSA $\leq$ 60 (%)
HiFrAMes	**99.7**	**99.4**	**97.7**	99.2	**90.4**	56.8	77.5
RECAP	93.8	92.6	93.6	**99.9**	73.4	**75.8**	**83.1**
BRICS	73.0	69.9	82.6	99.3	59.8	56.6	58.3

MW: molecular weight (Da). HAC: heavy atom count. ClogP: calculated octanol–water partition coefficient. HBD: hydrogen bond donors. HBA: hydrogen bond acceptors. NROT: number of rotatable bonds. TPSA: topological polar surface area. The highest compliance rate for each property is bolded.

When compliance with all Ro3 metrics is evaluated, requiring satisfaction of MW, ClogP, hydrogen bond donors (HBD), and hydrogen bond acceptors (HBA) thresholds, 87.6% of HiFrAMes fragments meet all four criteria, substantially exceeding the corresponding Ro3 compliance rates for RECAP (68.0%) and BRICS (45.9%), as shown in Table 3. We further evaluated fragment quality using an extended seven-criterion framework that includes HAC, number of rotatable bonds (NROT), and topological polar surface area (TPSA) [26], in addition to the Ro3 criteria. Under this set of constraints, 42.8% of HiFrAMes fragments are fully compliant, compared to 51.2% of RECAP fragments and 26.6% of BRICS fragments meeting all criteria; however, it is well established that successful fragments may violate at least one commonly used selection criterion without compromising ligand efficiency or downstream developability [39]. Relaxing the constraint of full compliance, we also evaluate the rates of high extended compliance of each set of fragments, in which a fragment must meet at least six of the seven thresholds. This allows for tolerance to single-criterion violations while still evaluating overall fragment appropriateness. We observed that 82.5% of HiFrAMes fragments satisfy at least six of the seven fragment-likeness thresholds, while 78.1% of RECAP fragments and only 49.3% of BRICS fragments do the same. Altogether, our results indicate that HiFrAMes produces appropriately sized, chemically balanced fragments that are well-aligned with common motifs observed in medicinal chemistry and are highly reusable, making them well-suited for downstream fragment-oriented computational tasks.

Table 3. Aggregate fragment compliance across fragmentation methods.

Fragmentation Method	Ro3 Compliance (4/4) (%)	High Extended Compliance ($\geq$6/7) (%)	Fully Compliant (7/7) (%)
HiFrAMes	**87.6**	**82.5**	42.8
RECAP	68.0	78.1	**51.2**
BRICS	45.9	49.3	26.6

Rule of Three (Ro3) compliance (4/4) requires all four Ro3 thresholds to be satisfied: molecular weight (MW) $\leq$ 300 Da, calculated octanol–water partition coefficient (ClogP) $\leq$ 3, hydrogen bond donors (HBD) $\leq$ 3, and hydrogen bond acceptors (HBA) $\leq$ 3. High extended compliance ($\geq$6/7) indicates that at least six of the seven fragment-likeness criteria are satisfied. Full compliance (7/7) requires all seven criteria to be satisfied: MW $\leq$ 300 Da, heavy atom count (HAC) $\leq$ 20, ClogP $\leq$ 3, HBD $\leq$ 3, HBA $\leq$ 3, number of rotatable bonds (NROT) $\leq$ 3, and topological polar surface area (TPSA) $\leq$ 60 Å^2. The highest compliance rate for each aggregate is bolded.

5. Conclusions

We introduce HiFrAMes, a hierarchical, graph–theoretic framework that decomposes molecules into chemically meaningful fragments through four sequential stages and organizes them into scaffold abstractions while preserving global topology. This multi-stage pipeline is efficiently scalable for molecular graphs, making it practical for generating interpretable fragments from large molecular libraries. We evaluate our framework through a comparison of the physicochemical characteristics of HiFrAMes fragments against those of fragments generated using established rule-based retrosynthetic fragmentation methods, as well as through multiple molecular case studies. HiFrAMes focuses on topological abstraction, leaving stereochemical reassignment and inference as a potential direction for future work. Across the FBDD literature-established fragment selection criteria evaluated, HiFrAMes fragments show higher compliance with commonly used thresholds for fragment selection than other methods, indicating that they more frequently fall within ranges used in fragment-based drug discovery. Additionally, the fragments extracted using HiFrAMes from ZINC-250K align well with drug-like motifs discussed in the medicinal chemistry literature. Case studies, such as that of vitamin B-12, demonstrate that the pipeline successfully isolates core cyclic frameworks, identifies linker regions, and cleanly separates terminal substituents, enabling construction of compact scaffold representations, even for large, structurally complex molecules. Overall, HiFrAMes provides a practical and flexible framework for fragment-based drug discovery workflows, producing fragments and scaffold representations that are suitable for downstream fragment-based computational screening and optimization tasks.

Author Contributions: Conceptualization, Y.Y. and J.-C.L.; methodology, Y.Y. and J.-C.L.; software, Y.Y.; formal analysis, Y.Y. and M.A.S.; investigation, Y.Y.; data curation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y., M.A.S., H.W. and J.-C.L.; visualization, Y.Y. and M.A.S.; supervision, J.-C.L.; project administration, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used BioRender for the purposes of generating some of the figures.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADME	Absorption, Distribution, Metabolism, and Excretion
BPE	Byte-Pair Encoding
FBDD	Fragment-Based Drug Discovery
HAC	Heavy Atom Count
HBA	Hydrogen Bond Acceptors
HBD	Hydrogen Bond Donors
HRMS	High-Resolution Mass Spectrometry
MMP	Matched Molecular Pair
MW	Molecular Weight
NLP	Natural Language Processing
NMR	Nuclear Magnetic Resonance
NROT	Number of Rotatable Bonds
RECAP	Retrosynthetic Combinatorial Analysis Procedure
Ro3	Rule of Three
SMILES	Simplified Molecular Input Line Entry System
SPR	Structure–Property Relationship
TPSA	Topological polar surface area

Appendix A

Algorithm A1: Leaf Chain Enumeration with Exocyclic Ring Atoms Integrated

Input : Graph $G(V, E)$
Output: Leaf chain induced subgraphs $(V_{LC_filtered}, E_{LC_filtered})$ and ring system induced subgraph (V_{RS}, E_{RS})
Function *EnumerateLeafChains*(G):
  $V_{LC} \leftarrow [\,], E_{LC} \leftarrow [\,], V_{RS} \leftarrow [\,], E_{RS} \leftarrow [\,]$
  `// Collect initial leaves`
  $queue \leftarrow$ empty list
  foreach $v \in V$ **do**
    if $deg(v) = 1$ **then**
      $queue$.`append`(v)
  while $|queue| > 0$ **do**
    $v = queue$.`popleft`$()$
    `// One and only one neighbor`
    $u \leftarrow$ neighbors$(v)[0]$
    remove edge (u, v) from E
    remove vertex v from V
    V_{LC}.`append`(v)
    E_{LC}.`append`$((u, v))$
    if $deg(u) = 1$ **then**
      $queue$.`append`(u)
  $V_{RS} \leftarrow$ vertices remaining in V
  $E_{RS} \leftarrow$ edges remaining in E
  `// Add exocyclic ring atoms into ring system`
  $R_{\text{exo}} \leftarrow$ `GetExocyclicRingAtoms`(G)
  $V_{RS} \leftarrow$ `AddVertices`(V_{RS}, R_{exo})
  $E_{RS} \leftarrow$ `UpdateEdges`(G, V_{RS})
  `// Filter leaf chains based on updated ring system`
  $(V_{LC_filtered}, E_{LC_filtered}) \leftarrow$ `FilterLeafChainsByRingSystem`(V_{LC}, E_{LC}, V_{RS})
  return $(V_{LC_filtered}, E_{LC_filtered}), (V_{RS}, E_{RS})$

Algorithm A2: Ring Mesh Reduction

Input : Subgraph mesh G′(V′, E′) without leaf chains

Output: Set of reduced edges *reducedEdges*, where each reduced edge is an induced path connecting two furcation nodes

Function `ConstructReducedEdges`(G')**:**

- $fNodes \leftarrow \{v \in V' \; if \; \texttt{deg}(v) > 2\}$ // furcation nodes
- $reducedEdges \leftarrow [\,]$
- $visitedNodes \leftarrow \{\,\}$
- $visitedEdges \leftarrow \{\,\}$
- **foreach** $f \in fNodes$ **do**
  - **foreach** $u \in$ `neighbors`(f) **do**
    - **if** `deg`$(u) = 2$ *and* $u \notin visitedNodes$ **then**
      - $path \leftarrow [f]$
      - $prev \leftarrow f, v \leftarrow u$
      - // grow a degree-2 chain starting from edge (f, u)
      - **while** `deg`$(v) = 2$ *and* $v \notin visitedNodes$ **do**
        - $path$.`append`(v)
        - $visitedNodes$.`add`(v)
        - $x, y \leftarrow$ `neighbors`(v)
        - $next \leftarrow x$ if $x \neq prev$ else y
        - $prev \leftarrow v, v \leftarrow next$
      - // terminate if v is a furcation node (including f) or a visited linking node
      - **if** `deg`$(v) > 2$ **then**
        - $path$.`append`(v)
        - $reducedEdges$.`append`$(path)$
    - **else**
      - **if** `deg`$(u) > 2$ **then**
        - // single-edge path (f, u)
        - **if** $(f, u) \notin visitedEdges$ **then**
          - $visitedEdges.add((f, u))$
          - $visitedEdges.add((u, f))$
          - $reducedEdges$.`append`$([f, u])$
  - return *reducedEdges*

Algorithm A3: Ring Enumeration

Input : Undirected MultiGraph G
Output: A list of edge-unique cycles, each as a list of $(u, v, edge)$
Function *EnumerateAllEdgeCycles(G)*:

```
parallelEdges ← FindParallelEdges(G)
allCycles ← empty list
// Retrieve all node-only simple cycles (using existing method)
foreach nodeCycle ∈ simple_cycles(G) do
    // Ensure the cycle ends at its start, if not already
    if nodeCycle.last ≠ nodeCycle.first then
        nodeCycle.append(nodeCycle.first)
    // Split each node-unique cycle into edge-unique cycles
    edgeUniqueCycles ← SplitCycleByParallelEdges(nodeCycle, parallelEdges)
    foreach edgeCycle ∈ edgeUniqueCycles do
        allCycles.append(edgeCycle)
return allCycles
```

Input : Undirected MultiGraph G (with possible parallel edges and self-loops)
Output: A mapping from unordered node-pair (u, v) to list of edges between node pairs with parallel edges
Function *FindParallelEdges(G)*:

```
parallelEdges ← empty map
foreach (u, v, edge) ∈ G do
    p ← OrderedPair(u, v)
    if p ∉ parallelEdges then
        parallelEdges[p] ← empty list
    parallelEdges[p].append(edge)
return parallelEdges
```

Function *OrderedPair(u,v)*:

```
return u <= v ? (u, v) : (v, u)
```

https://doi.org/10.3390/ai7020071

Algorithm A3: *Cont.*

Input : nodeCycles: a list of nodes $[v_0, v_1, \ldots, v_n]$ with $v_n = v_0$,
parallelEdges: a mapping from (u, v) pair to list of edges
Output: List of all edge-unique cycles of nodes in `node_cycle`, each as a list of $(u, v, edge)$
Function *SplitCycleByParallelEdges(nodeCycle, parallelEdges)*:

```
n ← |nodeCycle|
if n > 1 then
    n ← n − 1
cycles ← empty list
// 1-node:  self loop
if n = 1 then
    u ← nodeCycle[0]
    edges ← parallelEdges[ OrderedPair(u, u) ]
    foreach edge_key ∈ edges do
        cycles.append( [(u, u, edge_key)] )
    return cycles;
// 2-node simple case:  choose 2 distinct edges
if n = 2 then
    u ← nodeCycle[0], v ← nodeCycle[1]
    edges ← parallelEdges[ OrderedPair(u, v) ]
    for i ← 0 to |edges| − 2 do
        for j ← i + 1 to |edges| − 1 do
            cycles.append( [ (u, v, edges[i]), (v, u, edges[j]) ] )
    return cycles;
// more than 2 nodes, enumerate all edge combinations
steps ← empty list
foreach i ∈ 0, ..., n − 1 do
    steps.append(OrderedPair(nodeCycle[i], nodeCycle[i + 1]))
choices ← [parallelEdges[pair] for pair in steps]
choicePerStep ← [0] ∗ n
while true do
    cycle ← empty list
    foreach i ∈ 0, ..., n − 1 do
        edge = choices[i][choicePerStep[i]]
        cycle.append((steps[i][0], steps[i][1], edge))
    cycles.append(cycle)
    // Reset edge selection starting from the last step
    i ← n − 1
    while i >= 0 and choicePerStep[i] = |choices[i]| − 1 do
        choicePerStep[i] = 0
        i ← i − 1
    // Counter is maxed out (all cycles enumerated)
    if i < 0 then
        break
    // Update edge selection at step i
    choicePerStep[i] ← choicePerStep[i] + 1
return cycles
```

Algorithm A4: Linker Extraction

Input : G: an undirected MultiGraph (with parallel edges),
cycles: a list of all edge-unique cycles
parallelEdges: map from (u, v) pair to list of edges

Output: List of edges $(u, v, edge)$ in G not contained in any cycle

Function *FindLinkerEdges(G, cycles, parallelEdges)*:

```
cycleEdges ← empty set
// Collect every edge that appears in any cycle
foreach cycle ∈ cycles do
    foreach (u, v, edge) ∈ cycle do
        pair = OrderedPair(u, v)
        cycleEdges.add(pair[0], pair[1], edge)

linkerEdges ← empty list
// Any edge of G not in CycleEdges is a linker
foreach (u, v, edge) ∈ G do
    pair = OrderedPair(u, v)
    if (pair[0], pair[1], edge) ∉ cycleEdges then
        linkerEdges.append(p)

return linkerEdges
```

References

1. Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A.J.; Bambrick, J.; et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *630*, 493–500. [CrossRef] [PubMed]
2. Song, B.; Zhang, J.; Liu, Y.; Liu, Y.; Jiang, J.; Yuan, S.; Zhen, X.; Liu, Y. A systematic review of molecular representation learning foundation models. *Brief. Bioinform.* **2026**, *27*, bbaf703. [CrossRef] [PubMed]
3. Evans-Newman, K.C.; Schneider, G.L.; Perera, N.T. Classification of mass spectral data to assist in the identification of novel synthetic cannabinoids. *Molecules* **2024**, *29*, 4646. [CrossRef] [PubMed]
4. Teli, D.M.; Patel, B.; Chhabria, M.T. Fragment-based design of SARS-CoV-2 Mpro inhibitors. *Struct. Chem.* **2022**, *33*, 2155–2168. [CrossRef] [PubMed]
5. Dueñas, M.E.; Peltier-Heap, R.E.; Leveridge, M.; Annan, R.S.; Büttner, F.H.; Trost, M. Advances in high-throughput mass spectrometry in drug discovery. *EMBO Mol. Med.* **2023**, *15*, e14850. [CrossRef] [PubMed]
6. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2015**, arXiv:1508.07909. [CrossRef]
7. Brown, N.; Jacoby, E. On scaffolds and hopping in medicinal chemistry. *Mini Rev. Med. Chem.* **2006**, *6*, 1217–1229. [CrossRef]
8. Bemis, G.W.; Murcko, M.A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **1996**, *39*, 2887–2893. [CrossRef]
9. Bui, L.; Djikic-Stojsic, T.; Bret, G.; Bihel, F.; Kellenberger, E. Scaffold-based libraries versus make-on-demand space: A comparative assessment of chemical content. *ChemMedChem* **2025**, *20*, e202500518. [CrossRef]
10. Lewell, X.Q.; Judd, D.B.; Watson, S.P.; Hann, M.M. 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**, *38*, 511–522. [CrossRef]
11. Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem* **2008**, *3*, 1503–1507. [CrossRef]
12. Diao, Y.; Hu, F.; Shen, Z.; Li, H. MacFrag: Segmenting large-scale molecules to obtain diverse fragments with high qualities. *Bioinformatics* **2023**, *39*, btad012. [PubMed]
13. Yang, Y.; Zheng, S.; Su, S.; Zhao, C.; Xu, J.; Chen, H. SyntaLinker: Automatic fragment linking with deep conditional transformer neural networks. *Chem. Sci.* **2020**, *11*, 8312–8322. [CrossRef] [PubMed]
14. Landry, M.L. Deriving insights for molecular design with MMP analysis. *Trends Chem.* **2024**, *6*, 346–348. [CrossRef]
15. Ji, Z.; Shi, R.; Lu, J.; Li, F.; Yang, Y. ReLMole: Molecular representation learning based on two-level graph similarities. *J. Chem. Inf. Model.* **2022**, *62*, 5361–5372. [CrossRef]

16. Ye, X.B.; Guan, Q.; Luo, W.; Fang, L.; Lai, Z.R.; Wang, J. Molecular substructure graph attention network for molecular property identification in drug discovery. *Pattern Recognit.* **2022**, *128*, 108659. [CrossRef]
17. Gaulton, A.; Bellis, L.J.; Bento, A.P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; et al. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.
18. Yu, Y. HiFRAMES. Created in BioRender. 2026. Available online: https://BioRender.com/22cdsz2 (accessed on 28 January 2026).
19. Hagberg, A.A.; Schult, D.A.; Swart, P.J. Exploring Network Structure, Dynamics, and Function Using NetworkX. In Proceedings of the 7th Python in Science Conference (SciPy 2008), Pasadena, CA, USA, 19–24 August 2008; pp. 11–15.
20. Johnson, D.B. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.* **1975**, *4*, 77–84. [CrossRef]
21. NetworkX Developers. NetworkX Documentation: Simple Cycles. 2024. Available online: https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.cycles.simple_cycles.html (accessed on 28 January 2026).
22. National Center for Biotechnology Information. PubChem Compound Summary for CID 73415824, Cobalamin. 2025. Available online: https://pubchem.ncbi.nlm.nih.gov/compound/Cobalamin (accessed on 29 December 2025).
23. Darke, S.; Banister, S.; Farrell, M.; Duflou, J.; Lappin, J. 'Synthetic cannabis': A dangerous misnomer. *Int. J. Drug Policy* **2021**, *98*, 103396. [CrossRef]
24. National Center for Biotechnology Information. PubChem Compound Summary for CID 83933, Erythromycin C. 2025. Available online: https://pubchem.ncbi.nlm.nih.gov/compound/83933 (accessed on 29 December 2025).
25. Irwin, J.J.; Shoichet, B.K. ZINC—A free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.
26. Bon, M.; Bilsland, A.; Bower, J.; McAulay, K. Fragment-based drug discovery-the importance of high-quality molecule libraries. *Mol. Oncol.* **2022**, *16*, 3761–3777.
27. Carbery, A.; Skyner, R.; von Delft, F.; Deane, C.M. Fragment libraries designed to be functionally diverse recover protein binding information more efficiently than standard structurally diverse libraries. *J. Med. Chem.* **2022**, *65*, 11404–11413. [CrossRef] [PubMed]
28. Erlanson, D.A.; Fesik, S.W.; Hubbard, R.E.; Jahnke, W.; Jhoti, H. Twenty years on: The impact of fragments on drug discovery. *Nat. Rev. Drug Discov.* **2016**, *15*, 605–619. [CrossRef] [PubMed]
29. AlKharboush, D.F.; Kozielski, F.; Wells, G.; Porta, E.O.J. Fragment-based drug discovery: A graphical review. *Curr. Res. Pharmacol. Drug Discov.* **2025**, *9*, 100233. [CrossRef] [PubMed]
30. Wang, Z.Z.; Shi, X.X.; Huang, G.Y.; Hao, G.F.; Yang, G.F. Fragment-based drug discovery supports drugging 'undruggable' protein-protein interactions. *Trends Biochem. Sci.* **2023**, *48*, 539–552. [CrossRef]
31. Ramsden, J.I.; Cosgrove, S.C.; Turner, N.J. Is it time for biocatalysis in fragment-based drug discovery? *Chem. Sci.* **2020**, *11*, 11104–11112. [CrossRef]
32. Meanwell, N.A. Improving drug candidates by design: A focus on physicochemical properties as a means of improving compound disposition and safety. *Chem. Res. Toxicol.* **2011**, *24*, 1420–1456. [CrossRef]
33. Lipinski, C.A.; Lombardo, F.; Dominy, B.W.; Feeney, P.J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3–25. Reprint in *Adv. Drug Deliv. Rev.* **2001**, *46*, 3–26. [CrossRef]
34. Ertl, P. Cheminformatics analysis of organic substituents: Identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 374–380.
35. Taylor, R.D.; MacCoss, M.; Lawson, A.D.G. Rings in drugs. *J. Med. Chem.* **2014**, *57*, 5845–5859. [CrossRef]
36. Congreve, M.; Carr, R.; Murray, C.; Jhoti, H. A 'Rule of Three' for fragment-based lead discovery? *Drug Discov. Today* **2003**, *8*, 876–877. [CrossRef]
37. Keserű, G.M.; Erlanson, D.A.; Ferenczy, G.G.; Hann, M.M.; Murray, C.W.; Pickett, S.D. Design principles for fragment libraries: Maximizing the value of learnings from pharma fragment-based drug discovery (FBDD) programs for use in academia. *J. Med. Chem.* **2016**, *59*, 8189–8206. [CrossRef]
38. Scott, D.E.; Coyne, A.G.; Hudson, S.A.; Abell, C. Fragment-based approaches in drug discovery and chemical biology. *Biochemistry* **2012**, *51*, 4990–5003. [CrossRef]
39. Köster, H.; Craan, T.; Brass, S.; Herhaus, C.; Zentgraf, M.; Neumann, L.; Heine, A.; Klebe, G. A small nonrule of 3 compatible fragment library provides high hit rate of endothiapepsin crystal structures with various fragment chemotypes. *J. Med. Chem.* **2011**, *54*, 7784–7796. [CrossRef]

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