Compressing Chemistry Reveals Functional Groups

Downloaded from: https://research.chalmers.se, 2026-09-05 03:00 UTC

Citation for the original published paper (version of record):
Sharma, R., King, R. (2026). Compressing Chemistry Reveals Functional Groups. Journal of Chemical Information and Modeling, 66(7): 3562-3574. http://dx.doi.org/10.1021/acs.jcim.5c02917

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Open Access

This article is licensed under CC-BY 4.0

pubs.acs.org/jcim

Article

Compressing Chemistry Reveals Functional Groups

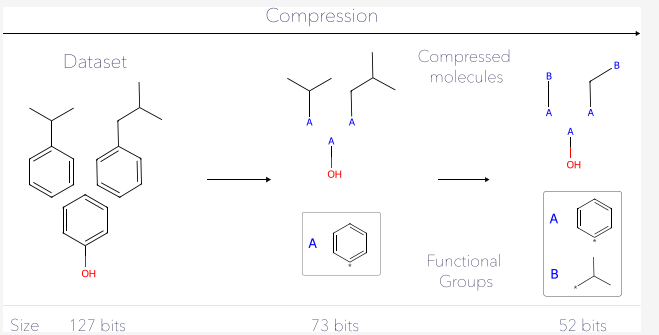
Ruben Sharma* and Ross D. King

Cite This: J. Chem. Inf. Model. 2026, 66, 3562−3574

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We introduce the first formal large-scale assessment of the utility of traditional chemical functional groups as used in chemical explanations. Our assessment employs a fundamental principle from computational learning theory: a good compression of data should reveal a good explanation. We introduce an unsupervised learning algorithm based on the Minimum Message Length (MML) principle that searches for substructures that compress around three million biologically relevant molecules. We demonstrate that the discovered substructures contain most human-curated functional groups as well as novel larger patterns with more specific functions. We also run our algorithm on 24 specific bioactivity prediction data sets to discover data set-specific functional groups. Fingerprints constructed from data set-specific functional groups are shown to significantly outperform other fingerprint representations, including the MACCS and Morgan fingerprint, when training ridge regression models on bioactivity regression tasks.



Downloaded via CHALMERS UNIV OF TECHNOLOGY on May 11, 2026 at 11:28:27 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

INTRODUCTION

Functional groups are a human-curated set of molecular substructures which are useful for expressing chemical explanations. In organic and biological chemistry molecules are often described as several connected functional groups.[1] In particular, explanations of chemical and biochemical activity are often described in terms of functional groups. For example, the efficacy of several antibiotic compounds requires the compound to contain the beta-lactam functional group, benzodiazepines are characterized by benzene and diazepine functionalities and NSAIDs contain the propionic acid functional group attached to an aromatic group (see Figure 1).[2−4] Caching useful substructures that occur in several explanations enables concise descriptions of molecules and their properties. Concise explanations are in line with Occam's Razor, a preference for simpler (formalized in computer science as shorter) explanations. Specifically, in Solomonoff's formal theory of inductive inference, the ability of a theory to compress data serves as a formally objective and mathematically optimal measure of the theory's explanatory and predictive power.[5] Due to their historical utility, functional groups are often used as a default chemical representation format. In this work we aim to answer the question:

Do substructures that compress a large corpus of biological molecules correspond to human-curated functional groups?

There exists no canonical functional group set. A positive answer to our research question could establish an objective basis for constructing such a list.

Aside from its intellectual interest, an objective functional group set would enable the development of new chemical similarity measures and representation techniques, such as molecular tokenisation schemes, and more. Further, a method of extracting truly useful substructures from data sets should improve the predictive performance of machine learning and clustering algorithms on chemical data sets.

There are several ways of formulating the compression problem. We choose to employ the Minimum Message Length (MML) principle.[6,7] MML aims to find the most concise explanation of data, where an explanation is a two-part message consisting of a hypothesis (the first part) followed by the data given that the hypothesis is true (the second part). Under MML, the most concise explanation trades off the complexity of the hypothesis, and how well the hypothesis fits the data. A complex hypothesis (large first part) must fit the data very well (small second part) to be selected over a simple hypothesis (small first part) which fits the data moderately well (moderately sized second part).

In this manuscript, we restrict the types of considered hypotheses to "substructure-based explanations." Specifically, we model the SMILES representation of a chemical data set as if it were generated by a series of independent draws from a multinomial distribution over SMILES substrings, which correspond to substructures.[8] We search for the substrings

Received: November 28, 2025
Revised: February 16, 2026
Accepted: February 18, 2026
Published: March 18, 2026

© 2026 The Authors. Published by American Chemical Society

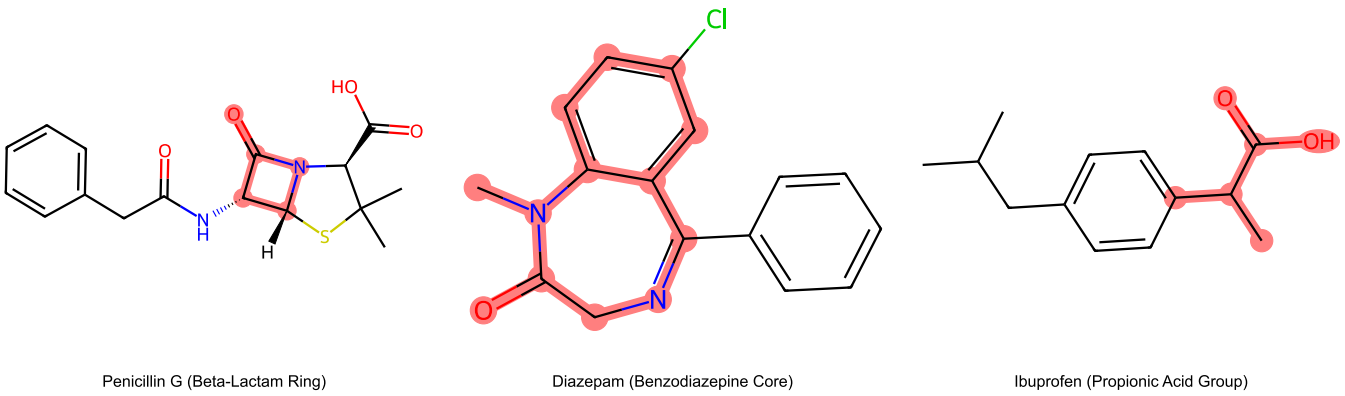


Figure 1. Example drug molecules and necessary functional groups.

and multinomial probabilities which best compress the data set according to MML. We use MML because the framework automatically chooses the continuous probability parameters of the multinomial distribution. MML's handling of continuous parameters provides automatic regularisation.

Specifically, we claim:

Claim 1: Standard chemical functional groups are objectively useful in general explanations of bioactivity as they emerge during compression of a large corpus of biologically relevant molecules.

Claim 2: Substructures that compress a data set are useful features for machine learning tasks.

Overall, our contributions are

Contribution 1: We introduce an unsupervised algorithm that identifies a set of compressing substructures from a string data set, in line with the MML principle.

Contribution 2: We run our compression algorithm on a data set containing almost three million biologically relevant molecules represented as SMILES strings and show that the discovered substructures validate conventional functional group theory. We provide a list of these substructures.

Contribution 3: We use the substructures learned by our compression algorithm to generate chemical fingerprints. We compare the performance of the chemical fingerprints against MACCS fingerprints, Morgan fingerprints, and neural molecular embeddings when learning linear models from bioactivity data.[9]

RELATED WORK

Molecular Representation

Most approaches to molecular representation convert chemical structures into vector representations suitable for computation. Vector representations may be continuous, discrete or propositional. We describe each in turn.

A continuous vector representation of a molecule is a sequence of real numbers

$$\boldsymbol{v}_c(m) = [v_1, v_2, ..., v_n] \in \mathbb{R}^n \quad (1)$$

where the vector $\boldsymbol{v}_c(m)$ encodes information derived from the molecule's structure, composition, or associated data. Examples of continuous vector representations include learned embeddings, where a machine learning model such as a graph neural network learns to map molecular structures into continuous spaces that capture chemical similarity and bioactivity.[10] In contrast, each vector index in our representation corresponds to a specific feature.

Discrete vector representations are vectors of integer-valued components. A discrete vector representation is defined as

$$\boldsymbol{v}_d(m) = [d_1, d_2, ..., d_n] \in \mathbb{Z}^n \quad (2)$$

where each d_i usually encodes a count or categorical indicator of a structural or compositional feature. Count-based molecular fingerprints are a common example of discrete vector representations. Each index of a count-based molecular fingerprint corresponds to the integer count of a particular feature, such as a substructure. Binary vector representations are a special case of discrete vector representations, where the components are restricted to elements of the set $\{0, 1\}$. Perhaps the most popular discrete vector representation is the Morgan fingerprint.[11] Morgan fingerprints are constructed by enumerating all substructures within a user-defined radius around each heavy atom, assigning each substructure a unique numerical identifier, and hashing these identifiers into a fixed-length binary vector. The hashing operation, however, does not guarantee a one-to-one mapping between substructures and vector indices. Consequently, some indexes of a discrete vector representation may correspond to multiple substructures. Such events are known as "bit collisions" but are rare in practice. In this report, we learn discrete, count-based vector representations of molecules. In contrast to the Morgan fingerprint, no hashing is applied: Each vector index explicitly represents the count of a unique substructure. Moreover, our approach operates on a substantially smaller set of substructures.

Another popular class of representations is propositional vector representations. Propositional vector representations are vectors with boolean-valued components.

$$\boldsymbol{v}_p(m) = [p_1, p_2, ..., p_n] \in \{\text{True}, \text{False}\}^n \quad (3)$$

Each component of a propositional representation corresponds to a logical proposition, which can either be true or false. For example, propositions may correspond to specific abstractions such as "*the chemical has a 7-membered ring*" or "*the chemical contains an actinide*" or specific groups "*the chemical has the functional group* `NC(O)N`." One popular propositional representation is the Molecular ACCess Systems keys fingerprint (MACCS) which is of length 166.[9,11] Each of the 166 entries corresponds to the truth value of a proposition. In contrast our approach uses a discrete vector representation, does not consider propositions which do not correspond to fully specified substructures and has variable length depending on the data set.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

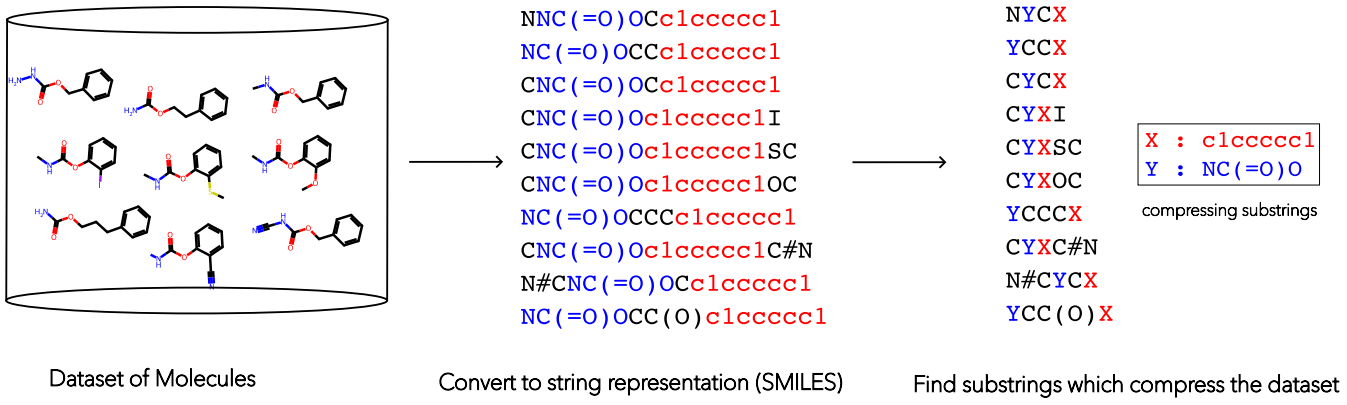


Figure 2. Procedure for extracting substructures (FGCompress).

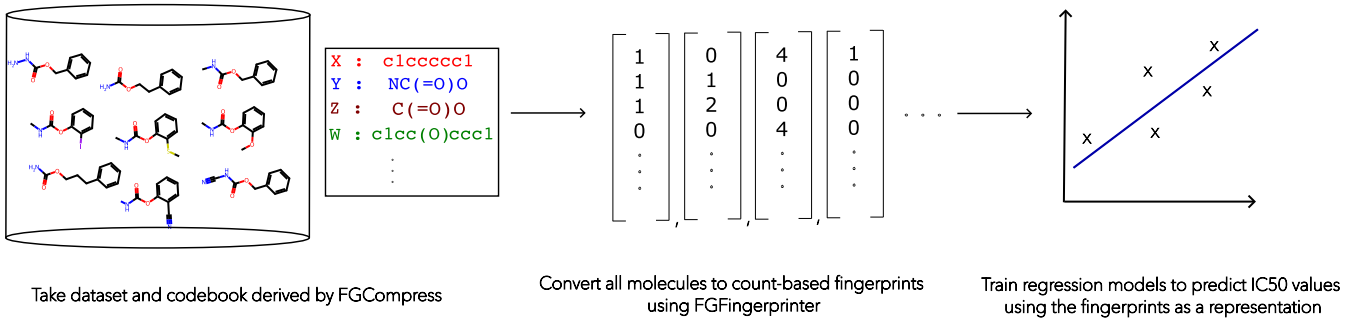


Figure 3. Overall FGFingerprinter Procedure. A data set and codebook are used to convert the molecules in the data set into vectors (count based fingerprints). The vectors are then used to train ridge regression models to predict IC_{50} values.

Compression and Pattern Finding

The most relevant prior works include the OSCR and MDLCompress algorithms.[12−14] Similarly, these algorithms seek to compress the data set using repeating substrings. In contrast to our approach, these algorithms do not calculate the exact length of the compressed data set and rely on heuristics. We also modify our algorithm to extract only those substrings which correspond to valid chemical substructures.

Identifying Functional Groups

The most similar work to ours is Erten et al.[15] which also uses an Occamist bias to discover functional groups in an unsupervised fashion. The authors consider a logic programming approach to functional group learning which is more rigorous, but also computationally much more expensive. In contrast to our work, the authors do not run the algorithm on a large scale data set, and do not report all discovered substructures. The authors also do not compare their discovered substructures to existing functional groups, and do not present a method of using them in supervised learning. Another similar work is Ertl.[16] The authors define functional groups as those which conform to a specified set of rules. The authors use prior chemical knowledge and a programming procedure to automatically partition molecules into functional groups. Conversely, we use no prior chemical knowledge and do not impose any definition regarding the structure of the groups we wish to extract, except those which are syntactically invalid. We simply aim to find groups which may be used to compress a data set in a lossless manner.

METHODOLOGY

We now describe our experimental procedure. To test our claims, we employ two algorithms.

1. FGCompress: This algorithm finds a set of compressing substructures from a data set.
2. FGFingerprinter: This algorithm uses a set of substructures to convert a molecule to a vector representation.

We use the algorithms to test claims 1 and 2. The experiments for each follow

Experiment 1: Assessing Functional Groups

To assess claim 1, we run FGCompress on a large set of biological molecules (ChEMBL) to extract compressing patterns. Due to computational restrictions we stop searching for patterns after extracting the top 500 most compressive substructures. We then manually assess the extracted substructures to determine if they correspond to human functional groups. We assess structures this way as there is no appropriate canonical human functional group set for comparison. A pictorial representation of the FGCompress algorithm is shown in Figure 2 below. To ensure computational tractability we convert the data set into the SMILES string representation and search for compressing substrings (contiguous lists of SMILES characters) that correspond to chemical substructures. The compression is lossless: together, the compressing substructures and compressed data set can be decoded to reconstruct the original molecules.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

Step	Codebook	Compressed Data	Length/bits
0	{}	{C=C(Br)C=O, C=CCOC=O, CC=CC=O}	32.1
1	{X:C=}	{XC(Br)XO, XCCOXO, CXXO}	18.2
2	{Y:C=C, X:C=}	{Y(Br)XO, YCOXO, CXXO}	16.6
3	{Z:C=O, Y:C=C, X:C=}	{Y(Br)Z, YCOZ, CXZ}	17.0

Figure 4. An illustration of an example run of the FGCompress algorithm. At iteration 2, adding the best substring C=O does not further compress the message. The algorithm is then terminated and the codebook at step 2 is taken as final. Length values are invented for illustrative purposes.

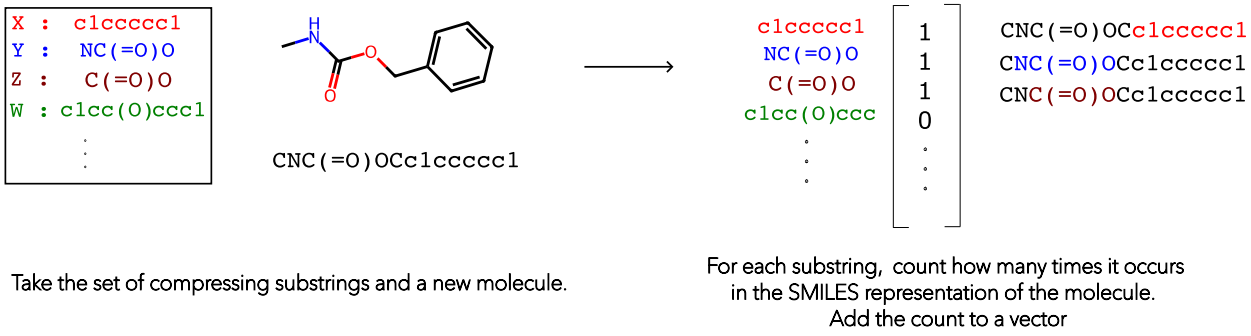


Figure 5. Conversion of a molecule to count-based fingerprint given a codebook.

Experiment 2: Assessing Learning Benefit of Compressing Groups

To assess claim 2, we independently run FGCompress to completion on the molecules in 24 bioactivity prediction data sets to extract 24 compressing substructure sets. For each data set, the corresponding set of compressing substructures is then used by FGFingerprinter to convert molecules into a count-based fingerprint representation. Representing a molecule as a count based fingerprint is a lossy compression, as it does not contain connectivity information between substructures.

We train ridge regression models to predict IC_{50} values using the FGFingerprinter derived fingerprints. We compare the predictive testing error of the learned regressors against alternative molecular representations: Morgan fingerprints, MACCS fingerprints and neural molecular embeddings from MolFormer-XL.[9,11,17]

A pictorial representation of the experimental procedure is shown in Figure 3. To ensure reproducibility, we repeat each experiment 5 times and take the average accuracy. We conduct statistical tests at $p < 0.05$ to ensure significance, with Benjamini Hochberg corrections.

We describe the introduced algorithms in more depth in the following section.

■ ALGORITHMS

To test our claims we require algorithms which (i) identify substructures which compress chemical data sets and (ii) use these compressing substructures in a chemical representation. Our contribution in this section is the introduction of two such algorithms: FGCompress and FGFingerprinter.

FGCompress

We now detail our substructure discovery algorithm FGCompress. Our algorithm is a greedy search, choosing substructures which best compress the data set at each iteration. The algorithm terminates when no substructure can further compress the data set. The algorithm requires a set of molecules represented as SMILES strings as input. A description of the FGCompress procedure follows.

1. Enumerate all substrings up to a user-specified maximum length in the data set.
2. Filter out substrings that do not correspond to valid chemical substructures. The remaining substrings are termed valid substrings.
3. For each valid substring, compute the total message length obtained if:
   (a) All instances of the substring in the data set are replaced by a single new symbol.
   (b) The new symbol−substring pair is added to the codebook.
   (c) The updated codebook and data set are transmitted as a message.
4. Select the substring that yields the shortest total message length (maximally compresses the data set).
5. If the selected substring reduces the total message length, add it to the codebook and return to step 1.
6. Otherwise, if no new substring can further reduce the message length, terminate and return the current codebook.

If a newly added substring (step 4) contains an existing codebook entry, the count of that entry is reduced accordingly. If the count of an entry reaches zero, the substring is removed from the codebook. A diagrammatic representation of a hypothetical FGCompress run is shown in Figure 4 below.

At each step, the message length is calculated as the number of bits required to transmit the codebook and data set, modeling molecules as draws from a multinomial distribution specified by the codebook. We note that our algorithm considers substrings which may themselves contain previously compressed substrings. For example, in step 3 of Figure 4 the algorithm assigns the candidate codeword Z to the substring XO, which is equivalent to C=O as X:C =. Consequently, even if the algorithm only enumerates substrings up to a small user-specified maximum length, larger substrings can still be extracted by the end of the algorithm. We discuss the effect of the user-specified maximum substring length (step 1 of the FGCompress procedure) with our experimental results.

We describe specific calculation of the message length in the next section.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

FGFingerprinter

This algorithm takes as input the set of compressing substructures from FGCompress and generates a molecular fingerprint. The fingerprint is a fixed length integer vector of counts. Each index of the vector corresponds to a substructure which was discovered during the FGCompress search. To generate a molecular fingerprint for a molecule, FGFingerprinter counts the number of times each substructure occurs in the molecule and sets the corresponding vector index to this count value. We refer to this fingerprint as the MML87 fingerprint, as the FGCompress search procedure relies on the MML87 approximation to calculating the MML codelength.[18] The MML87 fingerprint is a lossy chemical representation as it does not retain information pertaining to connections between the substructures. A pictorial representation of using a codebook to construct a fingerprint is shown in Figure 5. The larger the set of compressing substructures, the larger the molecular fingerprint vector.

DEFINITIONS

Definition 1 (**SMILES symbol**). A SMILES symbol is an element of the SMILES alphabet as defined in Weininger.[8] We denote the SMILES alphabet as Σ.

Definition 2 (**Valid Substring**). Several concatenations of SMILES symbols do not correspond to valid substructures. To ensure our codebook contains as few invalid substructures as possible, we filter the set of all substrings.

The set of all substrings Σ^* is the set of all finite strings which can be constructed from Σ. The set of valid substrings $S = F(\Sigma^*) \subset \Sigma^*$ is the set of all strings that satisfy the filter function F: $\Sigma^* \rightarrow \Sigma^*$. A valid substring is an element of the set of valid substrings $s \in S$. In this report, F filters Σ^* for those substrings using the following rules.

- If a substring contains a bracket, then it must also contain the corresponding matching bracket
- If the substring contains a bond character (=,#,-,/,\) or stereogenic center character (@), then it must contain at least one atom connected to this character.
- If the substring contains a number, then it must also contain a second instance of that number (c1cccc not permitted).
- The character is not permitted in any substring.

Informally, a SMILES substring is any contiguous list of SMILES characters that is present in a molecule. For example, given the molecule `C(=O)` there are 15 possible substrings.

```
C   C(  C(=   C(=O  C(=O)
(   (=  (=O   (=O)
=   =O  =O)
O   O)
)
```

Of which only 8 are valid substrings.

```
C                  C(=O)
(              (=O)
=   =O
O
)
```

Definition 3 (**Codebook**). A codebook is a set of pairs $\{(s_1,p_1),\ldots(s_n,p_n)$, where $s_i \in S$ is a valid substring and $p_i \in [0, 1]$ is a probability value. The probabilities are constrained such that $\sum_{i=1}^{n} p_i = 1$. The codebook is interpreted as a multinomial probability distribution: substring s_i has probability p_i of being drawn randomly, with replacement.

Definition 4 (**Optimal Coding Scheme**). We represent the codebook and compressed data set as binary strings. To ensure our encoding is as concise as possible, we use an optimal coding scheme to represent each part of the message.[19,20] An optimal coding scheme is designed to minimize the average length of an encoded message. The scheme assigns shorter binary strings to more probable events and long strings to less probable ones. Formally, an event x with probability $P(x)$ is represented as a binary string of length $-\log_2 P(x)$. Hereon, all log terms are assumed to be base 2. An optimal coding scheme ensures that the average length of the encoded data approaches the entropy of the true symbol generating distribution, the theoretical limit of compression. An optimal code is a code from an optimal coding scheme.

THE MESSAGE

We now describe the type of message that we send. The length of the message is used as our measure of compression and guides the search toward maximally compressing substrings. We use the terms codebook, valid substring and optimal coding scheme as defined in the previous section. Before the message is communicated, we assume that the sender and receiver have agreed that the probability of a SMILES symbol occurring in a codebook substring is equal to its relative frequency in the data set. Consequently, the receiver is assumed to already know the symbol frequencies in the data set. This assumption is equivalent to assuming that the receiver is aware of some preliminary analysis on the data set but does not know of a good codebook that compresses the data further. The message therefore consists of this codebook, and the data set compressed by the codebook. An example illustration of the message contents is shown in Figure 6.

Dataset:	{CX, NX, SX}
Codebook:	{X : (=O)}
Probabilities:	P(X) = 0.55 P(C) = 0.15 P(N) = 0.15 P(S) = 0.15

Figure 6. Example message contents for the original data set {C(=O),N(=O),S(=O)}.

Message Length Calculation

We now describe how we calculate the message length, the quantity we seek to minimize during our search procedure. Conventionally, a message is described in two parts: sending the codebook (and probabilities), and sending the data set. Instead, for clarity we describe the message in three parts. Part 1 communicates the substrings in the codebook. Part 2 sends both the probability values in the codebook and the substring and symbol counts in the data set given part 1. Part 3 communicates the specific sequence of symbols and substrings given parts 1 and 2. We split the message into these three parts because there exists a known expression for part 2 that we use in our implementation. We use the term "vocabulary" to refer to the minimal set of substrings and symbols required to cover the full data set. In Figure 6, the vocabulary is {X,C,N,S}.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

The length of the message communicating part 1 is determined by sending each part piecewise. As discussed, the length of communicating part 2 has an already known closed form. The length of communicating part 3 is calculated from the multinomial coefficient: the number of sequences one can make from a data set of length N, composed of i symbols with frequencies M_1, M_2, ...M_i, where all sequences are equally likely. We now describe calculation of the three parts in order.

Part 1: Communicating the Substrings. First the sender sends each substring in the codebook. The message that specifies all substrings is structured hierarchically. The sender first specifies the total number of substrings in the codebook, $|H_s|$ and then sends the data for each substring sequentially. Let T and s_i denote the codewords representing the total number of substrings in the codebook, and the i^{th} substring, respectively. Then the message format is:

`[T][s1][s2]...`

Each individual substring is also a two-part message, specifying its length followed by the constituent symbols. Let $l(s_i)$ and $\text{sym}_j(s_i)$ denote the codewords representing the length of substring i and the jth symbol of substring i. Then the individual substring message is:

$[s_i] = [l(s_i)][\text{sym}_1(s_i)][\text{sym}_2(s_i)]\ldots$

All integer values (total substring count and individual substring lengths) are communicated using an integer code. The substring symbols are encoded using their probabilities.

1. **Encoding integer values:** All integers N are encoded using the log-star universal code.[21] This is a special code which allows for specifying any integer in a concise manner. Under the log-star code, the transmission cost for the integer is $\log^*(N) \approx \log(N) + \log\log(N) + \ldots$ bits.
2. **Encoding substring symbols:** Each symbol s in substring S_i is encoded according to their preagreed probabilities of occurrence, as discussed at the beginning of this section. Under an optimal coding scheme, the cost of transmitting symbol s is $-\log_2 P(s)$ bits.

The cost of specifying a single substring $C_{\text{string}}(S_i)$ is the sum of the cost of its length and costs of all constituent symbols (eq 4).

$$C_{\text{string}}(S_i) = \log^*|S_i| - \sum_{s\in\Sigma} [\text{count}(s, S_i) \times \log P(s)] \tag{4}$$

where $\text{count}(s, S_i)$ is the number of times symbol s appears in substring S_i.

The total cost of all of the substrings is the cost of specifying the number of total substrings, plus the sum of the costs of each substring (eq 5).

$$P_1 = \log^*|H_s| + \sum_{i=1}^{|H_s|} C_{\text{string}}(S_i) \tag{5}$$

For example, consider sending the example substring S_{ex} = CCCN where $P(C) = 0.5$ and $P(N) = 0.25$.

- The cost to transmit the length ($|S_{\text{ex}}| = 4$) is $\log^*(4) \approx 2$ bits.
- The cost of transmitting the three C symbols is $3 \times (-\log\ 0.5) = 3$ bits.
- The cost of transmitting the single N symbol is $1 \times (-\log\ 0.25) = 2$ bits.

The total length $C(S_{\text{ex}})$ is then $2 + 3 + 2 = 7$ bits.

Part 2: Sending Vocabulary Probabilities and Counts. The part 2 message communicates two pieces of information (i) the underlying probabilities assigned to each item in the vocabulary and (ii) the specific counts of those items observed in the data set. A known approximation to the optimal length of this combined message is given by the MML87 formula for the multinomial distribution. We state this here and refer the reader to Wallace[6] and Wallace and Freeman[18] for a full derivation. The length of this part of the message, denoted P_2 is

$$P_2 = \log_2 \frac{\Gamma(N+M)}{\Gamma(M)} + \sum_m \log_2 \frac{1}{\Gamma(s_m+1)} + \frac{1}{2} \log((M-1)\pi) - 0.4 \tag{6}$$

where N is the total number of substrings and symbols in the data set, M is the number of distinct substring and symbols, $\Gamma(.)$ is the gamma function, s_m is the count of the m^{th} symbol and π is the standard mathematical constant.

Part 3: Specifying the Sequence of the Symbols. While the part 2 submessage specifies the number of symbols and their probabilities, it does not communicate their ordering. The final part of the message encodes the specific sequence of the N total substrings and symbols in the encoded data set. The number of unique sequences that can be formed is given by the multinomial coefficient. Assuming all valid orderings are equally likely, then the probability of a single sequence is simply $\frac{1}{\text{number of unique sequences}}$. Under an optimal code, the cost of specifying the specific sequence the negative logarithm of the probability of a specific sequence, denoted P_3. We assume that the receiver knows the number of individual symbols in each SMILES string, so that they can split the received string into the original molecules.

$$P_3 = \log_2 \frac{N!}{s_1!s_2!\cdots s_M!} \tag{7}$$

Full Message Length

The full message length is the sum of the lengths of the three parts

$$M = P_1 + P_2 + P_3 \tag{8}$$

We use eq 8 as our cost function, and implement a search procedure to minimize this value.

EXPERIMENTAL QUESTIONS

We conducted experiments to verify our claims. We first try to answer the question:

Q1 Do functional groups emerge when compressing a large corpus of biologically relevant molecules? We answer **Q1** by running the FGCompress search procedure on all molecules in the ChEMBL data set for 500 iterations, and caching the discovered substructures. We terminate the algorithm after 500 iterations due to computational constraints. While early termination may prevent the discovery of additional patterns, FGCompress extracts substructures in order of importance, so the top 500 substructures are expected to strongly represent the data set. Unfortunately, there is no canonical list of functional groups to which we could compute a similarity score. Instead, we manually assess the discovered substructures, and note exact and partial matches to human, named functional groups.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

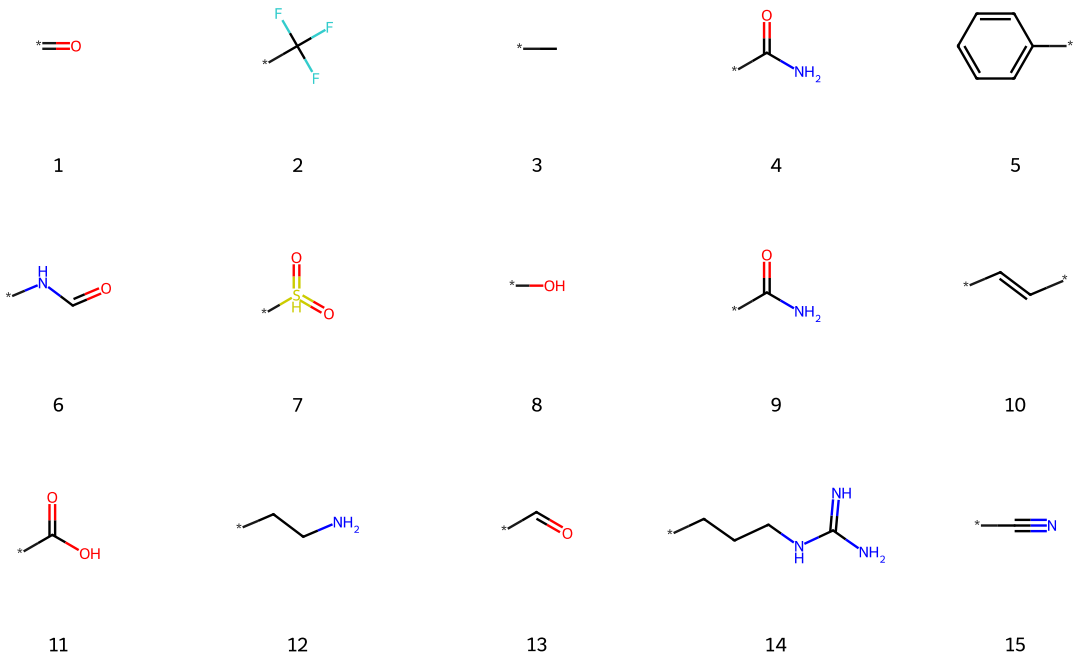


Figure 7. Top 15 substructures derived by the algorithm. Substructures are numbered in order of relative importance.

To test our claim that the discovered substructures aid learning, we try to answer the question:

Q2 Are the substructures extracted from compressing the data set useful for learning? We answer **Q2** by generating a unique MML87 fingerprint for each of 24 bioactivity prediction data sets using the FGFingerprinter algorithm. We compare the predictive performance of four fingerprint representations: our MML87 fingerprint, the 166 bit MACCS fingerprint, a Morgan fingerprint (radius 2) hashed to the same length as the corresponding MML87 fingerprint for that specific data set and a continuous neural representation (MoLFormer-XL).[9,11,17] We train a cross-validated ridge regression model for each data set.[22,23] We compare the mean squared error of the models on a test set using each fingerprint.

Data and Experimental Choices

We use canonical SMILES representations as input for all FGCompress runs. We use the following data sets for each experiment.

Experiment 1 (Q1). Data We ran the FGCompress algorithm on the ChEMBL data set (release 36), which contains 2,878,135 bioactive molecules.[24] We chose the ChEMBL data set as the data is high quality and manually curated. **FGCompress Parameters** We used a maximum substring enumeration length of 8 and a uniform prior distribution over all substring probabilities via a generalized beta function.

Experiment 2 (Q2). Data We ran the FGCompress algorithm on the 24 bioactivity prediction data sets, reported in Cortés-Ciriano and Bender.[25] These data sets contain IC_{50} data for 24 diverse protein targets and receptors from the ChEMBL database. In their study, the authors state that they only retained IC_{50} values for small molecules for which.

- The activity unit was equal to "nM"
- The activity relationship was equal to " = "
- The target type was equal to "SINGLE PROTEIN"
- The organism was equal to *Homo sapiens*.

Models. We choose ridge regression models over more expressive models such as random forests, gradient-boosted trees, or neural networks because our goal is to directly evaluate the quality of the substructures. Random forests and gradient-boosted trees implement learned feature transformations on top of the input features, while neural networks construct internal representations in their hidden layers.[26−28] In contrast, ridge regression provides a purely linear mapping from features to predictions, making it ideally suited for assessing raw representation quality.

Hardware and Implementation. We ran all FGCompress experiments on a server with 36 CPU cores and 125 GB RAM. We ran the FGFingerprinter experiments on an M2MacBook Air (8 CPU cores, 16 GB RAM). All experiments were repeated 5 times, using random 75/25 train/test splits, and we report the mean and standard error in each case. Statistical significance was assessed using the Wilcoxon signed-rank test with Benjamini-Hochberg multiple testing corrections at $\alpha = 0.05$.[29,30] Significance tests were conducted between each fingerprint representation over all data sets to satisfy the independence assumptions of the tests. Ridge regression models were implemented using scikit-learn.[31] The ridge α hyperparameter was optimized via leave-one-out cross-validation on the training set, selecting from {0.001, 0.01, 0.1, 1}.

RESULTS AND DISCUSSION

We now present the empirical results to address our experimental questions **Q1** and **Q2**.

Q1: Functional Group Verification

The full list of substructures discovered by FGCompress may be found in the appendix. In all figures, substructures are listed in order of the iteration at which they were derived. This reflects the explanatory power of the substructure with respect to the data set. The resulting substructures compress the data set by 30% relative to the data set encoded in terms of SMILES symbols only.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

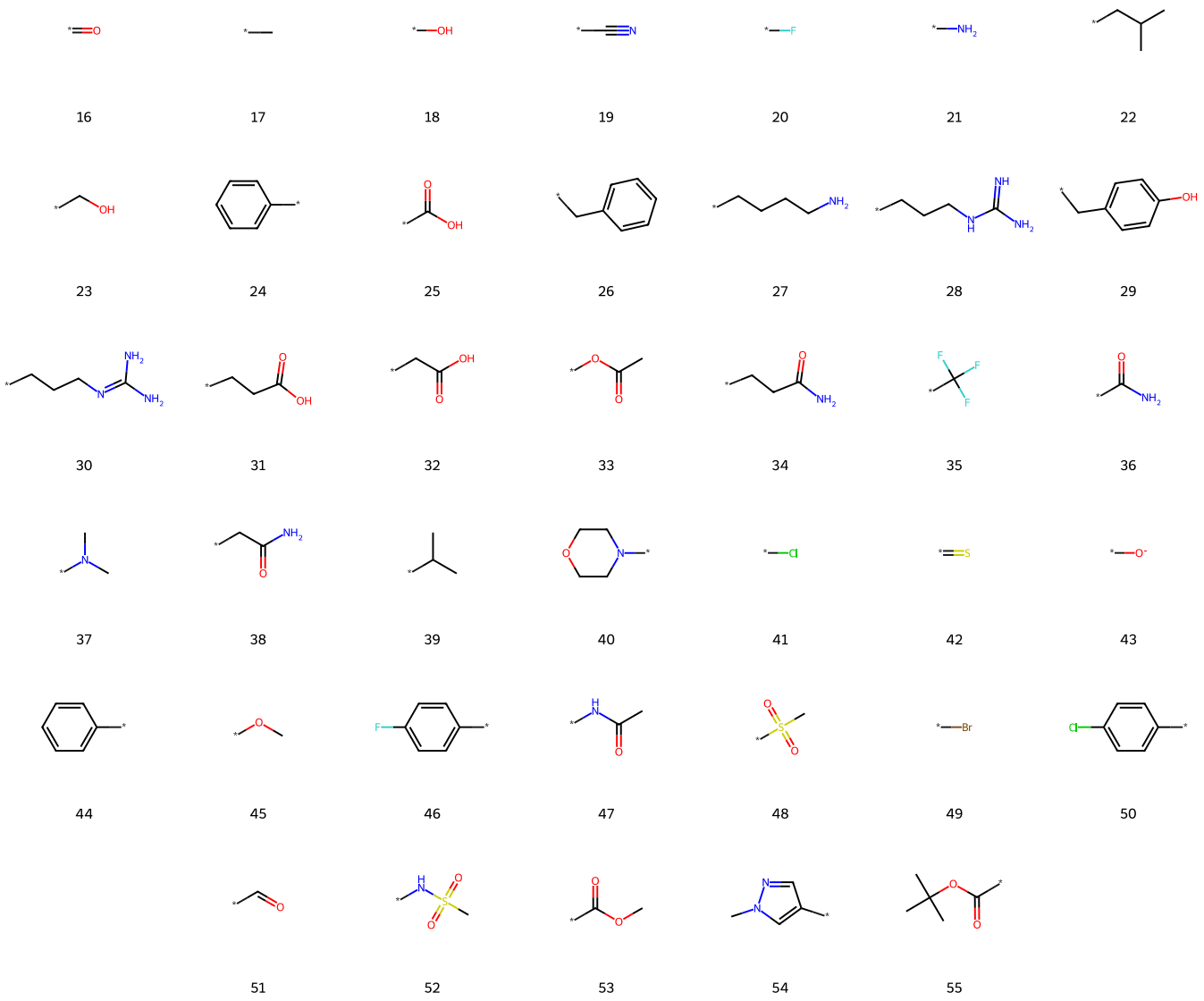


Figure 8. Top 40 substructures extracted as complete branches discovered by FGCompress. Substructures are numbered in order of relative importance.

Of the 500 substructures, 494 were successfully converted to SMARTS patterns using RDKit. A substructure was considered convertible if it is already a valid SMARTS pattern or becomes valid by adding the wildcard "*" at the beginning, end, or both. Consequently, we conclude that use of the SMILES representation in our procedure did not significantly affect the interpretability of the extracted substrings.

After filtering for substrings that can be converted to valid SMARTS patterns, we further remove substructures that are identical up to numerical relabeling. For example, the substring `c2ccccc2` would be removed if `c1ccccc1` is already present. After filtering, FGCompress returned 304 substructures. We next analyze the filtered list.

The top 15 substructures found by the algorithm are shown in Figure 7 below. All of these substructures are well-known functional groups. For example, the top five groups include the carbonyl, trifluoromethyl, methyl, amide, benzene functional groups. As ChEMBL is a bioactivity data set we expect functional groups that are strongly represented in the biological literature, such as those which are derivatives of amino acids (substructures 4, 6, 9, 11, 12, 14). Substructures 4 and 9 are identical as they are two SMILES representations of the amide group `C(=O)N` and `NC(=O)` that were not caught by our filtering procedure. The duplication of the amide group is a direct artifact of the SMILES representation.

We now analyze subsets of the discovered groups: branches, small and large rings.

The top 40 branch substructures found by the algorithm are shown and numbered in Figure 8. Nearly all discovered branches are standard human functional groups. For example, the branches include carbonyl (16), methyl (17), alcohol (18) nitrile (19), fluoro (20), and amine (21) functional groups. As noted previously, due to the nature of the ChEMBL data set the number of peptide related functional groups is high. Specifically, there is an abundance of amino acid side chains and derivatives (substructures 22, 23, 26, 27, 28, 29, 30, 31, 32, 34, 38, 39). Other standard functional groups include benzyl (24, 44), trifluoromethyl (35), tertiary amines (37), morpholine (40), aromatic halides (46, 50), and the Boc protecting group (55). Interestingly, all halide substituents in the top 40 branches are in the para configuration. We were unaware a priori of the relative prevalence of ortho, meta, or para configurations.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

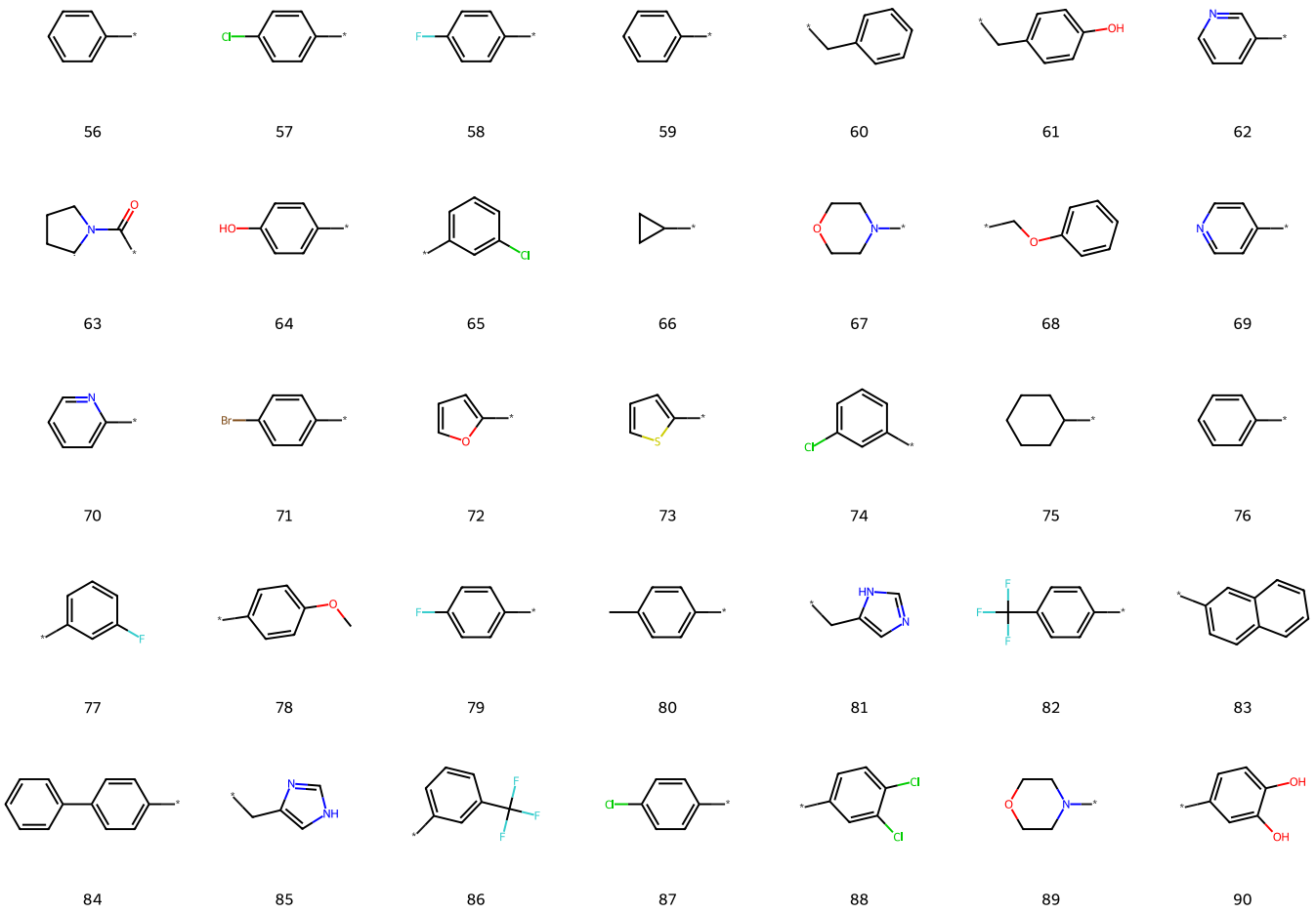


Figure 9. Substructures containing complete rings expressed in less than 20 symbols.

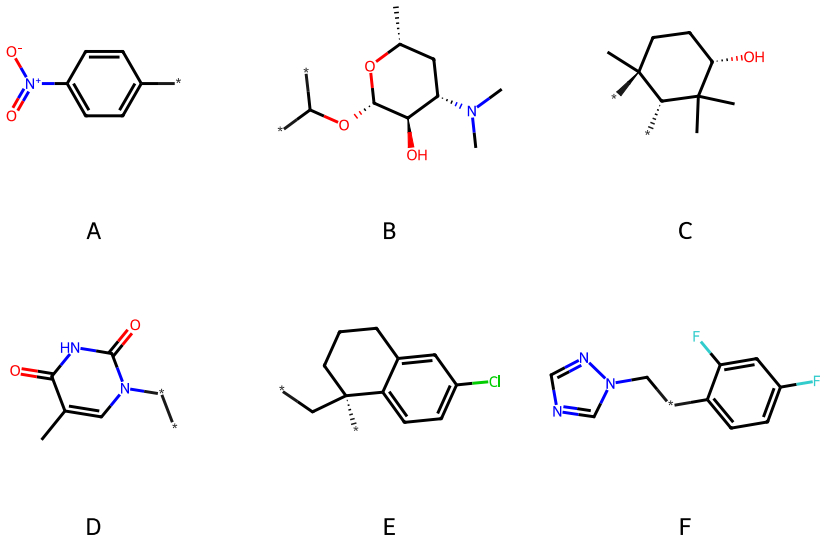


Figure 10. Large rings expressed in more than 19 symbols, amide containing ring fragments removed.

The top 35 small ring containing substructures that may be expressed in fewer than 20 SMILES symbols are presented in Figure 9 below.

Most presented rings in Figure 9 are generally known substructures. As expected, the functional group benzene is ranked as the most important ring substructure by FGCompress. While many rings are known functional groups, such as morpholine (substructure 67), pyridine (substructure 70) furan (substructure 72), thiazole (substructure 73), napthalene (substructure 83), a large number may more fittingly be termed "substructures," or combinations of primitive functional groups, such as substructures 68, 74, and 90.

Aside from the substituted benzene derivatives, the rings similarly contain several amino acid side chains (substructures 60 and 61). The stereochemistry of the disubstituted aromatic groups (substructures 88 and 90) appears reasonable, as the

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

Table 1. Mean Squared Error of the Ridge Regression Models Across Datasets and Fingerprint Representations. The Standard Error from the 5 Trials is Reported

data set	MML87	MACCS	molformer	morgan
B-raf	0.61 ± 0.00	0.72 ± 0.00	0.59 ± 0.01	**0.46 ± 0.01**
ephrin	0.73 ± 0.01	0.78 ± 0.01	0.74 ± 0.02	**0.68 ± 0.01**
caspase	**0.47 ± 0.01**	0.65 ± 0.01	0.62 ± 0.01	0.53 ± 0.01
monoamine	0.75 ± 0.06	0.66 ± 0.01	0.69 ± 0.02	**0.59 ± 0.01**
vanilloid	**0.60 ± 0.01**	0.68 ± 0.01	0.72 ± 0.01	0.66 ± 0.02
acetylcholinesterase	**0.80 ± 0.01**	1.06 ± 0.01	0.93 ± 0.01	0.83 ± 0.01
COX-1	**0.64 ± 0.01**	0.70 ± 0.01	0.66 ± 0.01	0.82 ± 0.01
ABL1	**0.80 ± 0.01**	0.92 ± 0.02	1.00 ± 0.02	0.89 ± 0.02
A2a	**0.68 ± 0.02**	0.82 ± 0.02	0.74 ± 0.04	0.82 ± 0.03
COX-2	**0.80 ± 0.01**	0.92 ± 0.01	0.80 ± 0.01	0.92 ± 0.02
dihydrofolate	**0.88 ± 0.03**	1.01 ± 0.03	1.13 ± 0.04	1.04 ± 0.02
opioid	**0.73 ± 0.01**	0.87 ± 0.01	0.83 ± 0.01	0.77 ± 0.02
glycogen	0.84 ± 0.04	0.96 ± 0.01	0.93 ± 0.01	**0.77 ± 0.01**
erbB1	**0.70 ± 0.00**	1.02 ± 0.01	0.76 ± 0.00	0.72 ± 0.00
LCK	**0.86 ± 0.01**	1.06 ± 0.01	0.92 ± 0.01	0.93 ± 0.01
aurora-A	**0.82 ± 0.01**	1.20 ± 0.01	0.92 ± 0.01	0.87 ± 0.01
glucocorticoid	**0.40 ± 0.00**	0.52 ± 0.01	0.50 ± 0.01	0.48 ± 0.01
cannabinoid	**0.65 ± 0.02**	0.94 ± 0.02	0.78 ± 0.01	0.81 ± 0.02
carbonic	0.57 ± 0.03	**0.51 ± 0.01**	0.60 ± 0.01	0.70 ± 0.01
JAK2	0.66 ± 0.01	0.96 ± 0.01	0.72 ± 0.01	**0.62 ± 0.01**
HERG	**0.43 ± 0.01**	0.63 ± 0.01	0.50 ± 0.00	0.53 ± 0.00
coagulation	**0.85 ± 0.01**	1.14 ± 0.02	1.13 ± 0.03	1.02 ± 0.01
estrogen	**0.49 ± 0.00**	0.57 ± 0.01	0.67 ± 0.01	0.65 ± 0.00
dopamine	0.89 ± 0.03	**0.73 ± 0.02**	0.84 ± 0.01	0.85 ± 0.02

substitutions are positioned as far as possible from the connection point, thereby minimizing steric hindrance.

Larger rings that may be expressed in more than 19 symbols are shown in Figure 10. The substructures of Figure 10 are generally not themselves standard functional groups, but have a more specific biological function. We omit those substructures that are simple fragments of peptides. We also omit the discovered substring `C[C@@H]1OP(=O)(O)OC[C@H]1O[C@@H]`, which, although technically valid, misrepresents the substructure in the original molecules. The original molecules contain two rings, both labeled with 1 to indicate each ring opening and closure. In the extracted substring, the two 1s are misinterpreted as belonging a single ring. We do not observe this specific issue in any other substring. Substructure *A* is a para-nitrobenzene which we do not find surprising. It is common knowledge that nitrobenzenes are ubiquitous in bioactive molecules. Substructure *B* forms part of a desosamine fragment, a central part of the pharmacophore of macrolide antibiotics.[32] Substructure *C* is found in a number of compounds of the triterpene class such as lanosterol. Lanosterol is the precursor to cholesterol, the compound from which all animal and fungal steroids are derived and plays a role in maintaining lens health.[33,34] A search on the ChEMBL database reveals that substructure *C* is present in 169 SMILES strings, and 85 distinct literature sources. These sources refer to investigations into compounds with promise in anticancer, neuroprotective, hepatoprotective, cataract therapy, and treatments of Chagas disease.[35−40] Substructure *C* therefore features in a variety of biochemical explanations. Substructure *D* occurs in a number of antiviral therapies, including Zidovudine, Telbivudine, and Trifluridine.[41−43] Substructure *E* features in the MCL-1 inhibitor AMG-176.[44] Substructure *F* mainly occurs in a number of antifungal medications such as Fluconazole and Miconazole.[45,46] In addition to those substructures shown in Figure 10, the amino acid sequence Leu-Arg-Glu-Phe-Tyr-Gly was also discovered. A search in the ChEMBL database reveals the sequence in peptides which bind Tissue Factor Pathway Inhibitor (TFPI).[47]

From our analysis we conclude that the answer to **Q1** is yes, that functional groups emerge from data compression of the ChEMBL data set. We additionally observe that FGCompress discovers data set specific functional groups, which correspond to more specific molecular behavior.

Q2: Learning Benefit

Over the 24 data sets, we find that the Ridge Regression models trained using the MML87 fingerprint representation significantly outperform those trained using the MACCS fingerprint, Morgan fingerprint, and Neural embeddings (MolFormer-XL) representations at $p < 0.05$. The performance increase of the MML87 fingerprint was confirmed by Benjamini−Hochberg corrected significance tests at an overall $p < 0.05$. Specifically, we validate the claim that the MML87 fingerprint results in improved interpolative model predictions. The MML87 fingerprint outperforms both the MACCS and MolFormer-XL representations in 21 out of 24 cases and outperforms the equivalently sized Morgan fingerprint in 18 out of 24 cases. The results for each data set are summarized in Table 1.

These results suggest that the answer to **Q2** is yes, substructures that compress the data set are useful for learning and result in improved performance over the MACCS fingerprint on linear regression tasks. Note that we do not claim that the MML87 fingerprint is preferable for all learning models, just simple regression. For example, the random forest (RF) model is successful in QSAR tasks and insensitive to large numbers of low information features.[26,48] Representations that enumerate vast numbers of substructures such as the Morgan fingerprint are instead likely to be beneficial for the RF model.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

	Run 1		Run 2	
Step	Codebook	Compressed Data	Codebook	Compressed Data
0	{}	{CCN(=O), CN(=O)CC}	{}	{CCN(=O), CN(=O)CC}
1	{X:CC}	{XN(=O), CN(=O)X}	{A:CN(=O)}	{CA, ACC}
2	{X:CC, Y:=O}	{XN(Y), CN(Y)X}	-	-

Figure 11. Effect of maximum substring length on discovered substructures.

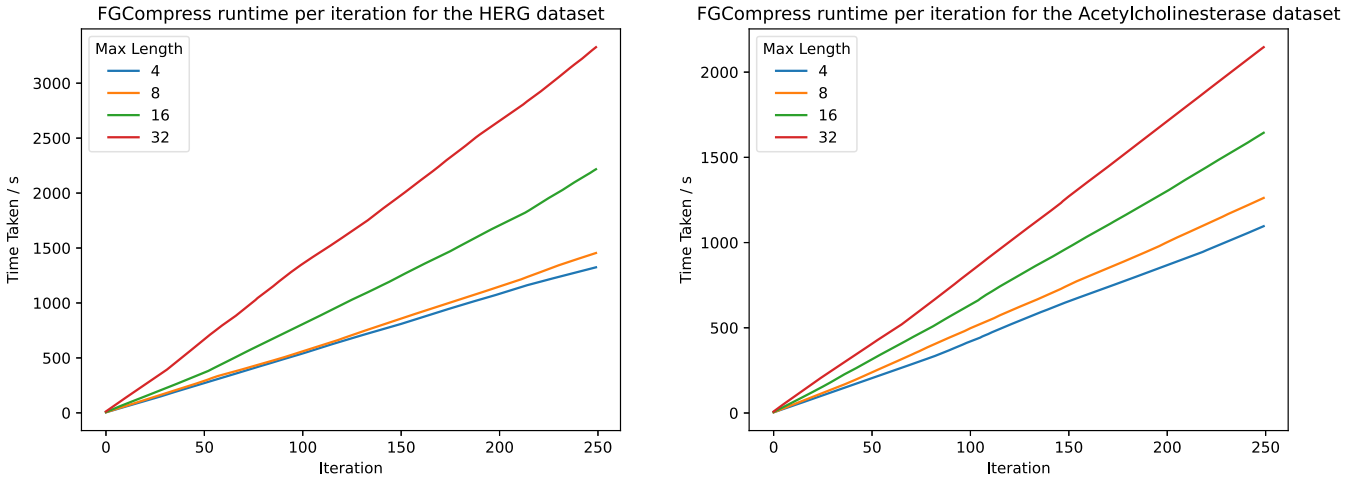


Figure 12. Runtime of the FGCompress over 250 iterations on the HERG (5207 molecules) and Acetylcholinesterase (3159 molecules) data sets. Larger substring lengths result in exponentially longer runtimes. Data sets with fewer molecules result in shorter runtimes.

Effect of Substring Length

We now describe the effect of the user-specified maximum substring length on the FGCompress algorithm. The substring length affects (i) the specific substructures extracted and (ii) the algorithm runtime. We describe each in turn.

Effect on Extracted Substrings

There may be substring codebooks that are overall better for compression, but cannot not be found by smaller maximum substring lengths. As an example, consider compressing a data set consisting of only two molecules, represented as the SMILES strings {CCN(=O), CN(=O)CC}. Consider two FGCompress runs with differing maximum substring lengths: run 1 has a maximum length of 2, while run 2 has a maximum length of 5. The two runs are shown in Figure 11 below.

Run 2 terminates at iteration 1: It finds the maximally compressing substring CN(=O) and replaces it with the symbol *A*. Run 1 terminates at iteration 2: it finds the substrings CC and =O but fails to find the more compressing substring CN(=O). The difference in codebooks occurs, because to add any expression in brackets to the codebook, such as (*Y*) (where *Y* denotes =O) one requires consideration of substructures of size greater than 2. There are two substrings of (*Y*) of length 2. The substrings are ((*Y* or *Y*)), neither of which are valid. Larger maximum substring lengths may be required to extract other meaningful patterns, such as those with nested brackets. Larger maximum substring lengths prevents the algorithm from missing large, compressing substrings.

Effect on Runtime

The maximum substring length, however, also affects the runtime of the algorithm. The time complexity of enumerating all substrings up to length *N* increases exponentially with *N*. We plot the time taken per iteration for increasing maximum substring lengths 4, 8, 16, and 32 on the HERG and Acetylcholinesterase data sets in Figure 12 below. We plot the first 250 iterations in each case. The full codebook contains 3398 substructures. The smaller Acetylcholinesterase data set results in shorter runtimes, as there are fewer substrings to enumerate. We run the experiment on an M2MacBook Air (8 CPU cores, 16 GB RAM) using all CPU cores per experiment.

CONCLUSIONS

We showed that compressing chemistry results in the automatic identification of the standard set of human-identified functional groups, alongside several data set specific substructures. This work therefore provides computational validation that partitioning of molecules into functional groups is indeed a good general description of chemistry. We highlighted several interesting patterns that do not form part of the standard chemical functional group set and showed that several compressing functional groups correspond to specific biological function. Empirically, we have shown that fingerprints generated from the discovered substructures aid learning, and that ridge regression models trained on the MML87 fingerprints significantly outperform the MACCS and Morgan representations on 24 IC_{50} prediction tasks. The improved performance of the MML87 fingerprint over MACCS and Morgan is in spite of the reduced expressiveness of the MML87 fingerprint and comparable fewer substructures compared with the Morgan algorithm. Additionally, chemical fingerprints are often used for similarity scoring. Specifically, techniques like Taylor–Butina clustering leverage a chemical fingerprint to cluster molecules by similarity.[49] Future work could also investigate the efficacy of the MML87 fingerprint in similarity scoring, comparing it to such techniques, and other common methods such as using Bemis–Murcko scaffolds.[50]

LIMITATIONS

For computational tractability, we conducted our search procedure over SMILES strings. Consequently, our search space was restricted to only those substructures which may be

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574

represented as SMILES substrings. Unfortunately, SMILES substrings are a subset of the complete set of substructures within a molecule. Moreover, the same group in different compounds may be represented with different SMILES strings. We attempted to minimize this multiple representation issue by using canonical SMILES representations. Despite this issue, functional groups are still extracted by the algorithm because they occur at sufficiently high frequencies. Searching for subgraphs in graph molecule representations would be complete and remove representational redundancy, but computationally very expensive. Future work should explore a trade-off between the two such as lookahead searches, or hybrid representations. One could adapt recent advances in inductive logic programming (ILP), such as the use of constraint solvers to compress data sets to remedy this.[51−53] We demonstrated that compressive substructures enhance predictive performance in learning tasks. However, certain features may be highly informative for explaining specific molecular properties even if they contribute little to overall data set compression. Moreover, widely used QSAR models such as random forests are relatively insensitive to the inclusion of many low-information features and often perform best with representations that enumerate large numbers of substructures, such as the Morgan fingerprint.[26] The substructures identified by FGCompress therefore remain valuable and could serve as complementary features to enumerative fingerprints, particularly given that FGCompress imposes no restriction on maximum substructure size, unlike many conventional fingerprinting methods. The loss of connectivity information when constructing the fingerprint described in this report is a limitation of the FGFingerprinter method. We argue that this limitation is not necessarily problematic, as sophisticated machine learning methods, such as logic programming approaches, can easily incorporate connectivity information without much overhead. Additionally, the method could be used as an alternative tokenization scheme for neural network based modeling. When tokenizing a molecular data set, one simply may directly use the compressed data set representation as input, which preserves connectivity information. Further, the strong empirical results from the FGFingerprinter experiments lend evidence to the efficacy of the fingerprint, even when omitting the connectivity information.

■ ASSOCIATED CONTENT

Data Availability Statement

All code to run the FGCompress algorithm is public and may be found at the github link: https://github.com/bars20/compressing-chemistry.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.5c02917.

DiscoveredGroups.txt: An exhaustive listing of the top 500 functional groups analyzed in this report. All groups are listed in the original SMILES string form (TXT)

■ AUTHOR INFORMATION

Corresponding Author

Ruben Sharma − *Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, U.K.;* orcid.org/0009-0004-0923-6383; Email: bars2@cam.ac.uk

Author

Ross D. King − *Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, U.K.; Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg S-412 96, Sweden*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.5c02917

Author Contributions

R.S. formalized the idea, designed the algorithms, and ran the experiments, supported by R.D.K. R.D.K. conceived the initial idea. Both authors discussed the results and edited the paper.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

R.S. was supported by the EPSRC Centre of Doctoral Training in Automated Chemical Synthesis Enabled by Digital Molecular Technologies (Syntech, EP/S005722/1). R.D.K. was supported by the EPSRC (EP/R022925/2, EP/W004801/1, and EP/X032418/1), and by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Alice Wallenberg Foundation.

■ ABBREVIATIONS

MML	minimum message length
SMILES	simplified molecular input line entry system
MACCS	molecular ACCess system

■ REFERENCES

(1) Clayden, J.; Greeves, N.; Warren, S. *Organic Chemistry*; Oxford University Press, 2012.

(2) Bush, K.; Bradford, P. A. β-Lactams and β-lactamase inhibitors: an overview. *Cold Spring Harb. Perspect. Med.* **2016**, *6*, a025247.

(3) Sanabria, E.; Cuenca, R. E.; Esteso, M. Á.; Maldonado, M. Benzodiazepines: their use either as essential medicines or as toxics substances. *Toxics* **2021**, *9*, 25.

(4) Elliott, G. A.; Purmalis, A.; Vandermeer, D. A.; Denlinger, R. H. The propionic acids. Gastrointestinal toxicity in various species. *Toxicol. Pathol.* **1988**, *16*, 245−250.

(5) Solomonoff, R. J. A formal theory of inductive inference. Part I. *Inf. Control* **1964**, *7*, 1−22.

(6) Wallace, C. S. *Statistical and Inductive Inference by Minimum Message Length*; Springer, 2005.

(7) Allison, L. *Coding Ockham's Razor*; Springer, 2018.

(8) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.

(9) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1273−1280.

(10) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2215−2223.

(11) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(12) Evans, S. C.; Bush, S. F. *Symbol Compression Ratio for String Compression and Estimation of Kolmogorov Complexity*; Manuscript submitted for publication, 2001.

(13) Evans, S.; Eiland, E.; Markham, S.; Impson, J.; Laczo, A. In *MILCOM 2007*, 2007; pp 1−7.

(14) Evans, S. C.; Kourtidis, A.; Markham, T. S.; Miller, J.; Conklin, D. S.; Torres, A. S. MicroRNA target detection and analysis for genes related to breast cancer using MDLcompress. *EURASIP J. Bioinformatics Syst. Biol.* **2007**, *2007*, 43670.
(15) Erten, C.; Algahtani, E.; Fairlamb, I.; Garcia-Padilla, E.; Lynam, J.; Manandhar, S.; Slattery, J.; Kazakov, D. In *ILP*, 2019.
(16) Ertl, P. An algorithm to identify functional groups in organic molecules. *J. Chem. Inf.* **2017**, *9*, 36.
(17) Ross, J.; Belgodere, B.; Chenthamarakshan, V.; Padhi, I.; Mroueh, Y.; Das, P. Large-scale chemical language representations capture molecular structure and properties. *Nat. Mach. Intell.* **2022**, *4*, 1256−1264.
(18) Wallace, C. S.; Freeman, P. R. Estimation and inference by compact coding. *J. R. Stat. Soc., Ser. B (Methodol.)* **1987**, *49*, 240−252.
(19) Shannon, C. E. A mathematical theory of communication. *Bell Syst. Technol. J.* **1948**, *27*, 379−423.
(20) Cover, T. M. *Elements of Information Theory*; John Wiley & Sons, 1999.
(21) Rissanen, J. A universal prior for integers and estimation by minimum description length. *Ann. Stat.* **1983**, *11*, 416−431.
(22) Stone, M. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc., Ser. B (Methodol.)* **1974**, *36*, 111−133.
(23) Hoerl, A. E.; Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55−67.
(24) 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.
(25) Cortés-Ciriano, I.; Bender, A. Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. *J. Chem. Inf. Model.* **2019**, *59*, 1269−1281.
(26) Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5−32.
(27) He, X.; Pan, J.; Jin, O.; Xu, T.; Liu, B.; Xu, T.; Shi, Y.; Atallah, A.; Herbrich, R.; Bowers, S., et al. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 2014; pp 1−9.
(28) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016; . http://www.deeplearningbook.org (accessed 9th October 2025).
(29) Wilcoxon, F. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: 1992; pp 196−202.
(30) Benjamini, Y.; Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc., Ser. B (Methodol.)* **1995**, *57*, 289−300.
(31) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(32) Burgie, E. S.; Holden, H. M. Molecular architecture of DesI: a key enzyme in the biosynthesis of desosamine. *Biochemistry* **2007**, *46*, 8999−9006.
(33) Zhao, L.; Chen, X.-J.; Zhu, J.; Xi, Y.-B.; Yang, X.; Hu, L.-D.; Ouyang, H.; Patel, S. H.; Jin, X.; Lin, D.; et al. Lanosterol reverses protein aggregation in cataracts. *Nature* **2015**, *523*, 607−611.
(34) Nes, W. D. Biosynthesis of cholesterol and other sterols. *Chem. Rev.* **2011**, *111*, 6423−6451.
(35) Rios, J.-L.; Andujar, I.; Recio, M.-C.; Giner, R.-M. Lanostanoids from fungi: a group of potential anticancer compounds. *J. Nat. Prod.* **2012**, *75*, 2016−2044.
(36) Nakata, T.; Yamada, T.; Taji, S.; Ohishi, H.; Wada, S.-i.; Tokuda, H.; Sakuma, K.; Tanaka, R. Structure determination of inonotsuoxides A and B and in vivo anti-tumor promoting activity of inotodiol from the sclerotia of Inonotus obliquus. *Bioorg. Med. Chem.* **2007**, *15*, 257−264.
(37) Wang, Y.-Y.; Gao, Y.-X.; Gao, W.; Xu, Y.; Xu, Y.-Z.; Wang, Y.-J.; Chang, S.; Yu, L.-G.; Zhang, L.-Y.; Liao, H.; et al. Design, synthesis and biological evaluation of tricyclic diterpene derivatives as novel neuroprotective agents against ischemic brain injury. *Eur. J. Med. Chem.* **2015**, *103*, 396−408.
(38) Xu, G.-B.; Xiao, Y.-H.; Zhang, Q.-Y.; Zhou, M.; Liao, S.-G. Hepatoprotective natural triterpenoids. *Eur. J. Med. Chem.* **2018**, *145*, 691−716.
(39) Yang, X.; Chen, X.-J.; Yang, Z.; Xi, Y.-B.; Wang, L.; Wu, Y.; Yan, Y.-B.; Rao, Y. Synthesis, evaluation, and structure−activity relationship study of lanosterol derivatives to reverse mutant-Crystallin-induced protein aggregation. *J. Med. Chem.* **2018**, *61*, 8693−8706.
(40) Saidu, M. B.; Krstic, G.; Barta, A.; Hunyadi, A.; Berkecz, R.; Gallah, U. S.; Cholke, K.; Gertsch, J.; Rédei, D.; Hohmann, J. Euphane and Tirucallane Triterpenes with Trypanocidal Activity from Euphorbia desmondii. *J. Nat. Prod.* **2024**, *87*, 2281−2291.
(41) Volberding, P. A.; Lagakos, S. W.; Koch, M. A.; Pettinelli, C.; Myers, M. W.; Booth, D. K.; Balfour Jr, H. H.; Reichman, R. C.; Bartlett, J. A.; Hirsch, M. S.; et al. Zidovudine in asymptomatic human immunodeficiency virus infection: a controlled trial in persons with fewer than 500 CD4-positive cells per cubic millimeter. *N. Engl. J. Med.* **1990**, *322*, 941−949.
(42) Lai, C.-L.; Leung, N.; Teo, E.-K.; Tong, M.; Wong, F.; Hann, H.-W.; Han, S.; Poynard, T.; Myers, M.; Chao, G.; et al. A 1-year trial of telbivudine, lamivudine, and the combination in patients with hepatitis B e antigen—positive chronic hepatitis B. *Gastroenterology* **2005**, *129*, 528−536.
(43) Carmine, A.; Brogden, R.; Heel, R.; Speight, T.; Avery, G. Trifluridine: a review of its antiviral activity and therapeutic use in the topical treatment of viral eye infections. *Drugs* **1982**, *23*, 329−353.
(44) Caenepeel, S.; Brown, S. P.; Belmontes, B.; Moody, G.; Keegan, K. S.; Chui, D.; Whittington, D. A.; Huang, X.; Poppe, L.; Cheng, A. C.; et al. AMG 176, a selective MCL1 inhibitor, is effective in hematologic cancer models alone and in combination with established therapies. *Cancer Discov.* **2018**, *8*, 1582−1597.
(45) Grant, S. M.; Clissold, S. P. Fluconazole: a review of its pharmacodynamic and pharmacokinetic properties, and therapeutic potential in superficial and systemic mycoses. *Drugs* **1990**, *39*, 877−916.
(46) Piérard, G. E.; Hermanns-Lê, T.; Delvenne, P.; Piérard-Franchimont, C. Miconazole, a pharmacological barrier to skin fungal infections. *Expert Opin. Pharmacother.* **2012**, *13*, 1187−1194.
(47) Dockal, M.; Ehrlich, H.; Scheiflinger, F.; Reimer, U.; Reineke, U.; Polakowski, T.; Schneider, E. TFPI inhibitors and methods of use Patent, US20100173847A1 (United States), Filed Dec 21, 2009. Published Jul 8, 2010. Granted as US8466108B2 in 2013., 2010.
(48) Olier, I.; Sadawi, N.; Bickerton, G. R.; Vanschoren, J.; Grosan, C.; Soldatova, L.; King, R. D. Meta-QSAR: a large-scale application of meta-learning to drug design and discovery. *Mach. Learn.* **2018**, *107*, 285−311.
(49) Butina, D. Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747−750.
(50) Bemis, G. W.; Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.
(51) Sharma, R.; Dumančić, S.; King, R. D.; Cropper, A. Learning Logical Rules Using Minimum Message Length. *arXiv* **2025**, 06230.
(52) Cropper, A.; Morel, R. Learning programs by learning from failures. *Mach. Learn.* **2021**, *110*, 801−856.
(53) Hocquette, C.; Niskanen, A.; Järvisalo, M.; Cropper, A. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 10553−10561.

https://doi.org/10.1021/acs.jcim.5c02917
J. Chem. Inf. Model. 2026, 66, 3562−3574