*Molecular Informatics*

molecular informatics

RESEARCH ARTICLE OPEN ACCESS

# Predicting the Toxicity of Chemical Compounds via Hyperdimensional Computing

Fabio Cumbo[1] | Kabir Dhillon[1,2] | Jayadev Joshi[1] | Bryan Raubenolt[1] | Davide Chicco[3,4] | Sercan Aygun[5] | Daniel Blankenberg[1,6]

[1]Computational Life Sciences, Cleveland Clinic Research, Cleveland Clinic, Cleveland, Ohio, USA | [2]College of Engineering, Ohio State University, Columbus, Ohio, USA | [3]Dipartimento di Informatica Sistemistica e Comunicazione, Università di Milano-Bicocca, Milan, Italy | [4]Institute of Health Policy Management and Evaluation, University of Toronto, Toronto, Ontario, Canada | [5]School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, Louisiana, USA | [6]Department of Molecular Medicine, Cleveland Clinic Lerner College of Medicine, Case Western Reserve University, Cleveland, Ohio, USA

**Correspondence:** Daniel Blankenberg (blanked2@ccf.org)

**Received:** 3 March 2026 | **Revised:** 12 August 2026 | **Accepted:** 17 August 2026

**Keywords:** chemical compounds | cheminformatics | hyperdimensional computing | supervised learning | toxicity prediction | vector symbolic architectures

## ABSTRACT

Accurately and efficiently assessing the potential toxicity of chemical compounds is critical given their wide application across pharmaceutical, industrial, and environmental domains. Traditional toxicological evaluations, which predominantly rely on intensive in vitro and in vivo assays, are frequently slow and expensive. Here, we introduce a novel application of hyperdimensional computing (HDC), a recently developed computational paradigm inspired by the way the human brain works in encoding information, for the efficient classification of chemical compounds as either toxic or nontoxic. Our methodology employs Simplified Molecular Input Line Entry System (SMILES) representations of compounds, drawing data from the comprehensive Tox21 dataset. We delineate a pipeline wherein these chemical structures are encoded into high-dimensional binary vectors, which subsequently serve as the foundation for training and classification within the HDC framework. This approach leverages HDC's inherent advantages, including its resilience to noise, parallel processing capabilities, and efficacy in identifying intricate patterns. This work demonstrates the viability of HDC as a computationally lightweight first-pass solution for preliminary toxicity screening. This research significantly contributes to the field of cheminformatics by validating HDC's potential in chemical property prediction, thereby facilitating accelerated identification of hazardous substances and mitigating the reliance on intensive laboratory experimentation.

## 1 | Introduction

The proliferation and application of novel chemical compounds across modern society are expanding at an unprecedented rate [1]. From life-saving pharmaceuticals and crop-enhancing agrochemicals to industrial solvents and everyday consumer products, chemical compounds are integral to countless facets of daily life [2, 3]. However, this ubiquity carries an inherent risk:

**Abbreviations:** CPU, Central Processing Unit; HDC, Hyperdimensional Computing; MAP, Multiply-Add-Permute; NR-AhR, Nuclear Receptor - Aryl Hydrocarbon Receptor; NR-AR, Nuclear Receptor - Androgen Receptor; NR-AR-LBD, Nuclear Receptor - Androgen Receptor Ligand Binding Domain; NR-Aromatase, Nuclear Receptor - Aromatase; NR-ER, Nuclear Receptor - Estrogen Receptor alpha; NR-ER-LBD, Nuclear Receptor - Estrogen Receptor Ligand Binding Domain; NR-PPAR-gamma, Nuclear Receptor - Peroxisome Proliferator-Activated; PyPi, Python Package Index; QSAR, Quantitative Structure–Activity Relationship; RAM, Random Access Memory; SMILE, Simplified Molecular Input Line Entry System; SR-ARE, Stress Response - Antioxidant Response Element; SR-ATAD5, Stress Response - ATPase Family AAA Domain; SR-HSE, Stress Response - Heat Shock Factor Response; SR-MMP, Stress Response - Mitochondrial Membrane Potential; SR-p53, Stress Response - p53 Pathway; Tox21, Toxicology in the 21st Century; VSA, Vector Symbolic Architecture.

the potential for adverse effects on human health and the environment. The huge volume of new and existing compounds entering the market vastly outspaces our capacity to rigorously evaluate their safety. Consequently, accurately and efficiently assessing the potential toxicity of these substances has become one of the most pressing challenges in public health, environmental science, and regulatory oversight [4]. Furthermore, stringent regulatory frameworks such as the European Union's Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) [5–7] and the US EPA's Toxic Substances Control Act (TSCA) [8, 9] mandate comprehensive safety assessments, underscoring the urgent need for efficient toxicity prediction methodologies.

Traditionally, toxicological assessment has depended on a paradigm of in vitro and in vivo assays [10]. While these methods have long been the gold standard, they are affected by significant limitations. They are notoriously slow, often requiring months or even years to complete for a single compound. Furthermore, they are exceptionally expensive and resource-intensive, demanding significant investment in laboratory infrastructure, materials, and specialized personnel [11]. Beyond the practical constraints, the reliance on animal testing raises substantial ethical concerns, prompting a global push toward the development of alternative, nonanimal-based testing strategies [12]. These bottlenecks create a critical gap in safety data, leaving regulators, manufacturers, and the public with an incomplete understanding of the potential hazards associated with the vast number of chemicals currently in circulation.

To bridge this gap, the field of computational toxicology has emerged, seeking to leverage computational power to predict the biological activity of chemicals based on their molecular structure. These in silico methods offer a powerful alternative, promising rapid, cost-effective, and large-scale screening of chemical libraries [13–15]. Prominent among these approaches are Quantitative Structure–Activity Relationship (QSAR) models, which establish mathematical relationships between a compound's structural or physiochemical properties (known as molecular descriptors) and its toxicological endpoint [16, 17]. In recent years, various machine learning algorithms have been successfully applied to this problem, often demonstrating high predictive accuracy [18, 19]. However, many of these conventional methods depend on a crucial and often complex preprocessing step: the calculation of handcrafted molecular descriptors. This process of feature engineering could eventually affect the model's performance because of being limited by the scope of the preselected features, a process that can inadvertently discard subtle but important structural information encoded within the molecule [20].

This paper explores a fundamentally different computational paradigm called hyperdimensional computing (HDC) [21–23], also known as vector symbolic architectures (VSA), for toxicity prediction. It is a computational paradigm receiving renewed interest that is inspired by the distributed, robust, and efficient nature of information processing in the human brain, with growing applications in biomedical sciences [24]. Instead of operating on precise, low-dimensional numbers, HDC represents information using very high-dimensional vectors, often referred to as hypervectors, typically with thousands of elements. In this framework, information is distributed across the entire vector, making the representation inherently resilient to noise and errors [25]. HDC employs a simple yet powerful algebra of vector operations to construct simple representations of complex structures by performing cognitive tasks like learning, classification, and reasoning [26–28].

These principles make it particularly suitable for the challenges of cheminformatics. Chemical structures, as represented by notations like the Simplified Molecular Input Line Entry System (SMILES) [29], are inherently sequential and compositional. The HDC framework provides a natural mechanism for encoding such symbolic data into high-dimensional space without relying on precomputed descriptors, as in the study by Ma et al. [30], from which this study takes inspiration. Our work builds upon this foundation by providing a more focused and methodologically deep investigation. While the work of Ma et al. also explored atom-wise and k-mer tokenization, our study extends this analysis by introducing a chemically-informed, fragment-based approach, enabling a broader comparison across different levels of structural granularity. Furthermore, to the best of our knowledge, this is the first study to apply the HDC paradigm specifically to the problem of toxicity prediction. We do so by leveraging the comprehensive and challenging multi-target Tox21 dataset [31], implementing a robust validation protocol to handle its severe class imbalance. Because HDC is well-suited to handling raw input values with a single-pass learning strategy, extensive preprocessing or feature engineering is not always required. By directly processing SMILES strings, an HDC model can learn to represent and associate complex structural motifs and their arrangements, capturing the patterns that determine a compound's biological activity. This approach leverages HDC's ability to map similar symbolic sequences to similar regions in high-dimensional space through associative memory, its scalability, and its ability to operate efficiently on raw, unprocessed data.

In this study, we introduce and validate a novel HDC-based pipeline for classifying chemical compounds as toxic or nontoxic. Accordingly, the proposed approach is positioned as a computationally lightweight first-pass screening method that provides a foundation for future validation of its scalability and practical utility on substantially large chemical libraries. The remainder of this manuscript is organized to systematically present our investigation. The next section details the full computational methodology, beginning with the acquisition and preparation of the dataset. We then elaborate on the crucial step of transforming chemical structures into machine-readable formats by comparing different SMILES tokenization strategies. Subsequently, we outline the design of our HDC model, from the fundamental principles of VSAs to the specific encoding schemes and classification components used. We then present the empirical results, evaluating the performance of our approach in predicting toxicity across different biological targets. Finally, we provide a discussion of these results and their broader implications, followed by a conclusion that summarizes our key findings and suggests avenues for future work.

## 2 | Materials and Methods

Here, we delineate the experimental design and computational procedures adopted to investigate the efficacy of HDC for predicting the toxicity of chemical compounds, starting with data

acquisition, different tokenization strategies, model construction, to conclude with a test and evaluation protocol.

## 2.1 | Dataset Acquisition and Preprocessing

This study is based on developing novel predictive toxicity models by leveraging the Tox21 dataset, a publicly available collection of chemical compounds and their associated toxicity labels [31]. This dataset originates from the Tox21 program, a collaborative effort by federal agencies to identify compounds that disrupt human biological pathways. The dataset structure is characterized by two primary components for each chemical entry: a SMILES string representing the compound structure and 12 binary columns indicating its interaction with specific biological receptors. A value of 1 under a receptor column for a specific compound indicates an interaction, 0 otherwise. In this context, a compound is considered toxic if it shows an interaction with any of the 12 receptors.

A total of 7832 compounds are present in this dataset. However, it is important to note that not all compounds have a defined outcome for all 12 receptors. The dataset contains missing values, which signify that the original experiment for a specific compound-receptor pair was inconclusive (for example, due to noisy data or an ambiguous response). Consequently, the total number of compounds (toxic + nontoxic) analyzed for each individual receptor is typically less than the total number of compounds reported in the Tox21 dataset.

The 12 receptors included in the Tox21 dataset, along with their brief descriptions, are summarized in Table 1.

## 2.2 | SMILES Tokenization

Here, we introduce different techniques employed for transforming chemical structures, represented as SMILES strings, into a tokenized format for subsequent hypervector encoding. SMILES is a line notation that allows a chemical structure to be represented by a short ASCII string. It provides a compact and unambiguous way to describe molecular graphs, encoding information about atoms, bonds, and molecular topology [29]. For each

**TABLE 1** | List of the 12 biological receptors used in the Tox21 dataset to differentiate toxic and nontoxic chemical compounds.

| Receptor | Toxic compounds | Nontoxic compounds | Description |
|---|---|---|---|
| NR-AR | 309 | 6956 | *Nuclear Receptor – Androgen Receptor*: involved in male sexual development and function, often targeted by endocrine disruptors. |
| NR-AR-LBD | 237 | 6521 | *Nuclear Receptor – Androgen Receptor Ligand Binding Domain*: specifically targets the ligand-binding domain of the androgen receptor, a common site for chemical interactions. |
| NR-AhR | 768 | 5781 | *Nuclear Receptor – Aryl Hydrocarbon Receptor*: a ligand-activated transcription factor involved in the regulation of genes in response to environmental pollutants. |
| NR-Aromatase | 300 | 5521 | *Nuclear Receptor – Aromatase*: an enzyme responsible for a key step in the biosynthesis of estrogens, inhibition of which can have endocrine effects. |
| NR-ER | 793 | 5400 | *Nuclear Receptor – Estrogen Receptor alpha*: that is activated by the hormone estrogen, playing a role in female reproductive function and other processes. |
| NR-ER-LBD | 350 | 6605 | *Nuclear Receptor – Estrogen Receptor Ligand Binding Domain*: focuses on the ligand-binding domain of the estrogen receptor, a critical site for endocrine disruption. |
| NR-PPAR-gamma | 186 | 6264 | *Nuclear Receptor – Peroxisome Proliferator-Activated Receptor gamma*: a nuclear receptor that plays a central role in adipogenesis, lipid metabolism, and glucose homeostasis. |
| SR-ARE | 942 | 4890 | *Stress Response – Antioxidant Response Element*: a DNA regulatory element that controls the expression of a battery of genes involved in the cellular defense against oxidative stress. |
| SR-ATAD5 | 264 | 6808 | *Stress Response – ATPase Family AAA Domain Containing 5*: involved in DNA repair pathways, and its disruption can lead to genomic instability. |
| SR-HSE | 372 | 6095 | *Stress Response – Heat Shock Factor Response Element*: a DNA sequence that regulates the expression of heat shock proteins, crucial for cellular stress response. |
| SR-MMP | 918 | 4892 | *Stress Response – Mitochondrial Membrane Potential*: assesses the integrity of mitochondrial function, as disruption of membrane potential can indicate cellular toxicity. |
| SR-p53 | 423 | 6351 | *Stress Response – p53 Pathway*: a critical tumor suppressor protein that regulates cell cycle arrest, apoptosis, and DNA repair in response to cellular stress. |

chemical compound in the Tox21 dataset, its unique SMILES string served as the primary input for our tokenization and encoding pipeline. Unlike traditional approaches that rely on precomputed molecular descriptors [19], our method directly processes the SMILES strings, leveraging different tokenization granularities to explore their impact on HDC performance.

In order to capture diverse structural information from the SMILES strings, three distinct tokenization strategies were implemented and compared. Each strategy decomposes the SMILES string into a sequence of fundamental units, or tokens, which then form the basis for hypervector generation:

1. *Atom-wise*: each individual atom and special SMILES character (e.g., =, #, (, ), [, ]) was treated as a distinct token. This method provides the most granular representation of the chemical structure, preserving information down to the level of individual atomic species and bonding types. For example, the SMILES string *CC(=O)O* would be tokenized into *C*, *C*, (, =, *O*, ), *O*. This approach ensures that every atomic and structural detail explicitly represented in the SMILES string contributes to the token set;
2. *K-mer-based*: this involves segmenting the SMILES strings into overlapping sequences of characters of a fixed length *k*. This method is analogous to n-gram approaches used in natural language processing, allowing the capture of local structural patterns within the chemical compounds. For instance, with $k = 2$, the SMILES *CC(=O)O* would yield tokens such as *CC*, *C*(, (=, *=O*, *O*), )*O*. This approach helps in identifying common motifs and their immediate chemical environment.
3. *Fragment-based*: this tokenization method aims to capture chemically meaningful substructures or functional groups as tokens. This approach leverages predefined chemical fragments that represent common building blocks of molecules. For example, a carbonyl group *C(=O)* or a benzene ring *c1ccccc1* could be recognized as single tokens. This strategy provides a higher-level representation compared to atom-wise and k-mer tokenization, focusing on the presence of known chemical functionalities. It is important to note that this fragment-based approach partially overlaps conceptually with traditional substructure-based molecular descriptors (like MACCS keys). It serves as a bridge between pure, sequence-agnostic tokenization (like k-mers) and traditional feature engineering.

## 2.3 | HDC Model

This section details the core components of the HDC framework as applied in this study, including the architectural principles behind the HDC model and the strategies for encoding tokens into high-dimensional vectors.

### *2.3.1 | Design Principles of a VSA*

HDC, also known as VSA, is a computational paradigm receiving renewed interest that leverages high-dimensional vectors to represent and process information [21, 23]. Unlike traditional computing, which relies on precise, low-dimensional data, HDC operates on vectors with thousands of dimensions, where individual elements carry little meaning but the overall pattern and relationships between vectors are highly significant. Importantly, HDC can successfully represent nonnumerical, categorical, and temporal information thanks to its symbolic representation capability. HDC typically employs either binary hypervectors (logic-1 and logic-0, particularly valuable for hardware implementations) or bipolar hypervectors (+1 and −1, commonly used in software platforms) as its atomic data type. The core principles that enable HDC to perform cognitive tasks, such as learning and classification, are:

1. *High dimensionality*: information is encoded into vectors of very high dimensionality (e.g., 1000 dimensions). Such long vectors can holistically represent multiple inputs, even when they come from different data formats, by embedding them into a single unified data structure. This high dimensionality provides a vast space for representing complex data and allows for robust and fault-tolerant operations;
2. *Distributed representation*: concepts and data are represented as distributed patterns across the entire hypervector. This means that information is not localized to a single bit or small group of bits, making the representation resilient to noise and errors;
3. *Associative memory*: HDC inherently supports associative memory, where similar inputs map to similar hypervectors. This property is crucial for tasks like classification, as it allows for the comparison of new, unseen data (from item memory) with learned patterns;
4. *Vector operations*: HDC relies on a set of fundamental algebraic operations that manipulate these high-dimensional vectors. These operations, such as binding, bundling, and permutation, allow for the encoding of relationships, the aggregation of information, and the creation of new, meaningful representations, and together are known as the Multiply-Add-Permute (MAP) model:
    1. *Binding*: A key operation that combines two hypervectors into a merged representation. Binding can be performed using element-wise multiplication for bipolar vectors (+1/−1) or logic XOR for binary vectors. This operation produces a new hypervector that encodes the association of the two input vectors. Importantly, the original hypervectors can be approximately recovered from the bound result if one of the components is known.
    2. *Bundling*: An operation (i.e., element-wise addition in bipolar space or population count in binary space) that aggregates multiple hypervectors into a single representative hypervector. Bundling is typically used to summarize information across features or samples. For example, in image processing, hypervectors of pixel intensity values after being bound with their corresponding positional hypervectors are bundled across all pixels to form a single hypervector representation. Bundling across samples of the same class yields a prototype hypervector that summarizes the overall class.
    3. *Permutation*: A scrambling operation (i.e., circular shift) that reorders the elements of a hypervector to encode

order or temporally relevant information of each hypervector. For instance, in language processing, atomic symbols (letters) represented as hypervectors can be permuted according to their position within a word, thereby preserving the sequential structure. Permutation also helps generate nearly orthogonal hypervectors, which improves independence among tuples and strengthens encoding capacity.

These design principles collectively enable HDC architectures to learn from data, make classifications, and exhibit properties akin to human memory and cognition, all while maintaining a high degree of robustness and efficiency.

### 2.3.2 | *Hypervector Encoding Strategy*

Because SMILES structures are inherently symbolic, HDC provides a naturally compatible framework for encoding them. After the tokenization of a SMILES string, the resultant tokens from each of the three strategies previously presented are transformed into high-dimensional bipolar vectors. This fundamental HDC encoding translates symbolic information into a numerical format suitable for high-dimensional yet simple arithmetic for a single-pass learning method. The specific encoding approach varied depending on the tokenization strategy to preserve or abstract positional information as required. For all tokenization strategies, we employed the random projection encoding method: each unique token identified across the entire dataset for a given tokenization strategy is assigned to a unique, randomly generated, and high-dimensional bipolar vector. These base hypervectors are sparse and balanced, containing equal numbers of $+1$ and $-1$ entries to maximize orthogonality. In probabilistic terms, the likelihood of a position taking $a+1$ or $-1$ value is approximately 0.5, representing an unbiased distribution between the minimum ($P=0$) and maximum ($P=1$) probabilities. The dimensionality of the generated hypervectors is typically set to $D=10{,}000$, which provides sufficient representational capacity and robustness.

In the case of atom-wise and k-mer-based tokenization, in order to preserve the sequential order of tokens within a SMILES string, a positional encoding scheme is applied. After generating a base hypervector for each unique token, the hypervector for a token at a specific position within a SMILES sequence is permuted. Specifically, the hypervector of the token at position $i$ (where the first token is at position 0) is circularly shifted by $i$ positions. This permutation operation effectively binds the token's identity with its positional context. Once all tokens in a SMILES string are encoded with their respective positional permutations, the hypervectors are all bundled together into a single, composite hypervector representing the entire chemical compound. For instance, if a SMILES string is tokenized into a sequence of tokens $T_0$, $T_1$, $T_2$, ..., $T_N$ with $N$ number of tokens, and their corresponding base hypervectors are $H_{T0}$, $H_{T1}$, $H_{T2}$, ...$H_{TN}$, then the compound hypervector $H_C$ is derived as $H_C = \rho^0(H_{T0}) \oplus \rho^1(H_{T1}) \oplus \rho^2(H_{T2}) \oplus \ldots \oplus \rho^N(H_{TN})$, where $\rho^i(H_{Ti})$ denotes a circular shift of the hypervector $H$ by $i$ positions, and $\oplus$ represents the bundling operation.

On the other hand, in the case of fragment-based tokenization, where the order of fragments is implicitly captured by the fragment definitions themselves, the hypervector for an entire SMILES string is constructed by directly bundling the base hypervectors of its constituent fragments. The overall encoding procedure is outlined in Algorithm 1.

### 2.3.3 | *Classification Model Components*

The HDC model implemented for predicting the toxicity of chemical compounds consists of several key components that facilitate the learning and inference processes:

1. *Item memory*: this serves as a dictionary or lookup table that stores the unique base hypervectors for all individual tokens (atoms, k-mers, or fragments) identified during the tokenization phase. Each unique token is mapped to a distinct, randomly generated bipolar hypervector of a chosen dimensionality $D$. This memory is static once initialized and provides the fundamental building blocks for constructing compound hypervectors;
2. *Associative memory (class prototype)*: for each of the 12 receptors, an associative memory is constructed. This memory stores a prototype hypervector for each class (toxic and nontoxic). These class prototypes are learned during the training phase and represent the aggregated characteristics

**ALGORITHM 1** | Pseudocode for the compound hypervector encoding process. The function takes a SMILES string and an encoding type as input and returns a single composite hypervector (HC) by bundling the permuted (for atom-wise and k-mer-based tokenizations) or base (for fragment-based tokenization) hypervectors of its constituent tokens.

**Input:** SMILES_string, ItemMemory (maps tokens to base hypervectors), EncodingType

**Output:** CompoundHypervector (HC)

```
function EncodeSMILES(SMILES_string, ItemMemory, EncodingType)
  Tokens ← Tokenize(SMILES_string, EncodingType)
  HC ← Initialize a zero vector of dimension D
  if EncodingType is "atom-wise" or "k-mer-based" then
    for i from 0 to length(Tokens) do
      Token ← Tokens[i]
      BaseHV ← ItemMemory[Token]
      PositionHV ← Permute(BaseHV, i) # Circular shift by i
      HC ← HC + PositionHV # Bundling
  else if EncodingType is "fragment-based" then
    for each Token in Tokens do
      BaseHV ← ItemMemory[Token]
      HC ← HC + BaseHV # Bundling
  return HC
```

of all compounds belonging to a specific class for a given receptor. The prototype for a class is formed by bundling all the individual compound hypervectors that belong to that class in the training set;

3. *Encoder*: the encoder module is responsible for taking a raw SMILES string and, based on the chosen tokenization strategy (atom-wise, k-mer, or fragment-based), generating its corresponding compound hypervector. This involves tokenizing the SMILES string, retrieving or permuting the base hypervectors from the item memory, and then building them according to the rules defined above;
4. *Classifier*: this module performs the inference step. For an unseen compound's hypervector, it calculates its cosine similarity to each of the class prototypes (toxic and nontoxic) stored in the associative memory for the specific receptor being evaluated. The compound is then classified into the class whose prototype hypervector exhibits the highest similarity to the compound's hypervector.

#### 2.3.4 | *Error Mitigation Process*

While HDC models are inherently robust and can achieve good performance with a single pass through the training data (a process that takes the name of one-shot learning), their accuracy can often be further refined through iterative training. This iterative process serves as a crucial error mitigation mechanism, particularly beneficial when dealing with noisy or imbalanced datasets.

During each training iteration, the class prototypes within the associative memory are updated. This update mechanism involves presenting training samples to the model and adjusting the respective class prototype hypervectors based on the classification outcome. Specifically, if a training sample is correctly classified, its hypervector reinforces its assigned class prototype. Conversely, if a training sample is misclassified, its hypervector can be used to slightly adjust the prototypes to minimize future misclassifications. This error mitigation process encompasses two key concepts of HDC's usage: one is incremental learning, which reinforces the correctly labeled class dynamically during training. The second approach is to dynamically unlearn the misclassified training sample from the trained model by simply applying a reverse operation, i.e., subtraction. Compared to traditional learning systems, this approach is cost-effective for both incremental and unlearning concepts, utilizing algebraic operations only. The proposed iterative refinement helps to:

1. *Reduce noise sensitivity*: by repeatedly exposing the model to training data, the prototypes become more robust to individual noisy samples, as the collective influence of correctly classified samples gradually outweighs the impact of outliers;
2. *Improve class separation*: over a few iterations (epochs), the prototypes for different classes (toxic and nontoxic) become more distinct and representative of their respective categories, leading to clearer boundaries in the high-dimensional space. This enhances the model's ability to differentiate between classes during inference;
3. *Address data imbalance*: in datasets where one class is significantly more prevalent than another, initial prototypes might be biased toward the majority class. Iterative updates can help the model to better learn the characteristics of the minority class.

The number of training iterations is empirically determined to achieve optimal performance and converge to a minimal error rate, close or ideally equal to 0. The retraining procedure is detailed in Algorithm 2.

To provide a holistic view of the entire experimental workflow, from model creation to the classification of new compounds, the complete process is summarized in Figure 1. This flowchart illustrates the distinct training and prediction phases, integrating the concepts detailed in Algorithms 1 and 2 into a single, cohesive architecture.

## 3 | Results

This section presents the empirical findings from our evaluation of the HDC model for toxicity prediction. The computational pipeline was constructed primarily using the Python *hdlib* library v0.1.21 [32] for the core HDC implementation. We first detail the experimental setup, our strategy for handling class imbalance, and the evaluation metrics. We then analyze the impact of k-mer length on model performance to identify an optimal configuration. Subsequently, we provide a comprehensive comparison analysis of the three primary tokenization strategies (i.e., atom-wise, k-mer-based, and fragment-based) across all 12 toxicity targets in the Tox21 dataset. This tokenization was performed using *SmilesPE* v0.0.3 [33] for the atom-wise and k-mer-based approaches, and the *RDKit* library [34] release 2025.03.5 to extract chemically meaningful fragments for the fragment-based approach. We then demonstrate the effectiveness of the iterative error mitigation process, analyze the computational performance, and provide a contextual comparison against traditional, descriptor-based machine learning methods.

### 3.1 | Experimental Setup and Evaluation Metrics

All experiments were conducted using hypervectors of dimensionality $D = 10,000$, a dimension shown to provide sufficient capacity for robust information encoding. The entire pipeline was developed in Python 3.11, and the computational tasks were executed on a workstation equipped with 4 Intel Xeon Platinum 8276 L Central Processing Units (CPUs) (112 cores/224 threads) @ 2.20 GHz and 6 TB of Random Access Memory (RAM) running CentOS Linux 7. The dataset for each of the 12 biological receptors was partitioned into a training set (80%) and test set (20%).

A significant challenge in the Tox21 dataset is the severe class imbalance for individual receptor targets, where the number of nontoxic compounds vastly exceeds the number of toxic compounds, as shown in Table 1. To address this problem, we implemented a repeated undersampling and averaging strategy. For a given receptor, let $X$ be the number of compounds in the minority (toxic) class, and $Z$ be the number in the majority (nontoxic) class. We calculated the ratio $N = Z/X$ and randomly partitioned

**ALGORITHM 2** | Pseudocode for the iterative model retraining procedure. The function initializes class prototypes using one-shot learning and then refines them over multiple epochs by adjusting the prototypes based on misclassifications to minimize the training error rate.

**Input:** TrainingSet (compound hypervectors and their true labels), NumEpochs

**Output:** TrainedPrototypes (associative memory)

```
function TrainModel(TrainingSet, NumEpochs)
    # Step 1: One-shot learning to initialize prototypes
    Prototype_Toxic ← Bundle all HCs from TrainingSet with label "toxic"
    Prototype_NonToxic ← Bundle all HCs from TrainingSet with label "nontoxic"
    # Step 2: Iterative refinement
    for epoch from 1 to NumEpochs do
        for each (HC, TrueLabel) in TrainingSet do
            # Classify the sample
            Sim_Toxic ← CosineSimilarity(HC, Prototype_Toxic)
            Sim_NonToxic ← CosineSimilarity(HC, Prototype_NonToxic)
            PredictedLabel ← "toxic" if Sim_Toxic > Sim_NonToxic else "nontoxic"
            # Update prototypes on misclassification
            if PredictedLabel != TrueLabel then
                if TrueLabel is "toxic" then
                    Prototype_Toxic <- Prototype_Toxic + HC
                    Prototype_NonToxic ← Prototype_NonToxic - HC
                else
                    Prototype_NonToxic ← Prototype_NonToxic + HC
                    Prototype_Toxic ← Prototype_Toxic - HC
    return (Prototype_Toxic, Prototype_NonToxic)
```

the $Z$ nontoxic compounds into $N$ distinct, nonoverlapping subsets. Subsequently, we performed $N$ independent training and evaluation runs in fivefolds cross-validation. In each run, the full set of $X$ toxic compounds was combined with one of the $N$ subsets of nontoxic compounds to create a nearly perfectly balanced training dataset.

While 400–2000 molecules per class (depending on the receptor) might be considered a small sample size for traditional deep neural networks, it is well-suited for the HDC framework. Because HDC leverages one-shot or few-shot learning by aggregating data into holistic, high-dimensional class prototypes, it is significantly more data-efficient and less prone to the rapid overfitting seen in traditional complex machine learning models trained on limited sample sizes [35].

A complete HDC model was trained on each of these $N$ balanced datasets, and its performance was evaluated on the test set. The final performance metrics reported for that receptor (i.e., Accuracy, Precision, Recall, and F1-score) are the average of the scores obtained from all $N$ models. This entire procedure was repeated independently for all 12 receptors. This approach ensures that the model is evaluated against all nontoxic samples while mitigating the bias introduced by class imbalance during training. Given this setup, the F1-score is emphasized as the primary indicator of robust performance. The entire process of building, training, and evaluating the models for all 12 receptors, including all undersampling runs in fivefolds cross-validation, considering all 3 tokenization methods for a total of 2960 classification models, was completed in approximately 2 h and 23 min, underscoring the computational efficiency of the overall methodology.

## 3.2 | Optimization of K-Mer Length

To determine the optimal configuration of the k-mer-based tokenization strategy, we systematically evaluated model performance using different k-mer lengths, specifically for $k$ values ranging from 2 to 10. The objective was to identify a length that effectively captures meaningful local structural motifs from the SMILES strings without becoming overly specific or sparse.

Our analysis revealed a clear and consistent trend: shorter k-mer lengths yielded superior predictive performance. As illustrated in Figure 2, the highest average F1-score across all toxicity targets was achieved with $k = 4$, reaching a value of ~64%.

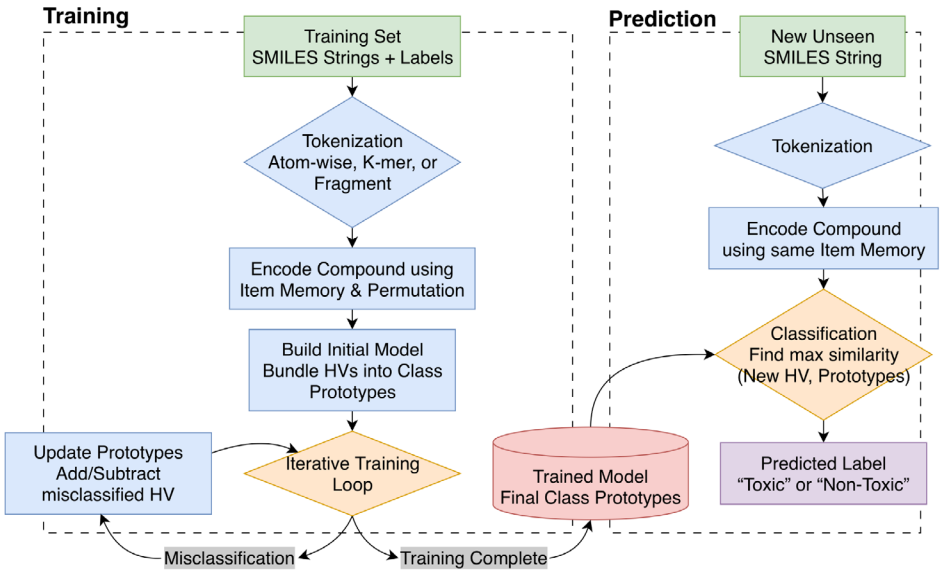


**FIGURE 1** | Flowchart of the complete VSA-based toxicity prediction pipeline. The diagram is divided into two main phases: (i) the Training phase begins with the labeled training dataset, proceeds through tokenization and encoding (as detailed in Algorithm 1), and uses an iterative refinement loop (as detailed in Algorithm 2) to produce the final, trained class prototypes; (ii) the Prediction phase shows how a new, unlabeled SMILES string is processed through the same encoding pipeline and then classified by comparing its resulting hypervector (HV) against the trained prototypes to determine the most likely class.

As the value of $k$ is increased, a gradual but steady decline in performance was observed. For instance, with $k$= 5, the average F1-score dropped to 62%, and for $k$ = 8 it further decreased to 59%. This performance degradation suggests that while longer k-mers capture more extended local contexts, they do so at the cost of generality. Longer sequences are statistically rarer, leading to a sparser vocabulary of tokens and reducing the chance of finding recurring, predictive patterns between the training and testing sets. In contrast, a k-mer length of 4 appears to strike an effective balance, creating tokens that are complex enough to represent local chemical environments (e.g., *C*(=*O*)) but common enough to be generalizable. Based on these findings, a k-mer length of $k$ = 4 was selected as the optimal value and used for all subsequent comparative analyses.

### 3.3 | Comparative Analysis of Tokenization Strategies

The main aim of our study was to compare the efficacy of different structural representation granularities for HDC-based toxicity prediction. We evaluated the performance of models built using atom-wise, fragment-based, and the optimized k-mer-based ($k$ = 4) tokenization strategies. Table 2 summarizes the

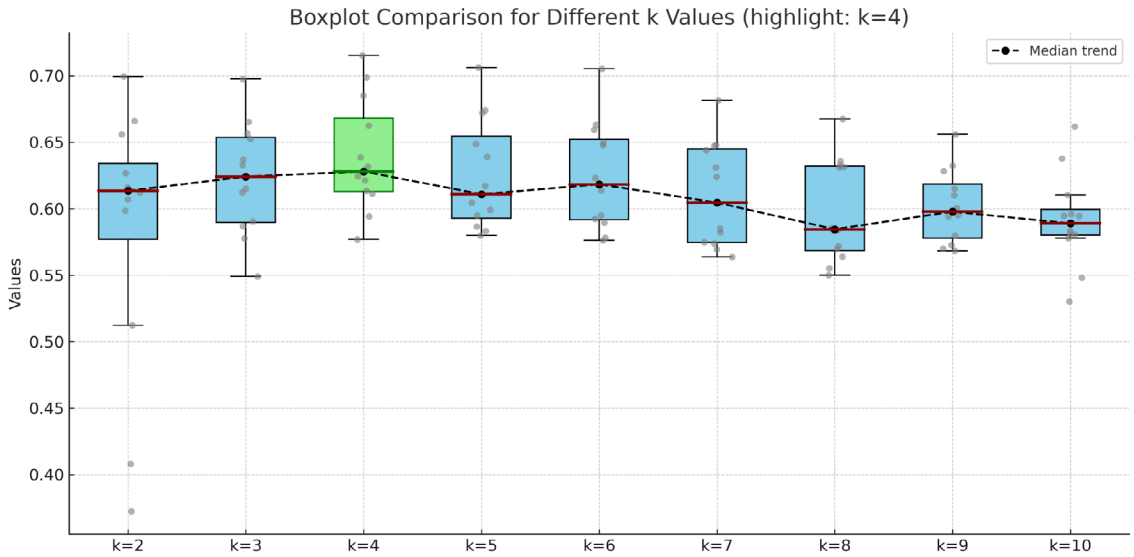


**FIGURE 2** | Boxplot built over the F1-scores computed on all 12 HDC models (one model per receptor) considering $Y$ different partitions, each of them evaluated in fivefolds cross-validation, highlighting $k$ = 4 as the top-performing one (colored in green). F1-score ranges from 0 to 1, with higher values indicating better performance.

**TABLE 2** | Average classification performance of different tokenization strategies, reporting accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC) with their standard deviations. Accuracy, Precision, Recall, and F1-score range from 0 to 1, while MCC ranges from −1 to + 1 (the higher, the better).

| Tokenization strategy | Average accuracy | Average precision | Average recall | Average F1-score | Average MCC |
|---|---|---|---|---|---|
| Atom-wise | 0.6293 ± 0.04 | 0.6441 ± 0.04 | 0.6050 ± 0.11 | 0.5962 ± 0.08 | 0.2730 ± 0.03 |
| K-mer-based, $k = 4$ | 0.6427 ± 0.04 | 0.6333 ± 0.04 | 0.6491 ± 0.04 | 0.6394 ± 0.04 | 0.2872 ± 0.03 |
| Fragment-based | 0.5907±0.02 | 0.5815 ± 0.02 | 0.5845 ± 0.04 | 0.5803 ± 0.03 | 0.1822 ± 0.03 |

average classification performance across all 12 Tox21 targets for each method.

The complete, disaggregated performance data for each of the $N$ undersampling runs, from which the averages in Table 2 were derived, are provided for atom-wise, k-mer-based, and fragment-based strategies in Supporting Information: Spreadsheets S1–S3, respectively.

The aggregated results clearly indicate that the k-mer-based ($k = 4$) tokenization strategy slightly outperformed the other two methods, achieving the highest average F1-score of ~64%. This performance suggests that capturing local, overlapping character sequences within the SMILES string provides an effective balance of granularity and contextual information for the HDC model.

In contrast, the atom-wise and fragment-based tokenization methods yielded the lowest performance, with an average F1-score of ~59% and ~58%, respectively. The performance of the atom-wise models is significant as it validates the hypothesis that the most granular representation of a chemical structure does not contain sufficient information for an effective classification. On the other hand, the performance of the fragment-based models is equally significant because, while this approach is designed to capture chemically meaningful functional groups, its effectiveness appears limited by the predefined nature of its fragment library. This method is inherently biased toward known chemistry and may fail to represent novel structural motifs or subtle atomic configurations that are critical for determining toxicity, which the other, more agnostic methods successfully capture.

To provide a more nuanced view, Table 3 presents the F1-scores for all 12 toxicity targets, showcasing how performance can vary depending on the biological endpoint.

For the NR-AR-LBD and SR-MMP receptors, the k-mer-based approach showed a distinct advantage, suggesting that specific local substructures, rather than just individual atoms or large functional groups, are key drivers of activity. Interestingly, for the NR-AR-LBD receptor, all models performed relatively well, with the fragment-based model being more competitive compared to the other models built with the same tokenization method for the other receptors. This may indicate that toxicity for this specific receptor is strongly linked to the presence of a few well-defined, reactive functional groups that are adequately captured by the fragment library.

**TABLE 3** | Comparison of F1-scores for the three tokenization strategies on the 12 Tox21 receptors. Each value is derived from a two-level averaging process: for each of $N$ random sampling runs, a fivefold cross-validation was performed. The reported score is the final average of the F1-scores from all $N$ runs. The number of runs $N$ varies for each receptor and it is defined as the number of nontoxic compounds divided by the number of toxic compounds. Scores exceeding 65% are highlighted in green. F1-score ranges from 0 to 1, with higher values indicating better performance.

| | Tokenization strategy | | |
|---|---|---|---|
| **Receptor (random samplings)** | **Atom-wise** | **K-mer-based, $k = 4$** | **Fragment-based** |
| NR-AR (22) | 0.6573 ± 0.01 | 0.6625 ± 0.01 | 0.6324 ± 0.01 |
| NR-AR-LBD (27) | 0.6936 ± 0.01 | 0.7152 ± 0.01 | 0.6386 ± 0.02 |
| NR-AhR (7) | 0.6628 ± 0.03 | 0.6850 ± 0.01 | 0.5499 ± 0.01 |
| NR-Aromatase (18) | 0.6742 ± 0.01 | 0.6113 ± 0.02 | 0.6135 ± 0.01 |
| NR-ER (6) | 0.3986 ± 0.07 | 0.5942 ± 0.00 | 0.5396 ± 0.02 |
| NR-ER-LBD (18) | 0.6138 ± 0.04 | 0.6387 ± 0.01 | 0.5746 ± 0.01 |
| NR-PPAR-gamma (33) | 0.6004 ± 0.02 | 0.6242 ± 0.02 | 0.5920 ± 0.01 |
| SR-ARE (5) | 0.4610 ± 0.09 | 0.6135 ± 0.01 | 0.5290 ± 0.02 |
| SR-ATAD5 (25) | 0.6028 ± 0.02 | 0.6316 ± 0.01 | 0.5645 ± 0.02 |
| SR-HSE (16) | 0.5536 ± 0.04 | 0.5768 ± 0.01 | 0.5502 ± 0.02 |
| SR-MMP (5) | 0.6314 ± 0.02 | 0.6988 ± 0.00 | 0.6112 ± 0.02 |
| SR-p53 (15) | 0.6059 ± 0.05 | 0.6214 ± 0.01 | 0.5692 ± 0.01 |

### 3.4 | Effectiveness of the Iterative Error Mitigation Process

To evaluate the efficacy of the iterative training procedure as an error mitigation mechanism, we monitored the model's training error rate at each iteration up to a maximum number of 100 iterations. The training error rate is defined as the sum of false positives and false negatives divided by the total number of samples in the training set. This metric provides a direct measure of how well the model's class prototypes have learned to represent the training data.

The results showed a rapid and consistent reduction in training error across all tokenization strategies and toxicity targets. Considering all models built in this study, we reached an initial error rate of ~40% in the worst case. As the iterative process began, the class prototypes were progressively refined, leading to a steep decline in the error rate. For the majority of models, the training error rate converged to a value at or near 0%. This convergence indicates that the iterative updates successfully adjusted the prototype hypervectors to correctly classify nearly all samples in the training set. This process effectively strengthens the representation of each class, improving the separation between them in the high-dimensional space and making the final model more robust. The findings confirm that iterative training is a crucial step for optimizing the HDC model and minimizing classification errors learned from the data.

### 3.5 | Analysis of Computational Performance

A key motivation for exploring HDC is its potential for computational efficiency. We measured the total training time for a single balanced model and the average inference time per compound for each tokenization strategy.

The k-mer-based model ($k = 4$) was the fastest to train, taking 1.85 s. The atom-wise and fragment-based models required slightly more time, taking 8.81 and 5.10 s respectively. However, all individual models were trained in under 10 s, a time drastically lower than that required for training other conventional machine learning models. Most importantly, the inference time for all models was negligibly small.

To further characterize the computational footprint, we also monitored the resource utilization for a single model run. The primary memory consumer was the item memory, which stores the token-to-hypervector mappings. Consecutively, the peak RAM usage was directly proportional to the vocabulary size of the tokenization strategy. The k-mer-based ($k = 4$) model, with its large vocabulary of unique 4-mers, required the most memory, typically utilizing around 750 MB of RAM. The fragment-based and atom-wise models, with their more constrained vocabularies, were significantly leaner, requiring approximately 300 MB and less than 100 MB of RAM, respectively. In all cases, the training process for a single model was computationally lightweight, utilizing a single CPU core.

This demonstrates the profound scalability of the HDC approach, making it a well-suited component for high-throughput screening protocols of massive chemical libraries.

### 3.6 | Contextual Comparison With Descriptor-Based Methods

To contextualize our findings, it is important to compare our approach with conventional machine learning models that represent the state-of-the-art in QSAR. Therefore, we compare our descriptor-free HDC approach with conventional machine learning models. The comparative analysis was streamlined using PyCaret [36], an automated machine learning framework that facilitates the rapid training and evaluation of multiple models. These methods, such as Random Forest, Support Vector Machines, or Deep Learning, typically rely on a pipeline where precomputed molecular descriptors serve as input features. These descriptors (i.e., numerical features) are engineered to explicitly quantify the molecule's topological, physiochemical, and electronic properties. They are designed to capture specific characteristics believed to govern biological activity [37–39]. A comprehensive feature set in our study includes 219 descriptors, categorized as molecular properties (physicochemical), electronic/charge descriptors, topological indices, geometric/shape descriptors, fingerprint-based density features (which explicitly include Extended-Connectivity Fingerprints such as ECFP/Morgan fingerprints), constitutional descriptors (counts), functional group counts (fragments) and structural/ring system descriptors, were calculated based on the *RDKit* library's built-in functionality.

Once this feature matrix is constructed, it serves as the input for training the machine learning models. As has been extensively reported in the literature [19, 40–43] and is confirmed by our benchmark analysis shown in Figure 3, such descriptor-based models can achieve very high predictive performance on the Tox21 dataset, with F1-scores often ranging from 59% to 84%.

While these scores are often higher than the 71.52% F1-score achieved by our best-performing k-mer-based HDC model, this comparison highlights a fundamental trade-off in methodology. The often superior performance of conventional models is contingent upon an extensive and computationally intense feature engineering step, the calculation of hundreds or thousands of chemical descriptors. This process requires significant domain expertise and separates the structural representation from the learning algorithm. This extensive feature engineering step, however, introduces significant complexity and a substantial computational cost. On average, the training and fivefold cross-validation time for these models ranged from a best case of 3.47 min for Naive Bayes to a worst case of 22.73 min for AdaBoost.

In contrast, our HDC-based approach achieves a competitive F1-score of 71.52% through a much simpler, end-to-end pipeline that operates directly on the raw SMILES strings, with an average training time of less than 15 s. The value of our method, therefore, lies not in surpassing the raw accuracy of highly optimized, descriptor-based systems, but in demonstrating a fundamentally different paradigm that offers an exceptional balance of moderate predictive performance, radical simplicity, and profound computational efficiency. This makes the HDC approach particularly attractive for scenarios requiring a computationally efficient first-pass filter for preliminary screening and exploratory analysis prior to the application of more complex, descriptor-based QSAR models.

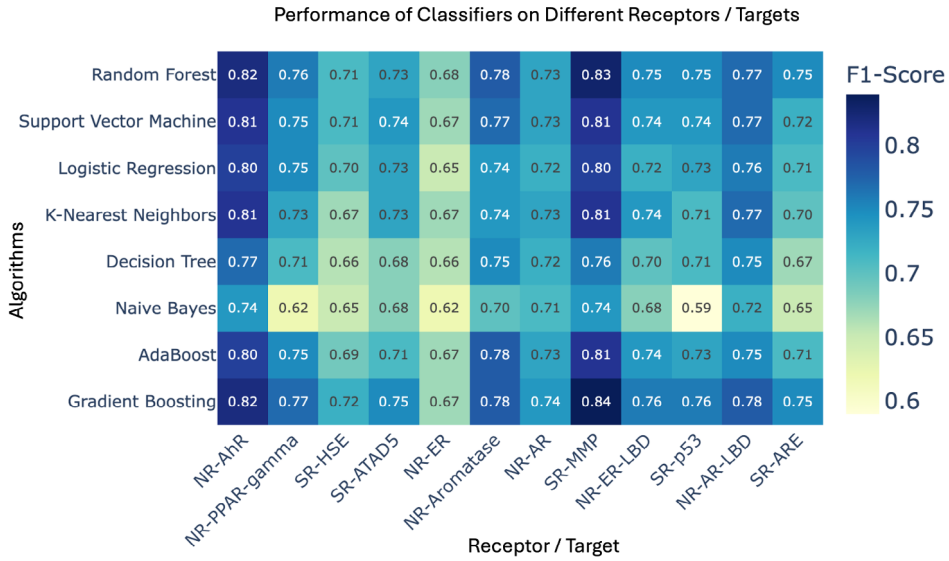


**FIGURE 3** | Comparative heatmap of F1-scores for state-of-the-art QSAR-based models trained on molecular descriptors. This figure illustrates the predictive performance of several conventional machine learning algorithms on the 12 individual toxicity prediction tasks from the Tox21 dataset. Each cell in the heatmap represents the F1-score achieved by a specific algorithm on a specific receptor target, with all results derived from a fivefold cross-validation. Unlike the HDC approach, these models were trained on a feature set of pre-calculated chemical descriptors. F1-score ranges from 0 to 1, with higher values indicating better performance.

### 3.7 | Model Interpretability and Feature Extraction

A fundamental advantage of HDC is its inherent transparency. Because our methodology explicitly stores the base hypervectors corresponding to specific SMILES tokens in the associative memory, we can directly "unbundle" the learned logic and extract human-readable rules by comparing individual token hypervectors back to the final aggregated class prototypes.

To investigate the chemically meaningful patterns learned by the model, we implemented a feature extraction analysis following the final error-mitigation step. For each of the twelve Tox21 targets, we calculated the cosine distance between the trained "Toxic" class prototype and the base hypervector of every unique token observed in the dataset. Tokens with the lowest cosine distance (highest similarity) to the toxic prototype represent the structural motifs driving the model's positive toxicity classifications.

It is important to contextualize these distance metrics mathematically. In a 10,000-dimensional hypervector space, any two randomly generated vectors are nearly orthogonal, possessing a cosine distance of approximately 1.0. Because the final class prototype is a highly superposed bundle composed of thousands of underlying structural tokens, the absolute distance to any single base token naturally remains high. However, statistically significant deviations from this 1.0 orthogonality baseline (e.g., distances dropping to 0.85, 0.75, or even lower) clearly indicate that a specific token is strongly and persistently embedded within the toxic class prototype.

To ensure the robustness of this extraction, we performed the analysis systematically across all three tokenization strategies: atom-wise, k-mer-based ($k = 4$), and fragment-based (BRICS). It should be noted that while atom-wise tokenization effectively captures basic molecular composition and SMILES syntax complexity, such as identifying that toxic compounds in this dataset frequently contain highly branched structures (denoted by "(") or multi-ring systems (denoted by ring closure digits like "6"), single characters inherently lack the structural context necessary to be considered true toxicophores [44]. Nevertheless, tracking them alongside the higher-granularity k-mer-based and fragment-based strategies demonstrates the model's consistent hierarchical learning. By evaluating the top-ranked tokens across these varying levels of chemical granularity, we identified clear consensus structural motifs for all 12 assays (Table 4).

The feature extraction protocol successfully isolated known toxicophores and distinct chemical signatures depending on the biological pathway, validating that the HDC model learns true chemical semantics rather than statistical noise.

For endocrine disruption assays targeting the androgen receptor (NR-AR, NR-AR-LBD), the model strongly weighted the sequence k-mer C[C@]12 and highly ranked BRICS fragments such as [16*]*c*1*ccccc*1 (aromatic rings) and [4*]CC([6*])═O (ketone/carbonyl groups). Chemically, these specific motifs correspond to the fused stereocenters and associated functional groups of the tetracyclic steroid backbone [45]. Similarly, for the estrogen receptor (NR-ER-LBD), the model identified the BRICS fragment [16*]*c*1*ccc*(O)*cc*1 (a phenol ring) and the sequence k-mer O*c*1*c* among the top features. The phenolic A-ring is widely documented as the defining pharmacophore strictly required for binding within the estrogen receptor pocket, perfectly aligning with established toxicological knowledge [46].

For the aryl hydrocarbon receptor (NR-AhR), which classically binds planar, halogenated, and polycyclic aromatic hydrocarbons (PAHs) such as dioxins, the model highly ranked aromatic fragments ([16*]*c*1*ccccc*1) and substituted aromatic k-mers (CO*c*1, N*c*1*c*), reflecting the receptor's documented structural preferences [47].

**TABLE 4** | Top 3 ranked consensus structural motifs (toxicophores) extracted from the fully trained "Toxic" class prototype across all 12 Tox21 assays. For each tokenization strategy, tokens are ranked primarily by their appearance consistency across all cross-validation folds and resampling iterations (expressed as a percentage), and secondarily by their lowest average cosine distance (Distance). A value of 100% indicates that the token was identified as a top feature in every single fold.

| Target receptor | Top 3 atom-wise tokens (%, distance) | Top 3 K-mer tokens (%, distance) | Top 3 BRICS fragments (%, distance) |
|---|---|---|---|
| NR-AR | C (100, 0.678) | C[C@]12 (100, 0.829) | [3*]O[3*] (100, 0.443) |
| | N (100, 0.950) | CC(=(100, 0.893) | [5*]N[5*] (100, 0.761) |
| | 6 (100, 0.980) | CCCC (100, 0.893) | [4*]CC([6*])=O (100, 0.678) |
| NR-AR-LBD | C (100, 0.650) | C[C@]12 (100, 0.823) | [3*]O[3*] (100, 0.387) |
| | N (100, 0.944) | CC(=(100, 0.865) | [5*]N[5*] (100, 0.759) |
| | [O-] (100, 0.972) | C#C[C@] (100, 0.932) | [4*]CC([6*])=O (100, 0.761) |
| NR-AhR | C (100, 0.881) | O=C( (100, 0.882) | [5*]N[5*] (100, 0.827) |
| | N (100, 0.933) | CO*c*1 (100, 0.891) | [16*]*c*1*ccccc*1 (100, 0.882) |
| | [Ca + 2] (100, 0.978) | N*c*1*c* (100, 0.893) | [3*]OC (100, 0.896) |
| NR-Aromatase | C (100, 0.734) | CCCC (100, 0.847) | [3*]O[3*] (100, 0.542) |
| | (100, 0.980) | CC(C (100, 0.891) | [5*]N[5*] (100, 0.803) |
| | N (97, 0.955) | 2C1N (100, 0.961) | [16*]*c*1*ccccc*1 (100, 0.804) |
| NR-ER | C (100, 0.900) | CCCC (100, 0.863) | [3*]O[3*] (100, 0.805) |
| | S (93, 0.963) | CC(C (100, 0.904) | [16*]*c*1*ccccc*1 (100, 0.874) |
| | [NH3+] (83, 0.979) | O=C( (100, 0.916) | [5*]N[5*] (97, 0.892) |
| NR-ER-LBD | C (100, 0.855) | O*c*1*c* (100, 0.870) | [3*]O[3*] (100, 0.541) |
| | O (100, 0.931) | CC(C (100, 0.899) | [16*]*c*1*ccc*(O)*cc*1 (100, 0.781) |
| | ( (92, 0.978) | CCCC (100, 0.899) | [16*]*c*1*ccccc*1 (100, 0.789) |
| NR-PPAR-gamma | C (100, 0.744) | CCCC (100, 0.875) | [3*]O[3*] (100, 0.472) |
| | [C@H] (99, 0.978) | O=C( (100, 0.884) | [16*]*c*1*ccccc*1 (100, 0.755) |
| | O (99, 0.942) | CC(C (100, 0.926) | [5*]N[5*] (100, 0.773) |
| SR-ARE | O (100, 0.935) | CCCC (100, 0.868) | [3*]O[3*] (100, 0.792) |
| | S (96, 0.956) | O=C( (100, 0.909) | [5*]N[5*] (100, 0.877) |
| | C (92, 0.943) | O*c*1*c* (100, 0.913) | [16*]c1ccccc1 (100, 0.903) |
| SR-ATAD5 | C (100, 0.738) | O=C( (100, 0.871) | [3*]O[3*] (100, 0.599) |
| | N (100, 0.929) | CO*c*1 (100, 0.895) | [5*]N[5*] (100, 0.729) |
| | O (99, 0.929) | CCCC (100, 0.909) | [3*]OC (100, 0.788) |
| SR-HSE | C (100, 0.887) | CCCC (100, 0.834) | [3*]O[3*] (100, 0.630) |
| | S (95, 0.949) | O=C( (100, 0.889) | [5*]N[5*] (100, 0.790) |
| | [SbH6 + 3] (76, 0.978) | CC(C (100, 0.899) | [16*]*c*1*ccccc*1 (100, 0.807) |
| SR-MMP | S (100, 0.966) | CCCC (100, 0.867) | [3*]O[3*] (100, 0.767) |
| | O (96, 0.965) | O*c*1*c* (100, 0.880) | [5*]N[5*] (100, 0.849) |
| | [nH+] (96, 0.979) | CC(C (100, 0.913) | [1*]C(=O)C1=C(C)NC(C)=C(C([1*])=O)C1[15*] (100, 0.738) |
| SR-p53 | C (100, 0.872) | O=C( (100, 0.887) | [3*]O[3*] (100, 0.624) |
| | O (99, 0.941) | O*c*1*c* (100, 0.892) | [5*]N[5*] (100, 0.745) |
| | [Mn+] (99, 0.975) | CCCC (100, 0.904) | [3*]OC (100, 0.815) |

Conversely, for broad cellular stress and genotoxicity pathways (e.g., SR-p53 for DNA damage, SR-ATAD5 for genomic instability, SR-ARE for oxidative stress), the model consistently weighted highly reactive motifs across validation folds. Features such as ether/hydroxyl linkages ([3*]O[3*]), primary/secondary amines ([5*]N[5*]), and carbonyl contexts (O=C) represent common structural alerts for electrophilic reactivity, Michael acceptors, and oxidative stress generation [48]. This cross-validated interpretability analysis confirms that despite the abstracted, high-dimensional encoding process, the HDC framework is

actively preserving and mathematically selecting for fundamental, chemically meaningful substructures across the entirety of the Tox21 dataset.

## 4 | Discussion and Conclusions

Here, we interpret the empirical findings presented in the previous section, contextualizing them within the broader field of computational toxicology. We discuss the significance of our results, from the proofs-of-concept for the descriptor-free paradigm to the practical implications of the trade-off between computational efficiency and predictive accuracy, before concluding with the limitations of this study and future research.

### 4.1 | A Proof-of-Concept for a Descriptor-Free Paradigm

The results of this study serve as a critical proof-of-concept in uniting HDC with the challenging domain of chemical toxicity prediction. Our findings establish a viable, descriptor-free pipeline and, while the absolute predictive performance does not surpass that of highly optimized, state-of-the-art methods, the work provides valuable insights into the methodology and its trade-offs. The primary contribution of this research is not the creation of a new top-performing model, but rather the systematic exploration of a novel computational paradigm with efficiency considerations, and the clear trends that emerged from it.

### 4.2 | The Importance of Local Chemical Context in HDC Encoding

The central finding of our work is the performance of the k-mer-based tokenization strategy, particularly with an optimal k-mer length of 4. This result is highly informative, suggesting that the most predictive structural information for toxicity is contained not merely in individual atoms or in large, predefined functional groups, but in the local chemical context and short-range (nearest neighbor) atomic arrangements. The k-mer approach, by creating overlapping sequences of characters, effectively captures these local motifs (e.g., C(═O)N, representing part of an amide group) without being constrained by a fixed chemical dictionary. It strikes an optimal balance between the high granularity of the atom-wise method and the high-level abstraction of the fragment-based method. This balance allows the model to learn a rich vocabulary of structural patterns that are directly relevant to biological interactions, supporting our second conclusion that the choice of tokenization strategy is critical for achieving optimal performance.

The performance of the atom-wise and fragment-based methods, while lower, provides essential context. Their results serve as a valuable baseline, demonstrating that simply encoding individual atoms or predefined fragments is less effective. This reinforces the importance of the local context captured by k-mers and highlights a potential limitation of using a fixed chemical dictionary (fragments), which may lack the flexibility to represent all relevant toxicophores. This comparative analysis leads to our second conclusion: the choice of tokenization strategy is a critical determinant of model performance in HDC-based cheminformatics.

### 4.3 | The Trade-Off: Radical Simplicity and Efficiency Versus Accuracy

One of the most significant outcomes of this study is the stark contrast in computational philosophy. Conventional QSAR models achieve higher accuracy, but this performance is contingent upon a complex, multistage pipeline involving the calculation of hundreds of chemical descriptors. Our work highlights a fundamental trade-off. This leads to our third conclusion: the HDC paradigm offers a compelling balance of moderate predictive power, radical pipeline simplicity, and profound computational efficiency. The ability to train models in seconds and perform inference in milliseconds directly from raw SMILES strings makes this approach exceptionally suitable for preliminary screenings and exploratory analyses prior to the application of more complex, descriptor-based QSAR models.

### 4.4 | Limitations and Future Directions

It is crucial, however, to acknowledge the limitations of the study. The primary limitation is the moderate predictive accuracy of the model when compared to descriptor-based systems. This indicates that while our descriptor-free approach captures essential information, it does not yet encapsulate the full spectrum of physicochemical properties that determine biological activity. Nevertheless, it opens a new and fruitful avenue in cheminformatics by emphasizing computation-efficient design. Secondly, while the HDC encoding process is transparent, interpreting the final, aggregated class prototype hypervectors to extract human-understandable chemical rules remains a significant challenge. Therefore, the structural motifs identified in this study should be interpreted as candidate toxicity-associated patterns rather than definitive toxicophores, and further chemical and statistical validation is required. Despite these limitations, this work, being the first of its kind, poses an important future research pathway for the integration of HDC with toxicity prediction. Future research should focus on bridging the performance gap. This could involve developing more sophisticated HDC encoding schemes that might incorporate physicochemical properties without full descriptor calculation, or exploring hybrid models that combine the speed of HDC for an initial screening with more complex models for subsequent refinement. The lightweight and highly parallel nature of HDC operations makes this especially attractive for application-specific and portable device design. Although additional experimental benchmarking on ultra-large datasets is required, HDC can be employed in scenarios where millions of virtual molecules must be screened (for example, drug discovery), or where on-site chemical safety monitoring requires immediate triage.

### 4.5 | Conclusions

In conclusion, this work successfully validates HDC as a promising and computationally efficient paradigm for toxicity

prediction. Future research should focus on bridging the performance gap. This could involve developing more sophisticated HDC encoding schemes that might incorporate physicochemical properties without full descriptor calculation, or exploring hybrid models that combine the speed of HDC for an initial screening with more complex models for subsequent refinement. Such advancements will be critical in developing the next generation of tools for enhancing our ability to ensure human and environmental health.

## Author Contributions

**Fabio Cumbo** and **Jayadev Joshi** conceived the research. **Fabio Cumbo** and **Sercan Aygun** designed the classification model and implemented the encoding and classification pipeline. **Fabio Cumbo**, **Kabir Dhillon**, and **Jayadev Joshi** performed the analysis. **Jayadev Joshi**, **Bryan Raubenolt**, **Davide Chicco**, and **Daniel Blankenberg** analyzed the results and provided insights from a biochemical perspective. **Daniel Blankenberg** supervised the research. **Fabio Cumbo**, **Kabir Dhillon**, **Jayadev Joshi**, **Bryan Raubenolt**, **Davide Chicco**, **Sercan Aygun**, and **Daniel Blankenberg** wrote the manuscript and agreed with its final version.


## Acknowledgments

We would like to acknowledge the use of Google's Gemini 3 Pro in refining the clarity and readability of this manuscript. The AI assistance was primarily used for tasks such as sentence restructuring, word choice suggestions, and identifying potentially unclear phrasing. We emphasize that the AI was used solely for language enhancement and did not contribute to the generation of research ideas, data analysis, or the interpretation of results. All conclusions drawn and insights presented in this manuscript are solely the product of the authors' own analysis and expertise.

## Funding

The authors have nothing to report.


## Conflicts of Interest

The authors declare no conflicts of interest.

## Data Availability Statement

The *hdlib* Python library (v0.1.21) [32] used for building the VSA in this study is an open-source tool and is freely available under the MIT license. The source code is available at https://github.com/cumbof/hdlib and it can be installed via the Python Package Index (PyPI – *pip install hdlib*) and Conda (*conda install -c conda-forge hdlib*). *SmilesPE* (v0.0.3) and *RDKit* (release 2025.03.5) are also open-source and their source code is available on GitHub at https://github.com/XinhaoLi74/SmilesPE and https://github.com/rdkit/rdkit and can be installed via the Python Package Index (*pip install SmilesPE rdkit*) and Conda as well on the Bioconda [49] channel (*conda install -c bioconda SmilesPE rdkit*). The specific pipeline developed for the classification of chemical compounds based on their SMILES representations, along with the preprocessed Tox21 dataset used in this research, are also publicly available under the examples directory of the *hdlib* GitHub repository at https://github.com/cumbof/hdlib/tree/main/examples/tox21.

## References


1. N. E. Thomford, D. A. Senthebane, A. Rowe, et al., "Natural Products for Drug Discovery in the 21st Century: Innovations for Novel Drug Discovery," *International Journal of Molecular Sciences* 19 (2018): 1578, https://doi.org/10.3390/ijms19061578.

2. W. Erskine, A. Sarker, and S. Kumar, "Crops that Feed the World 3. Investing in Lentil Improvement Toward a Food Secure World," *Food Security* 3 (2011): 127–139, https://doi.org/10.1007/s12571-011-0124-5.

3. K. Hardy, T. Orridge, X. Heynes, S. Gunasena, S. Grundy, and C. Lu, "Farming the Future: Contemporary Innovations Enhancing Sustainability in the Agri-Sector," *Annual Plant Reviews* 4 (2021): 263–294, https://doi.org/10.1002/9781119312994.apr0728.

4. A. M. B. Amorim, L. F. Piochi, A. T. Gaspar, A. J. Preto, N. Rosário-Ferreira, and I. S. Moreira, "Advancing Drug Safety in Drug Development: Bridging Computational Predictions for Enhanced Toxicity Prediction," *Chemical Research in Toxicology* 37 (2024): 827–849, https://doi.org/10.1021/acs.chemrestox.3c00352.

5. R. Cana, "Registration, Evaluation, Authorisation and Restrictions of Chemicals: An Analysis," *European Energy and Environmental Law Review* 13 (2004): 99–109, https://doi.org/10.54648/eelr2004013.

6. E. Ingre-Khans, M. Ågerstrand, A. Beronius, and C. Rudén, "Toxicity Studies Used in Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH): How Accurately Are They Reported?," *Integrated Environmental Assessment and Management* 15 (2019): 458–469, https://doi.org/10.1002/ieam.4123.

7. M. Pereira, D. S. Macmillan, C. Willett, and T. Seidle, "REACHing for Solutions: Essential Revisions to the EU Chemicals Regulation to Modernise Safety Assessment," *Regulatory Toxicology and Pharmacology* 136 (2022): 105278, https://doi.org/10.1016/j.yrtph.2022.105278.

8. S. D. G. Rayasam, P. D. Koman, D. A. Axelrad, T. J. Woodruff, and N. Chartres, "Toxic Substances Control Act (TSCA) Implementation: How the Amended Law Has Failed to Protect Vulnerable Populations From Toxic Chemicals in the United States," *Environmental Science & Technology* 56 (2022): 11969– 11982, https://doi.org/10.1021/acs.est.2c02079.

9. V. I. Singla, P. M. Sutton, and T. J. Woodruff, "The Environmental Protection Agency Toxic Substances Control Act Systematic Review Method May Curtail Science Used to Inform Policies, With Profound Implications for Public Health," *American Journal of Public Health* 109 (2019): 982–984, https://doi.org/10.2105/AJPH.2019.305068.

10. M. Sachana and A. J. Hargreaves, "Toxicological Testing: In Vivo and in Vitro Models," In *Veterinary Toxicology* (Academic Press, 2025), 143–160, https://doi.org/10.1016/B978-0-443-29007-7.00035-2.

11. C. Lynch, S. Sakamuru, M. Ooka, et al., "High-Throughput Screening to Advance In Vitro Toxicology: Accomplishments, Challenges, and Future Directions," *Annual Review of Pharmacology and Toxicology* 64 (2024): 191–209, https://doi.org/10.1146/annurev-pharmtox-112122-104310.

12. K. Taylor, S. Modi, and J. Bailey, "An Analysis of Trends in the use of Animal and Non-Animal Methods in Biomedical Research and Toxicology Publications," *Frontiers in Lab on a Chip Technologies* 3 (2024): 1426895, https://doi.org/10.3389/frlct.2024.1426895.

13. L. G. Valerio Jr., "In Silico Toxicology for the Pharmaceutical Sciences," *Toxicology and Applied Pharmacology* 241 (2009): 356–370, https://doi.org/10.1016/j.taap.2009.08.022.

14. H. Raunio, "In Silico Toxicology – Non-Testing Methods," *Frontiers in Pharmacology* 2 (2011): 9488, https://doi.org/10.3389/fphar.2011.00033.

15. G. J. Myatt, E. Ahlberg, Y. Akahori, et al., "In Silico Toxicology Protocols," *Regulatory Toxicology and Pharmacology* 96 (2018): 1–17, https://doi.org/10.1016/j.yrtph.2018.04.014.

16. D. A. Winkler, "The Role of Quantitative Structure--Activity Relationships (QSAR) in Biomolecular Discovery," *Briefings in Bioinformatics* 3 (2002): 73–86, https://doi.org/10.1093/bib/3.1.73.

17. P. Polishchuk, "Interpretation of Quantitative Structure–Activity Relationship Models: Past, Present, and Future," *Journal of Chemical Information and Modeling* 57 (2017): 2618–2639, https://doi.org/10.1021/acs.jcim.7b00274.

18. I. Grenet, Y. Yin, J.-P. Comet, and E. Gelenbe, "Lecture Notes in Computer Science," Machine Learning to Predict Toxicity of Compounds," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, (ICANN, 2018), 335–345, https://doi.org/10.1007/978-3-030-01418-6_33.

19. C. N. Cavasotto and V. Scardino, "Machine Learning Toxicity Prediction: Latest Advances by Toxicity End Point," *ACS Omega* 7 (2022): 47536–47546, https://doi.org/10.1021/acsomega.2c05693.

20. T. Stepišnik, B. Škrlj, J. Wicker, and D. Kocev, "A Comprehensive Comparison of Molecular Feature Representations for use in Predictive Modeling," *Computers in Biology and Medicine* 130 (2021): 104197, https://doi.org/10.1016/j.compbiomed.2020.104197.

21. P. Kanerva, "Hyperdimensional Computing: An Algebra for Computing With Vectors," *Advances in Semiconductor Technologies*. (2022): 25–42, https://doi.org/10.1002/9781119869610.ch2.

22. G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-Memory Hyperdimensional Computing," *Nature Electronics* 3 (2020): 327–337, https://doi.org/10.1038/s41928-020-0410-3.

23. P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation With High-Dimensional Random Vectors," *Cognitive Computation* 1 (2009): 139–159, https://doi.org/10.1007/s12559-009-9009-8.

24. F. Cumbo and D. Chicco, "Hyperdimensional Computing in Biomedical Sciences: A Brief Review," *PeerJ Computer Science* 11 (2025): e2885, https://doi.org/10.7717/peerj-cs.2885.

25. S. Zhang, R. Wang, J. J. Zhang, A. Rahimi, and X. Jiao, "Assessing Robustness of Hyperdimensional Computing Against Errors in Associative Memory : (Invited Paper)," in IEEE 32nd International Conference on Application-Specific Systems, Architectures and Processors (ASAP), (IEEE, 2021), https://doi.org/10.1109/asap52443.2021.00039.

26. F. Cumbo, E. Cappelli, and E. Weitschek, "A Brain-Inspired Hyperdimensional Computing Approach for Classifying Massive DNA Methylation Data of Cancer," *Algorithms* 13 (2020): 233, https://doi.org/10.3390/a13090233.

27. F. Cumbo, S. Truglia, E. Weitschek, and D. Blankenberg, "Feature Selection With Vector-Symbolic Architectures: A Case Study on Microbial Profiles of Shotgun Metagenomic Samples of Colorectal Cancer," *Briefings in Bioinformatics* 26 (2025): bbaf177, https://doi.org/10.1093/bib/bbaf177.

28. J. Joshi, F. Cumbo, and D. Blankenberg, Large-Scale Classification of Metagenomic Samples: A Comparative Analysis of Classical Machine Learning Techniques Vs a Novel Brain-Inspired Hyperdimensional Computing Approach, bioRxiv, 2025, https://doi.org/10.1101/2025.07.06.663394.

29. D. Weininger, "SMILES, A Chemical Language And Information System. 1. Introduction to Methodology and Encoding Rules," *Journal of Chemical Information and Computer Sciences* 28 (2000): 31–36, https://doi.org/10.1021/ci00057a005.

30. D. Ma, R. Thapa, and X. Jiao, "MoleHD: Efficient Drug Discovery Using Brain Inspired Hyperdimensional Computing," Efficient Drug Discovery Using Brain Inspired Hyperdimensional Computing," in 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), (IEEE, 2022), https://doi.org/10.1109/bibm55620.2022.9995708.

31. R. S. Thomas, R. S. Paules, A. Simeonov, et al., "The US Federal Tox21 Program: A Strategic and Operational Plan for Continued Leadership," *Altex* 35 (2018): 163–168, https://doi.org/10.14573/altex.1803011.

32. F. Cumbo, E. Weitschek, and D. Blankenberg, "Hdlib: A Python library for designing Vector-Symbolic Architectures," *Journal of Open Source Software* 8 (2023): 5704, https://doi.org/10.21105/joss.05704.

33. X. Li, and D. Fourches, "SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning," *Journal of Chemical Information and Modeling* 61 (2021): 1560–1569, https://doi.org/10.1021/acs.jcim.0c01127.

34. G. Landrum, n.d. "RDKit.," Accessed August 18, 2025, https://www.rdkit.org/.

35. P. Vergés, M. Heddes, I. Nunes, D. Kleyko, T. Givargis, and A. Nicolau, "Classification Using Hyperdimensional Computing: A Review With Comparative Analysis," *Artificial Intelligence Review* 58 (2025): 173, https://doi.org/10.1007/s10462-025-11181-2.

36. A. Moez, "An Open Source, Low-Code Machine Learning Library in Python," PyCaret, https://www.pycaret.org/.

37. R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors* (John Wiley & Sons, 2008), https://play.google.com/store/books/details?id=TCuHqbvgMbEC.

38. M. Dehmer, K. Varmuza, and D. Bonchev, *Statistical Modelling of Molecular Descriptors in QSAR/QSPR* (John Wiley & Sons, 2012), https://play.google.com/store/books/details?id=qi3EB-CMRAQC.

39. M. Karelson, *Molecular Descriptors in QSAR/QSPR* (Wiley-Interscience, 2000), https://books.google.com/books/about/Molecular_Descriptors_in_QSAR_QSPR.html?hl=&id=AeRqAAAAMAAJ.

40. L. Wu, R. Huang, I. V. Tetko, Z. Xia, J. Xu, and W. Tong, "Trade-Off Predictivity and Explainability for Machine-Learning Powered Predictive Toxicology: An In-Depth Investigation With Tox21 Data Sets," *Chemical Research in Toxicology* 34 (2021): 541, https://doi.org/10.1021/acs.chemrestox.0c00373.

41. D. Kim, J. Jeong, and J. Choi, "Identification of Optimal Machine Learning Algorithms and Molecular Fingerprints for Explainable Toxicity Prediction Models Using ToxCast/Tox21 Bioassay Data," *ACS Omega* 9 (2024): 37934, https://doi.org/10.1021/acsomega.4c04474.

42. G. Idakwo, S. Thangapandian, J. Luttrell, et al., "Structure–activity Relationship-Based Chemical Classification of Highly Imbalanced Tox21 Datasets," *Journal of Cheminformatics* 12 (2020): 1–19, https://doi.org/10.1186/s13321-020-00468-x.

43. A. Setiya, V. Jani, U. Sonavane, and R. Joshi, "MolToxPred: Small Molecule Toxicity Prediction Using Machine Learning Approach," *RSC Advances* 14 (2024): 4201–4220, https://doi.org/10.1039/D3RA07322J.

44. A. S. Kalgutkar, "Designing Around Structural Alerts in Drug Discovery," *Journal of Medicinal Chemistry* 63 (2019): 6276, https://doi.org/10.1021/acs.jmedchem.9b00917.

45. W. Gao, C. E. Bohl, and J. T. Dalton, "Chemistry and Structural Biology of Androgen Receptor," *Chemical Reviews* 105 (2005): 3352, https://doi.org/10.1021/cr020456u.

46. G. M. Anstead, K. E. Carlson, and J. A. Katzenellenbogen, "The Estradiol Pharmacophore: Ligand Structure-Estrogen Receptor Binding Affinity Relationships and a Model for the Receptor Binding Site," *Steroids* 62 (1997): 268–303, https://doi.org/10.1016/s0039-128x(96)00242-5.

47. M. S. Denison and S. R. Nagy, "Activation of the Aryl Hydrocarbon Receptor by Structurally Diverse Exogenous and Endogenous Chemicals," *Annual Review of Pharmacology and Toxicology* 43 (2003): 309–334, https://doi.org/10.1146/annurev.pharmtox.43.100901.135828.

48. T. W. Schultz, J. W. Yarbrough, R. S. Hunter, and A. O. Aptula, "Verification of the Structural Alerts for Michael Acceptors," *Chemical Research in Toxicology* 20 (2007): 1359–1363, https://doi.org/10.1021/tx700212u.

49. B. Grüning, R. Dale, A. Sjödin, et al., "Bioconda: Sustainable and Comprehensive Software Distribution for the Life Sciences," *Nature Methods* 15 (2018): 475–476, https://doi.org/10.1038/s41592-018-0046-7.

## Supporting Information

Additional supporting information can be found online in the Supporting Information section.