Distributed Representation of Chemical Fragments

Suman Chakravarti
chakravarti@multicase.com
MultiCASE Inc., 23811 Chagrin Blvd., Suite 305, Beachwood, OH 44122, USA

Abstract

This contribution describes a method for learning vector representation of chemical fragments to encode their meaningful features and to capture continuous relationships that exist between fragments. The learning method is an unsupervised word embedding algorithm borrowed from the natural language processing (NLP) field, and uses approximately 3.7 million unlabeled chemical structures from PubChem to build the training corpus. This fragment encoding addresses several issues associated with the traditional sparse ‘one-hot’ fragment representation by capturing rich relational structure that exists in the fragment space, enabling similarity calculation between fragments, avoiding bit collision in fragment based molecular fingerprints, and by expanding applicability domains of fragment based QSAR models. This work demonstrates the benefits of unsupervised learning of chemistry from very large sets of unlabeled chemical structures. The extracted knowledge can then be applied to small datasets of labeled chemicals to build QSAR models. The fragment vectors also proved to be useful for identification of structural alerts in query chemicals. In addition, they are suitable for visualizing fragment and chemical spaces. This work will have benefits in the fragment based drug discovery, QSAR analysis and in the general use of machine learning in chemistry.

Data

A number of chemical datasets were used in this study:

1. Approximately 3.7 million (3,700,103) PubChem compounds were used in training the fragment vectors. Five million compounds (from CID 1 to 5,000,000) were downloaded in SMILES format from PubChem Download Service\(^1\) for this
purpose. Inorganic part of salts, charges on certain atoms were neutralized, components of mixtures were split and only one chemical from sets of duplicates were kept. Approximately 100,000 chemicals were held out for future needs.

2. Hansen\textsuperscript{2} and Bursi\textsuperscript{3} Ames mutagenicity benchmark datasets were used. The data was preprocessed in the same way as the PubChem chemicals, except in the case of duplicates, the compound with the highest mutagenicity value was retained, leaving 6771 compounds (3639 mutagenic and 3132 non-mutagenic).

3. A dataset of 575 compounds was taken from the publication of Ghose et al\textsuperscript{4} to compute molar refractivity of the fragments.

4. An in-house dataset\textsuperscript{5} of 7000 chemicals with their experimentally observed LogP was also used to help in computing LogP contribution of the fragments.

**Introduction**

Chemical fragments are small pieces of chemical structures, representing molecular features useful in the modeling of biological or physicochemical properties of chemicals.\textsuperscript{6} Fragments represent chemistry of parts of molecules and can be generated algorithmically. They offer intuitive interpretation if used in QSAR, easy to generate and handle.

Traditionally, cheminformatics algorithms treat fragments as discrete symbols\textsuperscript{7,8}, as arbitrary indices in a list of unique fragments and provide no clue about possible relationships between individual fragments (Figure 1a). Essentially, the $i^{th}$ fragment is an $N$-dimension ‘one-hot’ vector where $N$ is the number of unique fragments in the fragment corpus (fragment vocabulary size). Only the $i^{th}$ element of the vector would be non-zero (i.e. a sparse vector). Therefore, when two such vectors are compared, it either is an exact match or no match at all. Such representation also depends heavily on the decisions made by experts, for example, an epoxide ring and an aziridine ring would be treated as totally different fragments unless listed as three membered rings with a heteroatom. Moreover, decisions made for one application may not be suitable to another.
Fragments are also commonly used for building fingerprints that encode chemical structures.\textsuperscript{7,8} Fingerprints are usually composed of a series of binary digits (bits). Each bit indicates presence/absence of a certain fragment in the molecule (Figure 2). Similarity between two molecules can be computed by counting the number of bits that are ‘on’ in both. However, there are some issues associated with such binary fingerprints:

i. Number of unique fragments (fragment vocabulary) from a set of chemicals is usually large and the fingerprint is much smaller (128, 256, 512 bits etc.), resulting in more than one fragments competing for the same bit (bit collision). A combination of hashing and random number generation is used to determine the bit position for a fragment. Since the computed bit position has no relation with chemistry, completely different fragments can end up in the same bin.

ii. ‘On’ bits in the fingerprint only denote fragments’ presence in the chemical. Two chemicals that contain very similar but not identical fragments will turn ‘on’ different bits and the computed similarity value will not reflect the true chemical similarity.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Two different representations of molecular fragments – (a) traditional sparse and ‘one-hot’ representation and (b) distributed and dense representation developed in the current work.}
\end{figure}

\[ N = \text{Size of fragment vocabulary (e.g. 61039 in the current work)}; \ k = \text{Number of elements in the fragment vector (adjustable, e.g. 100 or 500 in the current work).} \]
Figure 2. Computation of molecular fingerprint using traditional one-hot fragments. $N$ = Size of fragment vocabulary (e.g. 61039 in the current work); $m$ = Number of fragments generated by breaking up the molecule in question.

The ‘one-hot’ fragment representation also causes problems in building fragment based QSARs. Fragments counts or presence/absence in training chemicals are used as descriptors, therefore, the X matrix of the training data becomes sparse and large in dimension. A lot of data is required for successful statistical training because even highly similar fragments occupy separate columns in the X matrix. Moreover, during prediction, if a query chemical has a fragment which is not present in any of the training set chemicals, it is labeled as an uncovered fragment and may result in an out-of-domain outcome. This issue restricts the model’s applicability domain. This problem becomes particularly serious when models are built from small training sets. In our experience, out of domain outcomes are common in fragment based models even with training sets of ~1000 chemicals.
In contrast, distributed representations characterize symbols in a continuous high dimensional vector space where similar symbols are positioned near each other. Distributed representations are one of the most successful approaches in machine learning and widely used in natural language processing (NLP) applications. They are referred to as word embeddings in NLP. The vectors are learnt automatically by a neural network with training on a large corpus of actual text while being tasked to predict the next word in the sequence from the context of earlier words. The hypothesis that words sharing similar contexts have similar meaning is central to this process and it is driven by an unsupervised training, i.e. the input data is unlabeled. In addition, does not require any prior expert knowledge.

The word embeddings were originally invented by Bengio et al in 2003 who used neural networks for this purpose. Later in 2008, Collobert and Weston were first to demonstrate the effectiveness of word embeddings in many NLP tasks. However, Mikolov et al in 2013 made it mainstream and widely adopted by developing the Word2Vec algorithm, a fast and easy to use implementation due to reduced computational complexity. Success of the distributed representations was replicated in other fields, e.g. in bioinformatics, notably Asgari and Mofrad’s application of word embeddings for biomolecular sequences.

The present work is about learning distributed representation of chemical fragments (Figure 1b), essentially with two objectives in mind: i. solving the problems of traditional ‘one-hot’ fragment representations, and ii. to find out if the existence of a part of a chemical structure probabilistically depends on the surrounding structural features, much like words in sentences? If yes, is it possible to capture such relationships in fragment encoding? The answer to the 2nd question was indeed found to be affirmative and the Word2Vec algorithm borrowed from NLP was successfully used for embedding fragments.

Distributed representations for 50,375 4-atom linear fragments were generated from ~3.7 million PubChem chemicals. The vectors successfully captured characteristics of the fragments which is demonstrated using various 2D plots of physicochemical and
biological properties mapped on the fragment vector space. The vectors allowed straightforward similarity measurements between fragments and the results made sense from a chemist’s point of view. It was also demonstrated that distributed representations of molecules can be computed simply by averaging their fragment vectors. Molecule vectors were used to predict mutagenicity of compounds using k nearest neighborhood algorithm. It was shown that distributed representation offers robustness and wider applicability domain in comparison with other fingerprint types, e.g. fragment based hashed fingerprints and PubChem keys. A technique was developed to identify structural alerts in query molecules, by checking nearest neighbors in chemical space. This is difficult, if not impossible, using traditional fingerprints and aids in the interpretability of QSARs.

Methods

Software

Python package Gensim\textsuperscript{14} was used to access the Word2Vec algorithm for learning distributed representation of chemical fragments. The R package Rtsne\textsuperscript{15} was used for generating t-Distributed Stochastic Neighbor Embedding (t-SNE) plots for visualization of vector spaces.

An in-house cheminformatics software library was used for handling chemical structures, fragmenting chemicals, QSAR analysis and other operations described in this paper.

Molecular Fragment Generation

A graph traversing algorithm was applied to generate linear heavy atom fragments from training chemicals. Only fragments of 4 heavy atoms were used for clarity and for the possibility that they may have a better chance to represent significant chemistry as compared to 1, 2 or 3 sized fragments. The methodology is equally applicable to smaller, larger or branched fragments; however, a separate set of vectors should be generated for each case. The fragments were converted to string representations using atomic and bond symbols, e.g. C3H2-C3H2-C3H2-N3H2. Atomic symbols are composed of element, hybridization, aromaticity, number of attached hydrogens,
formal charge, aromatic ring joints, membership of 3 or 4 sized rings etc. The atom symbols in fragment strings differ slightly from that of SMILES coding, e.g. C3H3 denotes a sp3 carbon with 3 hydrogens, C1 stands for sp carbons, C2 is for sp2 carbons, [c.] stands for aromatic carbons located on a ring joint, N.PI3 stands for trigonal planar nitrogens, [N3^] denotes a sp3 nitrogen in a 3 or 4 membered ring.

A dictionary of key-fragment pairs was constructed, e.g. \{key = frg_54, fragment = C3H2-C3H2-C3H2-N3H2\}. A total of 61039 unique 4 atom fragments were recorded in this dictionary.

**Building Fragment Corpus**

A chemical was treated as a body of text and its structural fragments as words. The fragment corpus is essentially a big text file containing a list of so called *fragment sentences*. Each sentence is a list of fragments that are connected (shares at least one atom) to one fragment in a chemical. Fragment sentences were generated from every training chemical using the following steps:

```
Repeat for every training chemical
{
    Repeat for every 4-atom fragment in this chemical
    {
        i. Create a fragment sentence by appending all connected fragment to this fragment.
        ii. Randomize the order of the fragments in the sentence.
        iii. Write this sentence to the fragment corpus file.
    }
}
```

In the corpus file, every sentence in placed in a separate line. Following is an example of a typical fragment sentence:

```
C3H-N3H-C2=C2 C3H-C3H2-N3H-C2 C3H-C3H2-N3-C2H N2=C2-N3H-C2
```

In practice, the keys from the fragment dictionary were used in place of the actual fragments to save memory and disk space, e.g. the above fragment sentence will be stored as following:

```frg_102 frg_103 frg_111 frg_104```
The completed corpus contained a total of 111,493,546 (~111 million) sentences, and the training corpus file size was of ~20GB size.

**Generating Fragment Vectors**

Fragments vectors were generated using *Word2Vec* word embedding algorithm\(^9,12\), which takes real text as input and learns distributed representation of words using the contextual relationships that exist between words of the input text. Two neural network based unsupervised learning architectures are available in *Word2Vec*:

*Skip-gram*: In this architecture, given the target word, the model tries to predict \(n\) words before and \(n\) words after it. The training objective is to maximize the conditional probability of observing the contextual words. Figure 3 shows skip-gram architecture for generating fragment vectors.

*Continuous bag-of-words (CBOW)*: The model tries to predict the target word given \(n\) words before it and \(n\) words after it. The training objective is to maximize the conditional probability of observing the target word. CBOW is several times faster to train than the skip-gram.

In both, \(n\) is referred to as the window size that specifies the number of the contextual words before and after the target word to be used for training.

Traditionally, probabilistic language models are trained to maximize the outcome of a softmax function. However, this process is computationally very expensive because of the need to maximize the probability of each target word against all the other words in the vocabulary at every training step. *Word2Vec* is computationally much simpler because of a simple logistic regression type binary classification objective to discriminate the target words from a set of \(w\) randomly chosen noise words (also called as the *negative sample size*) in the same context (i.e. it tries to maximize a loss function containing two terms – probability of seeing the target word given the contextual words and probability of not seeing the target word given the noise words). The value of \(w\) is adjustable and usually much smaller than the size of the vocabulary (e.g. in this work, \(w=5\), vocabulary size=61039).
In the present work, the resulting fragment vector size was set to 100 or 500, the skip-gram model was used, the window size was kept at 5, negative sample size of 5 was used, and 5 passes were made over the corpus during the training. A total of 50375 fragment vectors were successfully learnt by the embedding procedure. Fragments with less than 5 occurrences in the training corpus were excluded by the Word2Vec procedure.

Figure 3: Skip-gram neural network architecture for generating fragmenting vectors. The target fragment is shown in red and its one-hot vector is fed as the input. The objective is to get highest probability for the context fragment (shown in blue) in the softmax output layer.

Computing Similarity Between Fragment Vectors

Cosine similarity function (Eq. 1) was used to compute similarity between two fragment vectors. Pairwise similarities were calculated and recorded for the 50375 fragment vectors. Cosine similarity measures the similarity of orientation of two vectors and ranges from 0 to 1, i.e. if the angle between two vectors is 0°, the cosine similarity is 1.0 and similarity is 0.0 if the angle is 90°.

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{k} A_i B_i}{\sqrt{\sum_{i=1}^{k} A_i^2} \sqrt{\sum_{i=1}^{k} B_i^2}} \quad \ldots \quad \text{Eq. 1}$$
Computing Vectors for Whole Molecules

When fragments vectors of a molecule are added, the resulting vector represents the combined features of the whole molecule because different elements of the fragment vectors encode different features. In practice, vectors of all the fragments of the molecule are added and the resultant vector is divided elementwise by the total number of fragments in the molecule (Figure 4). The resulting fingerprints are distributed representation of the molecules and their computation does not involve hashing and consequently no bit collisions. These molecular vectors can be subjected to all the usual operations, e.g. clustering, molecular similarity searching, QSAR modeling etc. Also, the molecule and fragment vectors are of same size and both can exist in the same vector space, which allows us to measure similarity between individual fragments and whole molecules. This is not possible with ‘one-hot’ fragment representations.

Figure 4. Computing molecular fingerprints using distributed representation of fragments. $k =$ Number of elements in the fragment vector (adjustable, e.g. 100 or 500 in the current work); $m =$ Number of fragments generated by breaking up the molecule in question.

Visualizing the Vector Space

t-Distributed Stochastic Neighbor Embedding (t-SNE) was used for creating 2D plots for visualizing the fragment/molecule vectors. Since the vectors are high dimensional, a proper dimension reduction enables 2D visualization to examine if fragments or
molecules with similar properties are located near each other in the vector space. t-SNE method, developed by van der Maaten et al.\textsuperscript{16}, is particularly suitable for visualizing high dimensional data. t-SNE preserves much of the local structure while reveals the global structure as well. It is commonly used for visualizing word vectors in NLP applications.

In this work, the vectors were subjected to the t-SNE algorithm for dimensionality reduction from 100 to 2 dimensions and 10,000 iterations were performed for every t-SNE run.

**Computing Physico-Chemical Properties of Fragments**

Physicochemical properties of the fragments were calculated in order to quantitatively check if fragments with similar properties are located near each other in vector space. Octanol-water partition coefficient (LogP) and molar refractivity (MR) of fragments were computed for this purpose. Contribution of various atom types were first estimated by least square fitting of experimental LogP/MR values with the counts of various atom types. Two separate training sets (Data section) were used for this purpose.\textsuperscript{4,5} Regression coefficients of the atoms of a fragment and the intercept term were added to compute LogP/MR of the fragment.

**Results and Discussion**

A thorough visual inspection of similar fragments revealed that high cosine similarity almost always corresponds to high chemical similarity. For instance, Table 1 lists 5 closest fragments for three fragments; we can see that the most similar fragments to OH-C3H-C3H-Br was found to be OH-C3H-C3H-Cl and N3-C3H-C3H-Br, both with similarity=0.696. In this case, the algorithm correctly identified the equivalency between Cl and Br atoms and OH and N.
Table 1. Results of similarity search for three example query fragments.

| Closest Fragments       | Similarity | Closest Fragments       | Similarity | Closest Fragments       | Similarity |
|-------------------------|------------|-------------------------|------------|-------------------------|------------|
| Cl-C3H2-C2-C2          | 0.773      | OH-C3H-C3H-Br           | 0.709      | s:[c.]:[n.]:n           | 0.991      |
| C2-C2-C2-C1            | 0.655      | OH-C3H-C3H-Cl           | 0.696      | [n.]:[c.]:s:[c.]        | 0.990      |
| OH-S3-C2-C2            | 0.648      | N3-C3H-C3H-Br           | 0.696      | s:[c.]:[n.]:[c.]        | 0.988      |
| S2=C2-C2-C2            | 0.641      | OH-C3H-C3H-C1           | 0.682      | [n.]:[c.]:[c.]:[n.]    | 0.984      |
| C3H2-S3-C2-C2          | 0.637      | C3H-C3H-C3H2-Br         | 0.670      | n:n:[n.]:[c.]           | 0.981      |

Next, t-SNE procedure was applied on the fragment vectors. Numerous clusters were seen in the 2D plot, because the 50375 fragments from ~3.7 million chemicals represent a wide range of chemistry. As a start, fragments with a few different atom types were mapped on the t-SNE plot to get a better insight:

- a. Fragments with one or more aromatic nitrogen, aromatic oxygen or aromatic sulfur atoms.

- b. Fragments containing selected atom types, i.e. nitrogen/oxygen in 3 or 4 membered rings, positively charged non-aromatic nitrogen and fragments with carbon-carbon triple bonds.

- c. Fragments with metalloid atoms (B, Si, As, Te, Ge and Sb).

The plots are shown in Figure 5. Well defined clusters for each class of fragments can be seen in each plot. The plots also show that t-SNE 2D projection preserves both local and global structures in the vector space. For example, aromatic nitrogen, oxygen and sulfur containing fragments are located together in a well-defined region, and aromatic oxygen and sulfur fragments form small localized clusters within this region (Figure 5a). Fragments with metalloids form well defined clusters very close to each other too. Similarly, fragments with N, S an O in 3/4 sized rings are all concentrated in a region, and N, S and O form separate individual clusters within.
Since physicochemical properties of molecules are functions of the contribution from the chemistry of their structural components, evaluating the distribution of such properties within the fragment vector space is valuable. Therefore, the fragments’ LogP and MR contributions were computed and mapped on t-SNE plots (Figure 6). Fragments with higher values of the respective property are colored red while fragments...
with lower values are colored blue. It seems that fragments with similar values tend to stay together and not randomly distributed.

To obtain further evidence, LogP of individual fragments were predicted using the computed LogP values of 5 of its closest neighbors in the 100-dimensional vector space; the results are shown in Figure 7a. A clear positive trend can be seen. It demonstrates that a significant amount of information is being captured from the contextual environment of fragments. For emphasis, an artificial fragment corpus was created by constructing fragment sentences by randomly picking fragments from the vocabulary (completely disregarding fragment contexts). After Word2Vec embedding, individual fragment’s LogP were predicted using 5 nearest fragments as before. The results are shown in Figure 7b. It is clear that the embedding process completely failed to capture any chemistry related information.

![t-SNE plots for 50,375 4-atom fragments highlighted with fragments with high (red) and low (blue) LogP and molar refractivity (MR).](image)

**Figure 6.** t-SNE plots for 50,375 4-atom fragments highlighted with fragments with high (red) and low (blue) LogP and molar refractivity (MR).
a. LogP prediction of fragments using vectors learnt from context based corpus.

b. LogP prediction of fragments using vectors learnt from randomized corpus.

Figure 7. Prediction of LogP contribution of individual fragments using computed LogP of its 5 closest fragments in the vector space. In plot (b), the predictions were random because the fragment corpus was artificially built using randomized sequences of fragments disregarding contextual information.

Use of fragment vectors in the modeling of biological properties of molecules (QSAR) was also explored. The results of classification of 6771 mutagenic and non-mutagenic molecules from Hansen and Bursi Ames mutagenicity benchmark datasets\textsuperscript{2,3} are reported here. As described in the Methods section, molecular vectors were computed using their 4-atom fragment vectors. 133 molecules had to be excluded because they did not have any 4-atom linear fragments due to small size or had fragments that were not part of the vector library of 50375 fragments. This left 6638 chemicals for use. Three additional commonly used fingerprints were computed for comparison: i. 1024-bit hashed fingerprints built from traditional ‘one-hot’ representation of 4-atom fragments, ii. 881-bit PubChem key binary fingerprints and iii. 58-bin fingerprints made of continuous molecular properties e.g. molecular weight, LogP, MR, vapor pressure, various surface descriptors and sum of E-State values on different atoms. For
visualization, the computed molecular vectors were subjected to t-SNE projection. Plots for the different fingerprints are shown in Figure 8 (mutagenic chemicals are shown in red and non-mutagenic in green).

![t-SNE plots using different types of molecular fingerprints](image)

- a. Distributed vector FP, 100 elements.
- b. Fragment Based Hashed FP, 1024 bits.
- c. PubChem Key FP, 881 bits
- d. FP with continuous properties, 58 elements

**Figure 8.** Mapping of Ames mutagenicity of compounds on t-SNE plots using different types of molecular fingerprints. ● Mutagenic ● Non-mutagenic.

It is evident from visual inspection that the traditional 1024 bit hashed fingerprints produce many small clusters than other fingerprints. This is expected because of the sparse ‘one-hot’ representations of fragments which do not have the ability to detect
similarity between fragments in a continuous manner. In addition, the separation between mutagenic and non-mutagenic chemicals is rather poor. The PubChem key based 881-bit fingerprints produce better separation between activity classes but suffer from similar issues as hashed fragment fingerprints, mainly because they are also based on sparse ‘one-hot’ fragment representations. The property based, 58-bin fingerprints did not give a clear visual separation of activity classes. On the other hand, distributed representation based fingerprints gave well defined clusters and improved separation of activity classes.

For a quantitative performance estimation, 10% out 10 times cross validations were conducted using k-NN method for every fingerprint type. Distributed 500 element fragment vectors are utilized at this stage for comparison. In every cycle of the validation process, 10% (~663) chemicals were taken out and their mutagenicity class was estimated from the labels of 5 nearest neighbors from the rest 90% (~5975) target chemicals. A query chemical was classified to be mutagenic if majority of the 5 neighbors are positive. As an additional condition, a neighbor was allowed to be counted only if it has equal or higher similarity than a set threshold. A prediction is excluded from consideration if equal number of positive and negative neighbors or no neighbors were returned. A series of 5 thresholds were tried for every experiment, a fixed random number seed was used to ensure reproducibility. The results are presented in Table 2. Sensitivity and specificity improves as the similarity threshold is increased, however, the coverage decreases. Coverage reduces rapidly for the binary bit based fingerprints, e.g. only 12.3 percent chemicals could be successfully predicted at 0.9 similarity threshold by the 1024-bit hashed fingerprints. On the other hand, distributed representation based fingerprints and 58-bin property based fingerprints both display excellent coverage across all threshold values. The distributed representation based fingerprints have much wider applicability domain than the traditional binary fingerprints. Overall, based on all three-performance metrics (sensitivity, specificity and coverage), the distributed representation based fingerprints and the 58-bin property based fingerprints seem to be the best. However, property based fingerprints depend on calculated molecular properties, e.g. LogP, MR, E-State
etc., therefore, their applicability depends on availability of parameters to calculate such properties. The distributed fingerprints, on the other hand, were obtained by unsupervised learning on ~3.7 million chemicals and have superior coverage, which can be easily increased by using more training chemicals.

Table 2. Performance of different fingerprint (FP) types in 10% 10 times cross validation to exercise for predicting mutagenicity using k-NN in molecular vector space.

| Similarity Threshold -> | 0.0 | 0.6 | 0.7 | 0.8 | 0.9 |
|-------------------------|-----|-----|-----|-----|-----|
|                         | Sensitivity\%, Specificity\%, Coverage% |
| Distributed vector FP, 100 elements | 81, 72, 100 | 81, 73, 100 | 81, 73, 99 | 82, 73, 97 | 85, 74, 85 |
| Distributed vector FP, 500 elements | 81, 74, 100 | 81, 74, 99 | 82, 74, 97 | 83, 75, 92 | 88, 74, 76 |
| Fragment based hashed FP, 1024 bits | 79, 74, 100 | 87, 74, 64 | 88, 73, 49 | 90, 77, 31 | 92, 81, 12 |
| PubChem key FP, 881 bits | 82, 75, 100 | 82, 75, 98 | 84, 76, 91 | 87, 77, 78 | 90, 77, 53 |
| FP with continuous properties, 58 elements | 78, 74, 100 | 78, 74, 100 | 78, 74, 100 | 79, 74, 98 | 83, 77, 86 |

Since structural alerts are crucial for interpretability of many QSAR outcomes, possible ways of using the distributed vectors to identify structural alerts in the query chemicals was investigated. Traditionally, a fingerprint based k-NN method can make a prediction but rarely can provide deeper insights such as structural alerts. However, the distributed representation based fingerprints offer a straightforward way to do so. Both fragment and molecule vectors contain the same number of elements and can co-exist in the same vector space. Consequently, the following steps were employed to find structural alerts in a query chemical:

1. The query chemical is broken down in its structural fragments.
2. The precomputed vectors for the fragments were obtained.
3. Each structural fragment was treated as a small query molecule and its activity class was predicted by applying k-NN method in the vector space of the target chemicals.

4. Fragments that return predominantly positive neighboring chemicals can be considered as structural alerts.

Two such examples are shown in Table 3 where alerts are identified in the query chemicals. This capability is very useful and much harder to achieve using traditional binary fingerprints or property based fingerprints.

| Table 3. Identification of mutagenicity alerts in two known mutagenic compounds using distributed fragment vectors to search nearest positive and negative mutagenic compounds in the compound vector space. Fragments that return mainly positive compounds were labeled as alerts. |
|---|
| Fragments that return predominantly positive neighboring chemicals can be considered as structural alerts. |

| Fragment | $P$ | $N$ | $P/(P+N)$ | Fragment | $P$ | $N$ | $P/(P+N)$ |
|----------|-----|-----|----------|----------|-----|-----|----------|
| N2-N3-C3H2-C3H2 | 5 | 0 | 1 | C3H2-[C3$\cdot$H$\cdot$]-[C3$\cdot$H2]-[O$\cdot$] | 5 | 0 | 1 |
| N2-N3-C3H2-C3H2 | 5 | 0 | 1 | C3H2-[C3$\cdot$H$\cdot$]-[O$\cdot$]-[C3$\cdot$H2] | 5 | 0 | 1 |
| N3-C3H2-C3H2-C2 | 5 | 0 | 1 | O=C3H2-[C3$\cdot$H$\cdot$]-[C3$\cdot$H2] | 5 | 0 | 1 |
| O=N2-N3-C3H2 | 5 | 0 | 1 | O=C3H2-[C3$\cdot$H$\cdot$]-[O$\cdot$] | 5 | 0 | 1 |
| O=N2-N3-C3H2 | 5 | 0 | 1 | C2-O-C3H2-[C3$\cdot$H$\cdot$] | 4 | 1 | 0.8 |
| C3H3-C3H2-C3H2-C3H2 | 1 | 4 | 0.2 | O=C2-C2-C3H3 | 2 | 3 | 0.4 |
| C3H2-N3-C3H2-C3H2 | 0 | 5 | 0 | O=C2-O-C3H2 | 2 | 3 | 0.4 |
| C3H2-N3-C3H2-C3H2 | 0 | 5 | 0 | C3H2-O-C2-C2 | 1 | 4 | 0.2 |
| N3-C3H2-C3H2-C3H2 | 0 | 5 | 0 | O=C2-C2=C2H2 | 1 | 4 | 0.2 |
| O=C2-C3H2-C3H2 | 0 | 5 | 0 | O=C2-C2=C2H2 | 0 | 5 | 0 |
| OH-C2-C3H2-C3H2 | 0 | 5 | 0 | O=C2-C2-C3H3 | 0 | 5 | 0 |

† Number of mutagenic neighbors among 5 closest chemicals to the fragment as measured by cosine similarity between the fragment and the molecular vectors.

§ Number of non-mutagenic neighbors among 5 closest chemicals to the fragment as measured by cosine similarity between the fragment and the molecular vectors.

**Conclusions**

A methodology is proposed for computing distributed, dense vector representations of molecular fragments. The proposed fragment embedding technique is based on
unsupervised machine learning method and requires only unlabeled chemical structures. The vectors captured meaningful physico-chemical properties, only from training on a corpus of instances of fragments in context without having access to any chemistry expert knowledge. Moreover, the vectors can be easily learned using publicly available software and data. The vectors need to be trained only once to be used in relevant applications afterwards.

The distributed fragment representation addressed some of the hard issues of traditional fragment representations - ability to compute meaningful similarities between fragments and to be able to combine fragment vectors in a meaningful way to compute molecular fingerprints without relying on hashing and random number generation. The addition operation combines features of individual fragments while preserving the feature identity of the component fragments. As a result, the molecular fingerprints developed in this study are able to model biological properties of molecules with an additional advantage of detecting structural alerts in test molecules.

It is also demonstrated that QSAR models built using distributed representation have wider applicability domains because the vectors encode chemistry information gleaned from a very large set of chemicals.

Last but not the least, distributed representations are particularly suited for t-SNE plots, enabling effective visualizations and exploration of chemical or fragment spaces.

References

1. PubChem Download Service, URL: https://pubchem.ncbi.nlm.nih.gov/pc_fetch/pc_fetch.cgi.

2. Hansen, K., Mika, S., Schroeter, T., Sutter, A., Laak, A. T., Steger-Hartmann, T., Heinrich, N. and Müller, K. R. (2009) Benchmark Data Set for in Silico Prediction of Ames Mutagenicity. J. Chem. Inf. Model. 49, 2077-2081.

3. Kazius, J., McGuire, R., Bursi, R. (2005) Derivation and validation of toxicophores for mutagenicity prediction. J. Med. Chem. 48, 312-320.
4. Ghose, A. K., Crippen, G. M. (1987) Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure-activity relationships. J. Chem. Inf. Comput. Sci. 27(1), 21-35.

5. Zhu, H., Sedykh, A., Chakravarti, S. K., Klopman, G. (2005) A new group contribution approach to the calculation of LogP. Current Computer-Aided Drug Design 1, 3-9.

6. Salum, L. B., Andricopulo, A. D. (2010) Fragment-based QSAR strategies in drug design. Expert Opin Drug Discov. 5(5), 405-412.

7. Rogers, D., Hahn, M. (2010) Extended-Connectivity Fingerprints. J. Chem. Inf. Model. 50, 742–754.

8. Riniker, S., Landrum, G. A. (2013) J. Cheininform. 5(1), 26.

9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013) Distributed representations of words and phrases and their compositionality. Proc. Advances in Neural Information Processing Systems. 26, 3111–3119.

10. Bengio. Y., Ducharme, R., Vincent, V., Jauvin, C. (2003) A Neural Probabilistic Language Model. Journal of Machine Learning Research. 3,1137–1155.

11. Collobert, R., Weston, J. (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. Proceedings of the 25th International Conference on Machine Learning (ICML-08), 160-167.

12. Mikolov, T., Chen, K. Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

13. Asgari, E., Mofrad, M. R. K. (2015) Continuous Distributed Representation of Biological. PLOS ONE, 10(11), 15.

14. Radim, R., Petr, S. (2010) Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. May, 45-50.

15. Krijthe, J. H. (2015). Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation, URL: https://github.com/jkrijthe/Rtsne.

16. van der Maaten, L. J. P., Hinton G. E. (2008) Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research. 9, 2579-2605.