Summer 2021

Identification of Chemical Structures and Substructures via Deep Q-Learning and Supervised Learning of FTIR Spectra

Joshua D. Ellis
Missouri State University, jde314@live.missouristate.edu

IDENTIFICATION OF CHEMICAL STRUCTURES AND SUBSTRUCTURES VIA DEEP Q-LEARNING AND SUPERVISED LEARNING OF FTIR SPECTRA

A Master's Thesis

Presented to

The Graduate College of

Missouri State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Natural and Applied Sciences, Computer Science

By

Joshua Ellis

July 2021

IDENTIFICATION OF CHEMICAL STRUCTURES AND SUBSTRUCTURES VIA DEEP Q-LEARNING AND SUPERVISED LEARNING OF FTIR SPECTRA

Computer Science

Missouri State University, July 2021

Master of Natural and Applied Sciences

Joshua Ellis

ABSTRACT

Fourier-transform infrared (FTIR) spectra of organic compounds can be used to compare and identify compounds. A mid-FTIR spectrum gives absorbance values of a compound over the 400-4000 cm^{-1} range. Spectral matching is the process of comparing the spectral signature of two or more compounds and returning a value for the similarity of the compounds based on how closely their spectra match. This process is commonly used to identify an unknown compound by searching for its spectrum's closes match in a database of known spectra. A major limitation of this process is that it can only be used to identify substances already in the database. An unknown compound not found in the database will likely match to a similar yet structurally different compound. Alternatively, FTIR has been used to identify characteristics, substructures, or functional groups of a compound based on the compounds IR spectral features. However, most works have only attempted to predict a limited set of substructures and there has only been limited success in predicting the full structure of an unknown compound based purely on its FTIR spectrum. For this thesis, I investigated the possibility of identifying compounds and identifying substructures present in the compound's structure by analyzing the compound's FTIR spectrum. This was dependent on the property that the infrared (IR) absorbances of a compound are the result of the physical interactions between bonded sets of atoms in the compound's structure. I hypothesized that different instances of the same substructures will either give similar spectral signatures or some pattern of spectral signatures that could be learned using machine learning. In this thesis I show that it is possible to use convolutional neural networks (CNN) to predict the presence or absence of substructures within a compound. Finally, I demonstrate a method of making predictions for the full structure of these compounds based on the substructure predictions and the compound's FTIR spectrum.

KEYWORDS: Fourier-transform infrared spectroscopy, chemistry, chemical structure, chemical substructures, deep learning, convolutional neural networks, evolutionary optimization, deep Q-learning, reinforcement learning

IDENTIFICATION OF CHEMICAL STRUCTURES AND SUBSTRUCTURES VIA DEEP Q-LEARNING AND SUPERVISED LEARNING OF FTIR SPECTRA

By

Joshua Ellis

A Master's Thesis
Submitted to the Graduate College
Of Missouri State University
In Partial Fulfillment of the Requirements
For the Degree of Master of Natural and Applied Sciences, Computer Science

July 2021

Approved:

Razib Iqbal, Ph.D., Committee Chair

Keiichi Yoshimatsu, Ph.D., Committee Member

Jamil Saquer, Ph.D., Committee Member

Julie Masterson, Ph.D., Dean of the Graduate College

In the interest of academic freedom and the principle of free speech, approval of this thesis indicates the format is acceptable and meets the academic criteria for the discipline as determined by the faculty that constitute the thesis committee. The content and views expressed in this thesis are those of the student-scholar and are not endorsed by Missouri State University, its Graduate College, or its employees.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Chemical compounds are made up of atoms which are connected by bonds. The number of atoms of each type and how those atoms are bonded together determine the properties of the compound. One of the methods to analyze the structure of a compound is infrared (IR) spectroscopy. The absorption of light at various wavelengths affects microscopic vibrations modes of chemical bonds between bonded atoms in a compound [1]. Fourier-transform infrared (FTIR) is a method which measures a compound's absorbance at various wavelengths. This method uses infrared light in the mid-IR range (400-4000 cm^{-1}) and provides an IR spectrum containing a distinct set of peaks and areas of absorbance [2]. These absorbances reflect the properties of chemical bonds between the atoms that are present within the structure of a compound. Adjacent sets of atoms form substructures within the greater structure. These substructures typically have a distinct spectral signature that can be observed by analyzing the FTIR absorbance peaks. In Figure 1 we can see the FTIR spectra from two heptanoic acid samples ($C_7H_{14}O_2$). The spectral features of these samples come from their C=O, H-O, C-C, and C-H bonds. These features can be observed in Figure 1. The majority of peaks which are frequently observed in FTIR spectra of organic compounds are shown in Figure 2.

Some substructures, such as carbonyl (C=O), are easier to identify than others because they produce large peaks in a consistent wavelength range. As shown in Figure 1 and elaborated in Figure 2, carbonyl and its variants strongly present absorbances in the 1600 to 1800 wavenumber range. Generally, one should be able to tell by looking at its spectrum whether a compound contains the carbonyl group in its structure. However, it is not always straight forward to extract further information such as which other atoms are connected to the carbonyl group. It is important to note that the peaks from different substructures could overlap with each other. Additionally, some substructures produce weaker absorption than others. For example, C#C (carbon triple bonded to carbon) produces a small peak, and it could overlap with C#N which produces peaks with variable intensity [3], [4].

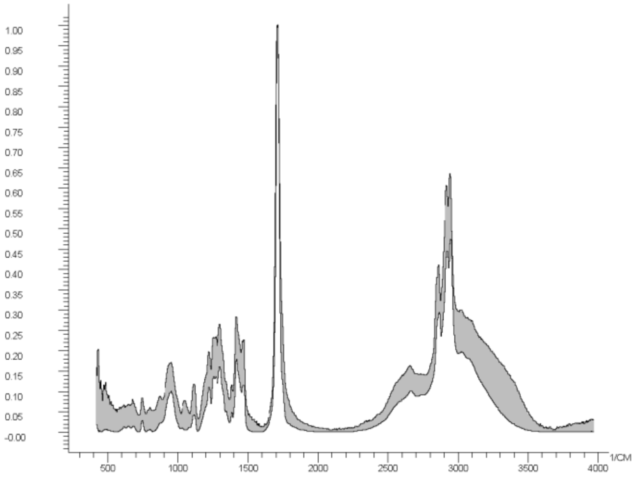


Figure 1. Two FTIR spectra of heptanoic acid.

While FTIR spectroscopy has widely been useful for spectral matching. However, two other methods - nuclear magnetic resonance (NMR) spectroscopy and mass spectrometry in analyzing the fine details of the structural makeup of a compound. NMR is particularly useful in terms of how thoroughly it can analyze the contents and structure of a compound [5]. However, the drawback of NMR is the size and expensive price of the instrument. Mass spectrometry can be manufactured in a much smaller sizes than NMR, but it only provides the information on the mass-to-charge rate of a compound and/or fragments of the compound [6]. FTIR spectrometer is even more cost effective and in certain cases more compact. However, FTIR spectroscopy has been utilized, only to a limited extent, in the structure determination of a compound [7]. The challenge is that, while there is a definitive relation between a compound’s structure and its IR spectrum, this relationship is rather complex. Furthermore, different FTIR scans of the same compound can produce slightly different sets of absorbances due to factors such as sample variance. This impreciseness also poses challenges in finding the correlation between the structure/substructure of compounds and spectral features.

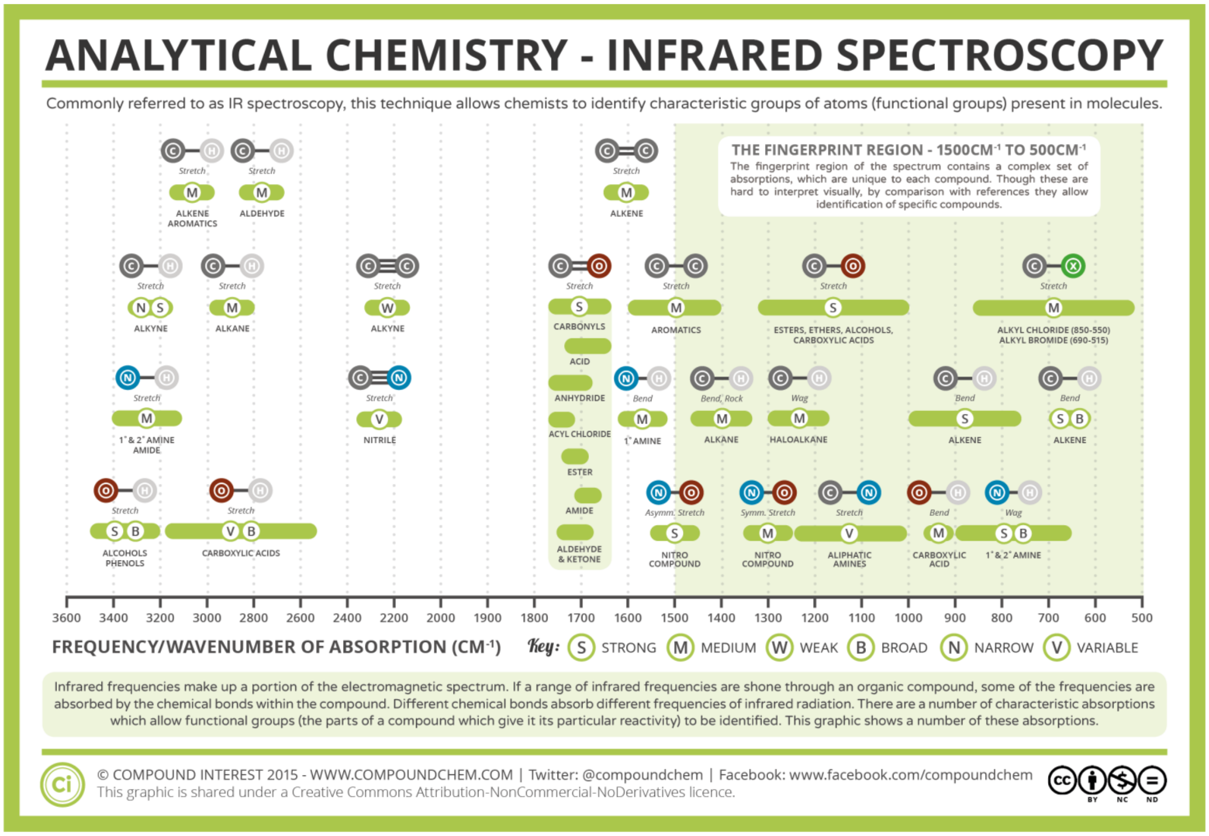


Figure 2. Diagram of substructures and the spectral ranges over which they present absorbance peaks [3].

Before discussing the substructures within a compound's structure in detail, I would like to describe the definition use of the term, substructure, in this paper. A substructure is any continuous subset of atoms and bonds within the structure of a compound. A substructure can branch or be a linear series of connections. Substructures can intersect or overlap. In this work, substructures were defined as a part of compounds consisting of at least 2 atoms connected by at least one bond. All bonds included in a substructure will also contain atoms from within the substructure at each of their 2 terminals. It will be assumed that an atom's unshared valence electrons for bonds will be used to form chemical bonds with other atoms outside of the substructure. For example, carbon, hydrogen, nitrogen, and oxygen have 4, 1, 3, and 2 valence electrons for forming covalent bonds. If a carbon atom has 2 valence electrons that are available for bond formation, the carbon atom could further connect to either 1 atom via a double bond or 2 different atoms via two single bonds.

While, substructures are allowed to contain overlapping atoms and bonds, another major challenge is that two different substructures could produce peaks that overlap each other. Both substructures can absorb light in similar wavelengths, resulting in the appearance of convoluted peaks. This means a smaller peak can be masked by a larger peak, effectively resulting in a loss of information. Identifying substructures in compounds will require a robust technique that can detect the spectral signs of the substructure while minimizing the negative effect of these issues. Currently FTIR spectroscopy is only used in limited capacities for analyzing chemical structure. FTIR spectrum is often used for spectral matching because two chemical samples with matching spectra have a high likeliness that they are also the same or similar substances. In this process, a database of spectra from known compounds is used to identify unknown compounds. Various algorithms such as Pearson correlation are used to match compounds and minimize inaccuracy due to variance in absorbances [2], [7]-[9]. However, the primary limitation of this approach is that for a compound to be matched the compound must already exist in the database. This means that previously unobserved compounds will not have an exact match in the database. The only information that can be gained with this method is looking at commonalities between compounds spectrally similar to this new compound. If there are common spectral features between two compounds, then it is likely there are also some meaningful structural similarities between them.

The hurdle in developing algorithms to identify substructures using FTIR spectra are that there are areas where more than one substructure's absorbances could appear in the similar spectral regions and the intensities of the substructure's absorbance peak could be weak, or in cases highly variable (Figure 2). This is especially true for substructures of size 2 (one atom bonded to another). However, it is reasonable to assume that other atoms that are connected to the substructures could, to some extent, influence the shape and/or intensity of the peaks. This suggests that it could be helpful to capture the spectral characteristics of at slightly larger substructures, such as substructures consisting of up to 4 atoms, because these structures would have more complex, prominent, and potentially more unique spectral signatures. Based on these information and considerations, we hypothesized that, by analyzing the spectral peaks from a compound's FTIR spectrum, it should be possible to predict substructures that are present within a compound and potentially predict the entire chemical structure of compounds.

In this work, I focused on applying deep learning, a powerful machine learning technique, to mapping the relationship between samples in two or more sets, for the FTIR spectrum-based prediction of the presence/absence of substructures within the compound. These methods are commonly used to learn the classifications of samples in a dataset. The structure of a neural network is a graph which contains parameters for the edge weights and node biases. In this learning process, a neural network starts with random parameters and over a number of training iterations, called epochs, these parameters are trained to improve the mapping of the input to the true output of the training set. Neural networks are exceptionally good at learning the data used to train them and therefore it is necessary to use a separate testing set to verify the accuracy of the network on new data.

Among different deep learning methods, I employed convolutional neural networks (CNN), a type of neural network which learn multiple kernels for each layer. Multi-layer perceptrons (MLP) contain edges connecting the nodes in each layer which is referred to as being densely connected. Conversely, CNNs are sparsely connected. The kernels in a CNNs layers are used to detect patterns in the previous layer. The first layers of a CNN will recognize simple patterns within the network's input. Subsequent layers will detect patterns within the previous layer's activations. As the network's input propagates through the network, the later layers will be able to detect much more complex patterns within the input data. The final layers of a CNN are typically an MLP. These final dense layers predict the class of the input sample based on the activations from the final CNN layer. This process is also called supervised learning because the network model's training is supervised by the loss between the model's output and the true class from the training data.

LITERATURE REVIEW

As mentioned earlier, FTIR spectroscopy have been utilized to identify substructures and structural features of a broad range of compounds. It is well understood that there is information about a substance that can be obtained from IR analysis. Kely et al. found that the octane rating of gasoline could be predicted by analyzing its IR spectrum. This is because a fuel's octane rating is dependent on the chemical makeup of the compounds in the fuel. Therefore, differences in fuel mixtures can be observed and analyzed [10]. Using similar principals, Soriano-Disla et al. found that many properties of soil can be determined with the use of IR [11]. Gosav, et al. have also reported successful classification of amphetamines using IR spectra and an MLP neural network [12].

Furthermore, neural networks and other machine learning algorithms such as support vector machines have been used to identify substructures within compounds. In 2005, Novic Marjana, and Jure Zupan investigated the use of Kohonen and counter propagation neural networks to analyze IR spectral features to recognize substructures. This method is similar to my proposal to use CNN's for identifying substructures. They were able to get a 33% average false positive rate over 34 substructures [13]. With more modern techniques and more powerful hardware it should be possible to achieve higher accuracy and classify more substructures than these past efforts. Jun Hong et al. applied a support vector machine (SVM) on a dataset with 823 samples. Out of 16 substructures, 12 were able to be classified with 90% accuracy and the other 4 with at least 80% accuracy [14].

In 2000, Markus Hemmer and Johann Gasteiger created a method to predict an unknown compound's structure using spectral similarity search and spectral modeling using a counterpropagation neural network in a system the authors call the STAR system (Search, Treatment and Adaptation of Radial Distribution Function Codes). When a spectrum is submitted to the STAR system, the system starts by searching for the most similar spectrum in a database of spectra. It is assumed that the submitted compound is structurally similar to this compound found in the database given that they are also spectrally similar because there tends to be a strong correlation between spectral similarity and structural similarity. The structure identified from the database is used as a starting point to generate the structure of the submitted unknown compound. This initial structure is stochastically modified over multiple iterations and with each iteration the neural network generates an artificial prediction of an IR spectrum based on the newly modified structure. If this new spectrum is closer to the submitted spectrum than any previously generated structures, then it is assumed this newly created structure is also closer to the true structure of the queried spectrum. This process stops when the STAR system is no longer able to generate new structures which have greater spectral similarity. The authors do not show any accuracy results for the project. They reported 6 examples of compounds identified by the STAR system, 5 positive examples and one example where the system identified a different yet similar chemical structure [15].

In 2020 Fine et al. used deep learning to train an autoencoder and a multi-layer perceptron (MLP) with FTIR and mass spectroscopy to predict the presence of various functional groups such as aromatic, hydroxyl, nitro, carboxylic acid, ether, alkyl, aldehyde, etc. With 7393 compounds in the dataset. Using FTIR spectroscopy only, their work was able to classify 9 of their 17 functional groups with at least an F1-score of 0.9, another 6 with an F1-score between 0.8 and 0.9, and the other two with F1-scores under 0.75. Combining FTIR and mass spectroscopy gave a mild improvement to their results but was largely similar to FTIR only [16].

In 2002, Neugebauer et al. showed that it was possible to simulate the spectra of molecules through quantum calculation. This simulation method was limited by the complexity of calculating the spectra of especially large molecules. Each additional atom in a compound's structure exponentially increases the time it takes to predict the compound's spectrum. Therefore, these methods have seen limited use due to the exploding complexity of calculating the spectra of large compounds [17].

An adjacent problem being researched is methods of simulating IR or FTIR spectra from a compound's structure. However, these calculations are very computationally expensive. Due to the difficulty in simulating larger structures, researchers are beginning to investigate methods of simulating IR spectra using deep learning. In 2019, Ghosh et al. used a deep tensor neural network to simulate molecular excitation spectra [18]. In 2020, Ye et al. used an MLP to simulate the IR spectra of protein structures [19]. In the same year, Kovács et al. used deep learning to simulate the IR spectra of interstellar polycyclic aromatic hydrocarbons [20]. These works show that deep learning is proving to be a viable replacement for quantum calculations and is enabling quick simulations of IR spectra from chemical structures.

Chemical structures can be represented in multiple forms. One of the more common and the method that will be used for this project is the SMILES format. A SMILES is represented by a string of letters, numbers, parentheses, brackets, and symbols. These characters represent the atoms in a structure and the bonds between atoms. Any chemical structure can be represented by a SMILES [21]. It is necessary for this project to create methods to represent chemical structures in a way that can be input into a neural network. M. Hirohara and K. Varmuza each propose similar methods for representing chemical structures using matrices. Varmuza's method referred to as a substructure isomorphism matrix is meant to be used more for structural similarity. It compares a set of queried substructures against a set of target structures. Each element of the matrix is given a binary value based on whether the query exists in the target [22]. Another method proposed by Hirohara uses 2D matrices to represent linear sections of a structure by representing the structure atoms as the columns and different SMILES characters as the rows. Therefor each column has a single 1 inserted into it on the row of the corresponding character. This is a simple method to input a structure or substructure into a network that does not require a 2D or 3D representation of a compound's structure [23].

DATASET

This project used a dataset of 5,297 unique FTIR spectra. Each spectrum is scanned from a compound with a unique chemical structure. The structures for each compound were represented by creating a doubly linked list of nodes and bonds based on the compounds SMILES string. The SMILES for the compounds were pulled from the PubChem website using an automated scraper which looked up each compound and copied the SMILES text. From these smiles I enumerated possible substructures and found 834 substructures up to size 5 with at least 10 examples in the dataset. The enumerated substructures contained no rings, but ring substructures were manually added to the substructure set.

Structures

The dataset was split into testing and training datasets using a genetic algorithm. This algorithm used genome which described whether each sample was in the training set or the testing set. This process targeted a 9:1 ratio between training and testing sets for each substructure. The algorithm was set to optimize the mean squared error between each substructures split and the targeted split ratio. This algorithm achieved a 0.2268% mean squared error when calculated by percentage in the training set and 5.2941 when calculated by the difference between targeted number of samples and final counts. Figure 3 shows the number of substructures which fall into positive count range. Some substructures had to be dropped because the most optimum split found left too few substructures in the testing set. This process resulted in a split with 4,767 samples in the training set and 530 samples in the testing set with 835 total substructures. An additional dataset was created from this dataset which will be referred to as the simplified dataset. The simplified dataset contains compounds with only hydrogen, oxygen, carbon, and nitrogen atoms. These structures are also restricted to contain between 5 and 25 atoms and not cyclic structures. This simplified dataset was created by dropping samples from the full dataset leaving it with 1,080 samples in its training set and 121 samples in its testing set. The simplified dataset uses 426 different substructures.

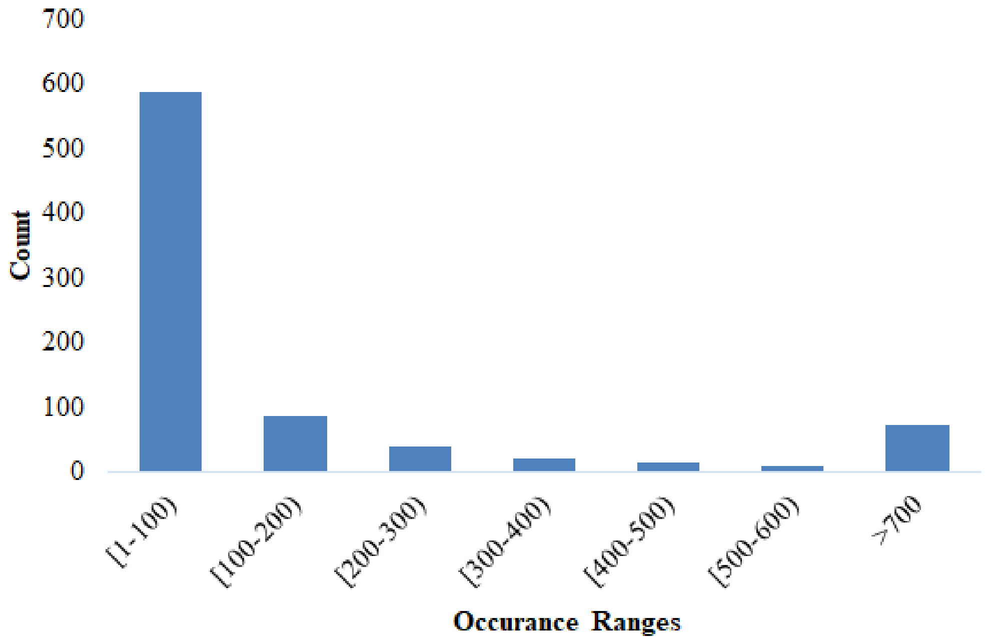


Figure 3: Histogram for number of samples which contain each substructure.

Substructures

The substructures analyzed in this project were chosen by permuting possible substructures and then searching the dataset for relevant structures which contained the substructure. Substructures are grouped into sizes where the size of the substructure refers to the number of atoms in the substructure. The process of permuting substructure starts with all substructures of size 1 found in the dataset. These atoms are then permuted by connecting to them each valid atom and bond pair to create new substructures of size 2. Any duplicate substructures created by this process are removed. I also removed any substructures which occurred in less than 10 samples in the dataset. The number is well below the number of samples required to accurately predict substructures. I still wanted to analyze the relationship between substructure count and prediction performance, so we chose an occurrence count requirement which balanced the need to observe infrequent substructures against the need to minimize the number of substructures observed.

In this project I identified substructures up to size 5. Substructure predictions appear to be relevant beyond size 5. However, each additional size contains far more substructures than the previous size. After manually adding in the aromatic and other ring structures the substructure dataset contained 864 total substructures. I limited the scope of this project to size 5 due to hardware and time limitations. Current hardware would be capable of using my methods on larger substructures, but this would have increased development time beyond this project's intended timeframe.

METHODS AND RESULTS

This project is broken up into three parts: the substructure prediction CNNs, greedy structure building method [24], and the deep Q-learning structure prediction network [25]. The first two methods are novel, and they are also necessary to implement the final structure prediction method. Each method was developed on the same FTIR dataset or on a subset of this dataset. Each of these methods will be accompanied by their results where abdicable.

Substructure Predictions

I chose CNNs to analyze the spectral patterns of the FTIR spectra in order to predict the substructures within compounds. The network needs to maximize the accuracy of its predictions for each substructure but using a multi-hot structure vector proved to be problematic. In this original method, each element in a vector referred to a specific substructure and would have a value of 1 if the substructure was present in the structure or 0 if it was not. This method suffered from significant bias towards substructures with a high occurrence rate and would effectively ignore substructures which occurred in few samples. I attempted to use sample and class weighting to balance the network's tendency to ignore infrequent substructures. However, I was unable to create an alternative method which worked as well as simply training a different network for each substructure. The process of tuning these networks, as will be discussed later in the Methods section, showed that an optimized network didn't require an excessive number of parameters. The substructure networks used in this project contain roughly 1.2 million parameters and their saved models require only 9.52 MB. This relatively small size made it reasonable to simply train a unique model for each substructure.

The substructure neural networks created for this project all use the same topology but different training configurations. The training hyperparameters are varied between substructures depending on the number of samples which contain the substructure. These networks are all CNNs which contain 4 convolutional layers. The layers respectively have 53, 140, 23, and 160 filters. Each of the convolutional layers feed into a batch normalization layer, next a leaky Relu layer which uses an alpha value of E-3.5839, and finally a max pooling layer which use a 2x1 pool size. The exception is the final convolutional layer which uses valid padding and skips the max pool layer. The output of this final convolutional layer is flattened and fed into a dense layer with 95 nodes which uses the same batch normalization and leaky Relu activation as the convolutional layers. The input to the network is 600x3x1 and the outputs of each hidden layer bock are 300x3x53, 150x3x140, 75x3x23, 73x1x160, and 95. The network's output is a simple perceptron with one node which uses sigmoid activation. The topology of this network can be seen in Figure 4. The hyperparameters listed denoting number of filters and dense nodes were found through an optimization process which used genetic algorithm. The genetic optimization process also determined the leaky Relu alpha value and additional hyperparameters which will be listed in the next subsection.

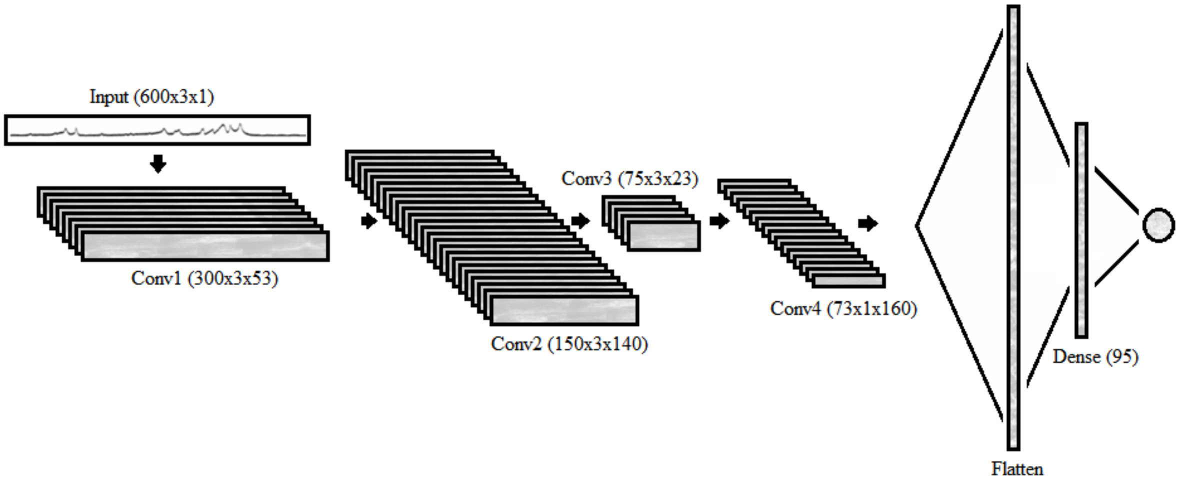


Figure 4: Topology of the proposed substructure network

Genetic Optimization. To optimize the proposed networks, I turned to evolutionary algorithms [26]. Changes were made to the typical evolutionary optimization process in order to independently optimize the hyperparameters for different substructures. Exploratory investigations seemed to indicate that the network's optimal hyperparameters varied depending on the ratio of positive to negatives samples for the given substructure. To create optimal networks for all substructures regardless of the number of positive samples, the genetic optimization process evolved curves based on polynomials. The genes for these curves were lists of the constant coefficients for these polynomials and take the form:

$Q_1(x, g) = g_1 * (x + g_2)^3 + g_3 * (x + g_4)^2 + g_5 * (x + g_6) + g_7$, where $g \in \mathbb{R}$.

Some curves require two inputs and take the form:

$Q_2(x, y, g) = g_1 * (x + g_2)^3 + g_3 * (x + g_4)^2 + g_5 * (x + g_6) + g_7 * (y + g_8)^3 + g_9 * (y + g_{10})^2 + g_{11} * (y + g_{12}) + g_{13}$.

Figure 5 shows the curves for the 1 input curves and Figure 6 the 2 input learning rate curves. The network also uses constant values which are similarly evolved using this process. This genetic optimization process used 10 folds of the full dataset with each substructure getting an independent set of folds. For this process I chose 10 substructures with different occurrence rates [CC(-O)N, CC≡CC, C=CSO, CC(=C)F, CN(-C)C, CC(=O)C, CC=CN, C=CCO, CC(=C)C, CC=CC]. This process minimized 2 objectives: The first objective is to maximize the f1-score of the networks results averaged over 10 folds and across the 10 substructures. The second objective is to minimize the variance of the network over the 10 folds. During the optimization process, any network that is Pareto efficient is given the highest fitness value. Over several recursive iterations, multiple fronts of Pareto efficient networks are grouped together where the networks in each subsequent group is dominated by the networks in the previous groups, co-dominant with networks in the same group, and dominates one or more networks in subsequent groups. Networks in the most efficient group are given the highest rank and each less efficient group is given a lower rank. Network ranks are recalculated each time a new network is trained. Each time a new network is added to the population, one of the lowest ranked networks is dropped. In the case that there is more than one network in the lowest rank, the choice of which network to drop is made at random from the lowest ranked group. The genetic optimization process uses two-point-crossover for each set of genes within the genome and standard Gaussian mutation. This optimization process required a total of 2139 unique sets of hyperparameters to be tested before the population converged. The evolved curves can be seen in Figures 5 and 6.

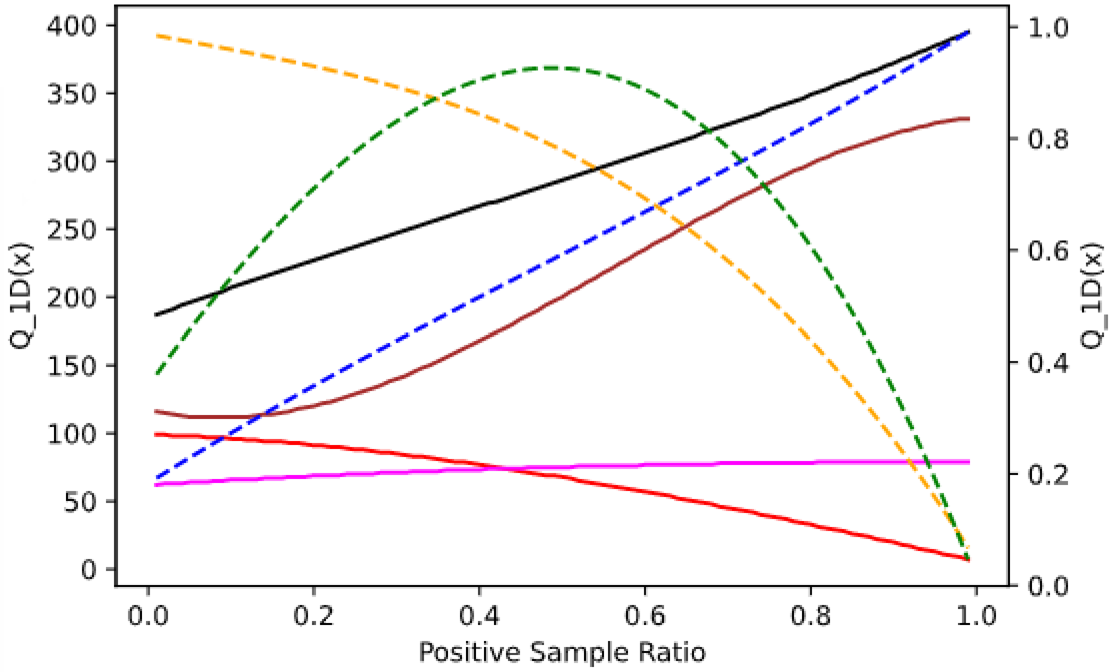


Figure 5: Curves for batch size (brown), number of epochs (black), learning rate step decay (blue), momentum (green), class weights (orange), patience (red), patience start epoch (magenta). Dashed lines use the number scale to the right of the graph and solid lines use the scale on the left.

Hyperparameters. The training process for the networks use a training process which varies based on the ratio of samples belonging to the positive class. This ratio will be referred to as *r.* The learning rate, batch size, number of epochs, learning rate step decay, momentum, class weights, patience, patience start epoch each vary in this way. The training process occurs over a number of epochs which ranges between 187 at *r*=0.01 and 395 at *r*=0.99. Optimum batch size also varied as *r* changes with the minimum batch size of 112 occurring at *r*=0.05 and maximum of 331 at *r*=0.98. Early stopping is used with a variable patience value ranging from 7 at *r*=0.99 to 99 at *r*=0.01. The early stopping function is skipped for the first few epochs ensuring the training process runs for at least 62 epochs at *r*=0.01 and 79 at *r*=0.81. The learning process uses the stochastic gradient descent (SGD) optimizer with variable momentum which reaches a maximum of 0.9846 occurs at *r*=0.49 and decreases as *r* increases and decreases. The training process weights the positive and negative classes based on a similar curve. The positive class uses the formulas $ClassWeight_1 = Q_{1D}(r, g_{ClassWeight})$ and $ClassWeight_0 = Q_{1D}(1 - r, g_{ClassWeight})$. This curve ranges from 0.0672 at *r*=0.99 to 0.9846 at r=0.01. The full list of hyperparameter curves can be found in Appendix A.

The training process uses the concept of cyclical learning rates [27]. The networks learning rate is decayed over the length of the training process, decayed after every 23 steps, and a cyclical learning rate is repeated every 23 epochs. The cycle/step length of 23 was also found through the evolution process. The network’s learning rate (LR) is updated each epoch using the formula:

$$LR(epoch) = ShortLR\left(\frac{epoch\,\%\,CycleLen}{CycleLen}, r, g_{ShortLR}\right) * LongLR\left(\frac{epoch}{Epochs}, r, g_{LongLR}\right) * e^{\left\lfloor\frac{epoch}{CycleLen}\right\rfloor} * \mathrm{LRStepDecay}(\mathrm{r}, \mathrm{g}_{\mathrm{LRStepDecay}}).$$

Cyclical learning rates have been shown to be useful in maximizing the accuracy or F1-Scores of neural networks. Learning rate decay is also commonly used when implementing variable learning rates. The curves shown in Figure 6 show both the cyclical variation of the learning rate and the long-term decay of the learning rate. The evolution process was given the ability to utilize and tune cyclical learning rates, long term decay, and stepped decay. This allowed the optimization process to tune or even suppress these methods as needed ensuring the network was able to optimally utilize these strategies. A graph of the final learning rate can be seen below in Figure 7.

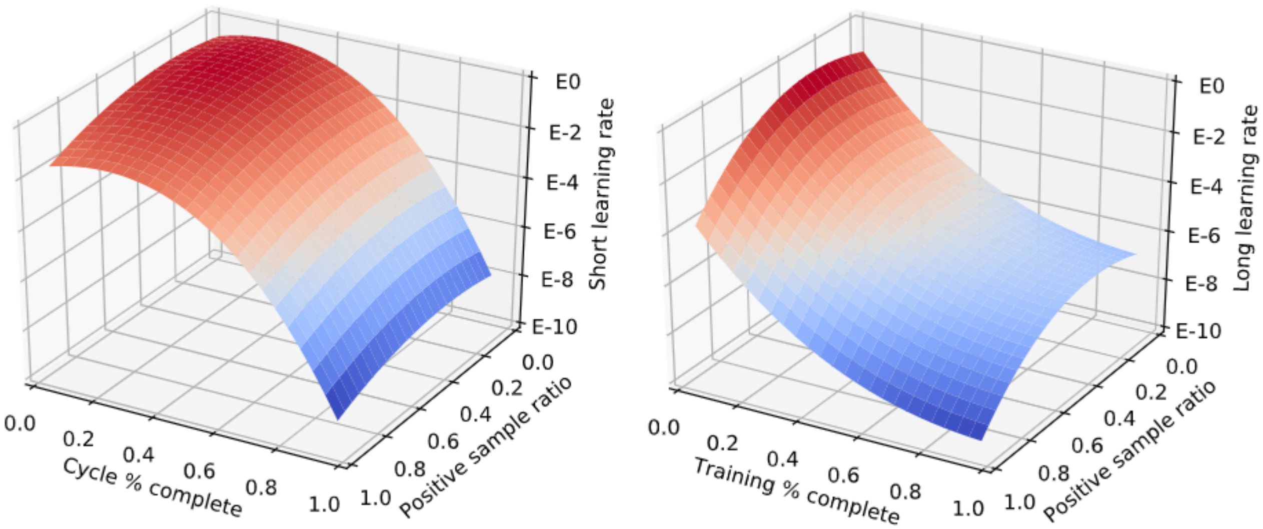


Figure 6: 2D curves used for the learning rate of the proposed network. Short cycle learning rate (left) and long cycle learning rate (right) are based on the positive sample ratio and the current epoch number.

Substructure Encoding. Some substructures have so few examples that even with hyperparameter tuning their predictions are too inaccurate to be useful. One way I mitigated this issue was by taking the networks of substructures with sufficient examples in the dataset and repurposing them as encoders. For this I removed the perceptron at the networks output and concatenated the network predictions for each sample. I then applied a 99% principal component analysis (PCA) dimensionality reduction to these predictions which reduced this combined size to 2219 datapoints. A new MLP was trained for the substructures on these new spectral encodings. This method resulted in moderately better performance for the majority of substructures with less than 700 examples, but also tended to either decrease accuracy or make little difference for samples with more than 700 samples. Therefore, it was simplest to allow substructures with more than 700 examples to use their original predictions and for other substructures to use this encoding method. Another motivation to create these spectral encodings was for use in predicting the compound's full structure as will be shown later in the paper.

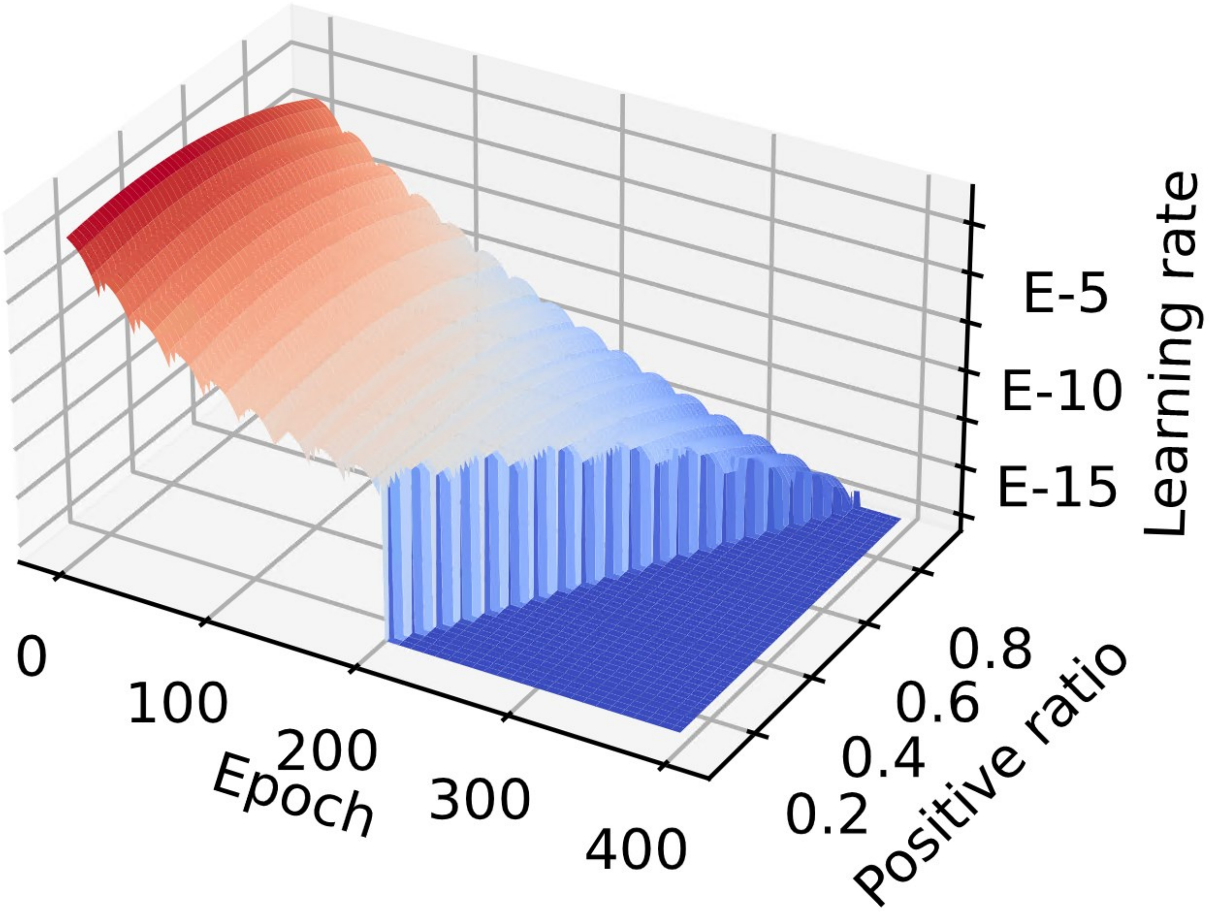


Figure 7: The network's learning rate over the epochs by positive sample ratio.

Substructure Results. Figure 8 shows the F1-scores of substructures in the dataset. While many of these substructures show low performance, recall Figure 3 showed that approximately 70% of the substructures occurred in less than 2% of samples. While the structures that occur infrequently do give poor results, many of the 834 substructures can be predicted with high recall and precision. The primary factor in accurately prediction a given substructure is having a sufficient number of examples of the substructure. A full list of substructures and their F1-scores can be found in Appendix B.

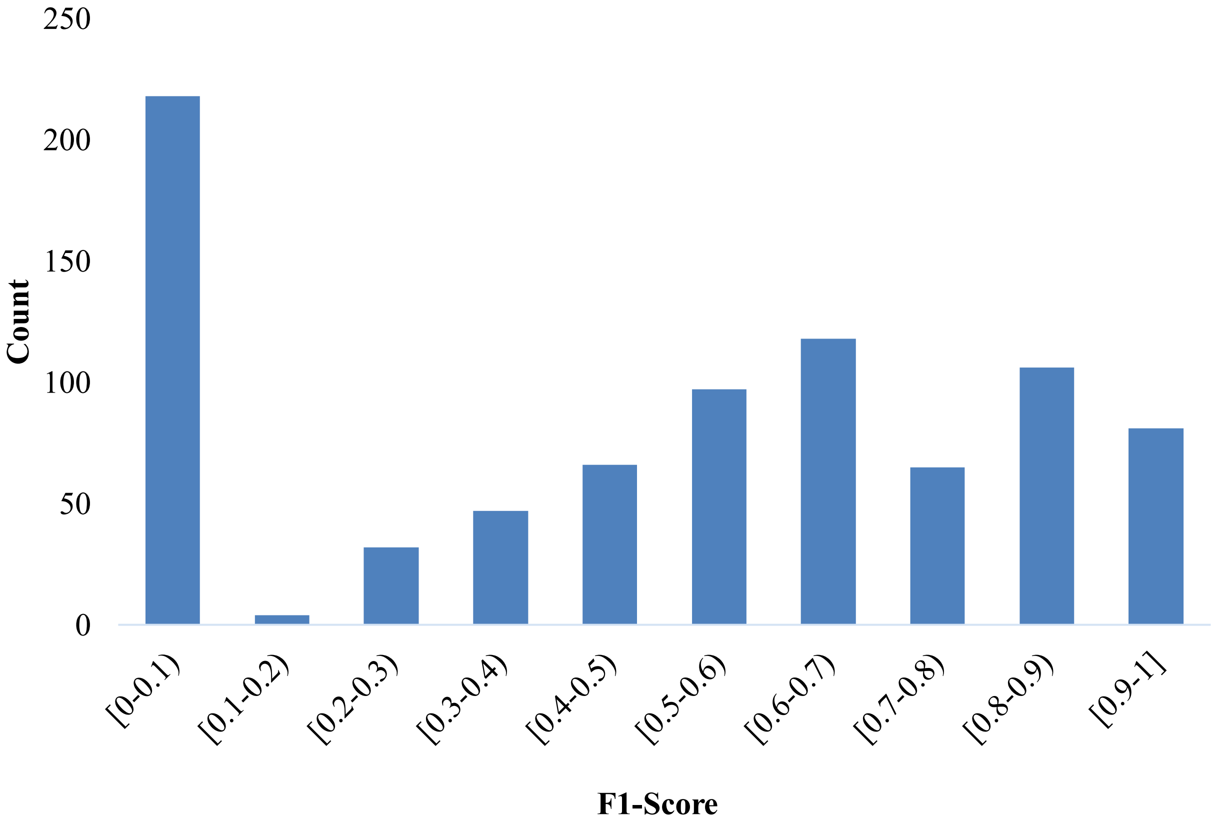


Figure 8: F1-scores for substructures in the dataset.

Figure 9 shows a histogram the Jaccard scores for samples in the dataset when compared to the base truth. Each score is based on the similarity to the prediction's substructures to the base truth's substructures. Of the 530 test samples, 363 were predicted with a similarity of over 0.6, 201 with a similarity over 0.8, and 104 with a similarity over 0.9.

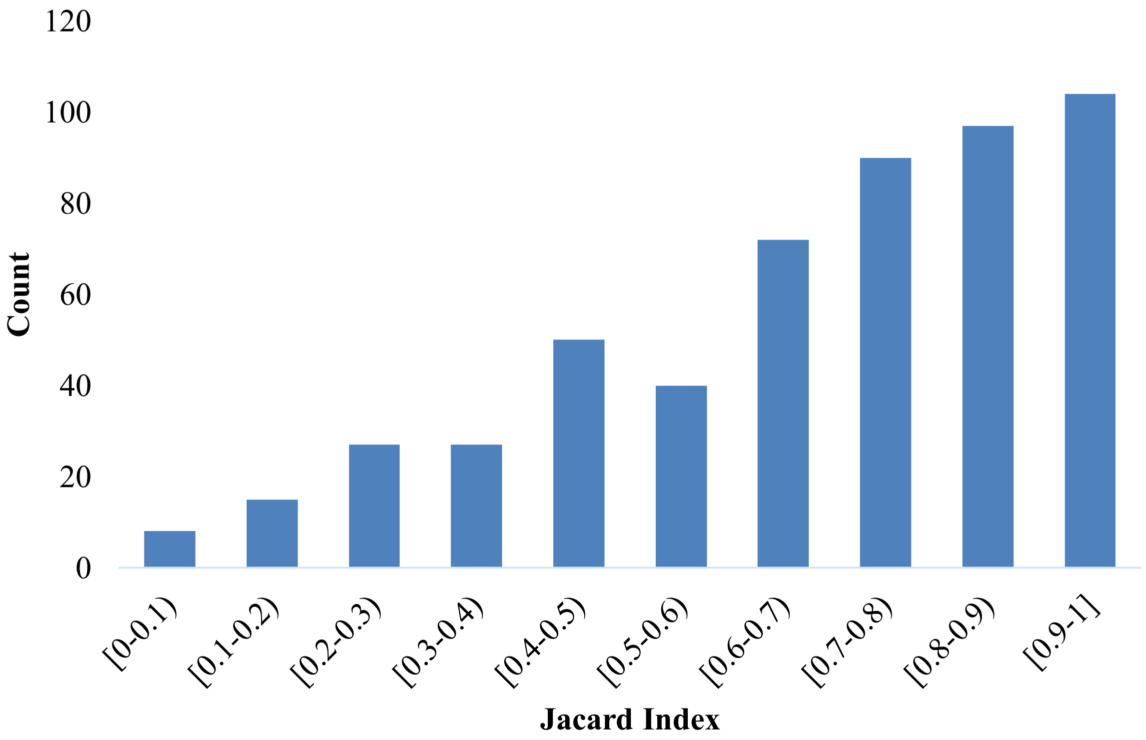


Figure 9: Jaccard index scores for samples in the dataset.

Figure 10 shows that high accuracy is possible with enough examples of each substructure. Therefore, samples which are comprised of infrequently occurring substructures should be expected to also have low Jaccard index scores. This suggests that the largest limitation of this project is the small number of samples in the dataset. Increasing the number of samples in the training set could have a significantly positive impact on the results of these methods.

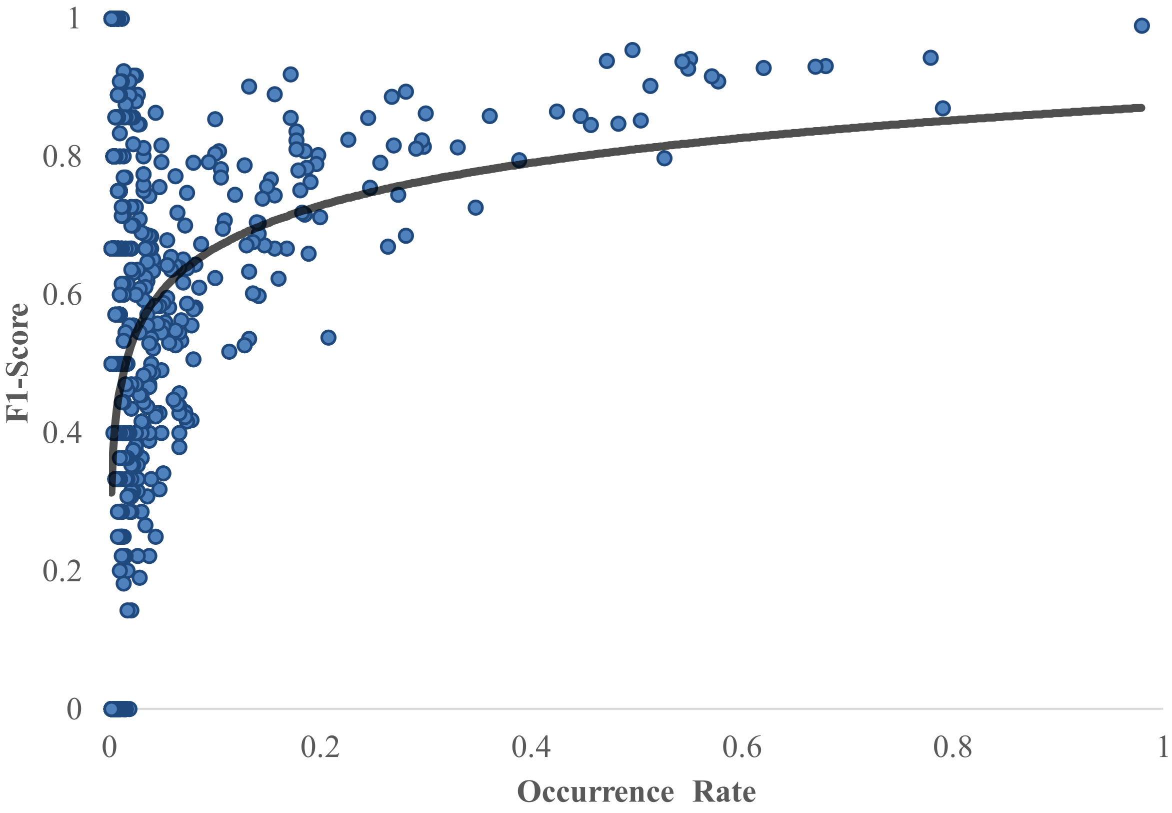


Figure 10: Plot of substructure occurrence rates vs. their resulting F1-Score.

Greedy Structure Prediction

The final goal of this project is to predict the true structure of a compound from its FTIR spectra. Initial investigations showed that a greedy algorithm [24] could almost reconstruct a structure from perfect substructure predictions. However, this method also had a number of drawbacks. The primary limitation of this method was that it required the true substructures to function correctly and performed poorly when given the predictions from the substructure networks. The other issue was that there is some ambiguity between substructure predictions and the compound's true structure. Greedily choosing a modification that improves substructure similarity is not guaranteed to create a structure that optimizes this similarity score. Furthermore, two different structures can have the same substructures.

At the core of this method was an evaluation function that simply found all substructures in a predicted structure and compared this to the prediction. The predicted structure would then be given a score based on the similarity between the predicted structure's substructures and the true structure's substructure predictions. This method started with a single carbon atom. Then it enumerated each possible atom bond pair that could be connected to the structure and each pair of atoms in the structure that could be bonded together. Each of these possible modifications to the structure was evaluated and compared to the current structure evaluation. The algorithm would greedily choose the best modification with each iteration of this process which would grow the predicted structure based on the substructure predictions. This method would stop modifying the structure once all proposed modifications were given lower evaluations than the current structure. At this point the function would return its prediction for the structure. A diagram of this process can be seen below in Figure 11.

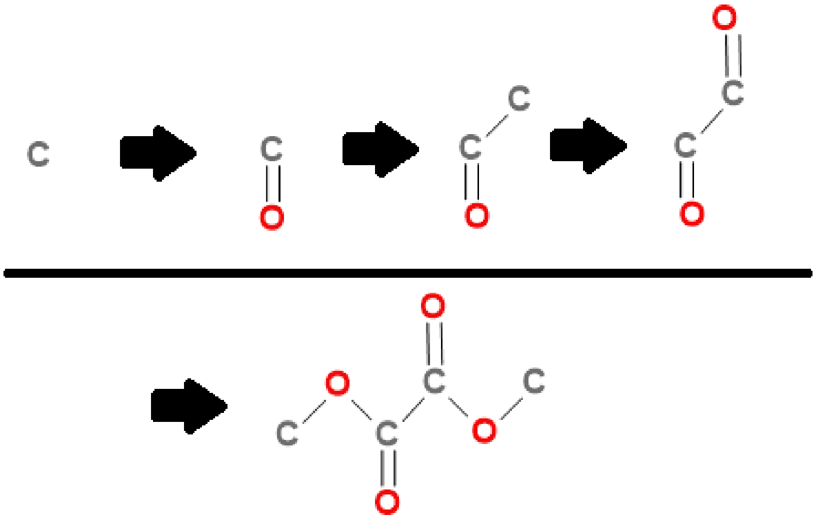


Figure 11: Iterative creation of Dimethyl oxalate's structure. This figure shows the proposed process of iteratively adding atoms and bonds in order to create a prediction for the compound's full structure.

Evaluation method. The greedy structure prediction method bases the value of its predictions on an evaluation function which gives structures a score between 0 and 1 based on the structure's similarity to the compound's true structure. This score is based on three metrics: First is the cosine similarity between substructures found in the true structure and the predicted structure. The Jaccard index and f1-score were also tested in place of cosine similarity. The f1 score resulted in similar final predictions to cosine similarity, but the Jaccard index resulted in the creation of highly dissimilar structures. Second is the similarity between atom bond paints found in the true and predicted structures, which uses the formula:

$$1 - \frac{\sum |card(true\_stucts[ab]) - card(pred\_structs[ab])|}{\sum card(true\_stucts[ab]) + \sum card(pred\_structs[ab])}, a \in \{C, O, N\}, b \in \{-, =, \equiv\}.$$

The third is the similarity between the structure's extended connectivity fingerprint with radius 6 (ECFP6) [28] which is compared with the Jaccard/Tanimoto index [29]. The evaluation function uses the harmonic mean of these three values. This evaluation method will be referred to as the Structure Based Similarity Score (SBSS).

Greedy prediction results. Figure 12 and 13 show that this greedy algorithm can at least partially recreate structures from perfect predictions. Smaller structures tend to work better than larger structures. The larger the structure is the more ambiguity there is in how substructures could be put together. Figure 13 also contains mirrored structures which further complicates the problem.

Figure 14 shows that the greedy algorithm has issues with ring substructures. In the future it will be necessary to find a method for correctly piecing together these ring-based structure. However, for now this project will simply limit the dataset by removing samples with ring substructures.

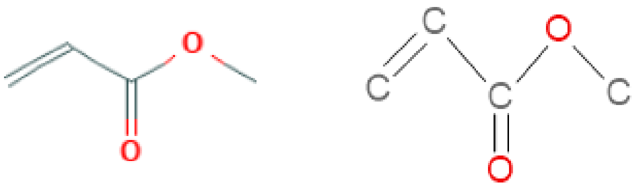


Figure 12: Methyl Acrylate truth (left) [30] and prediction (right) created from using the greedy prediction method using the base truth structure.

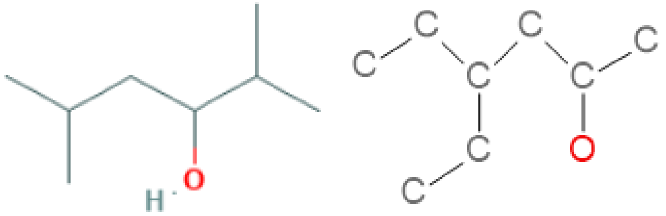


Figure 13: 2,5-Dimethyl-3-hexanol truth (left) [30] and prediction (right) created from using the greedy prediction method using the base truth structure.

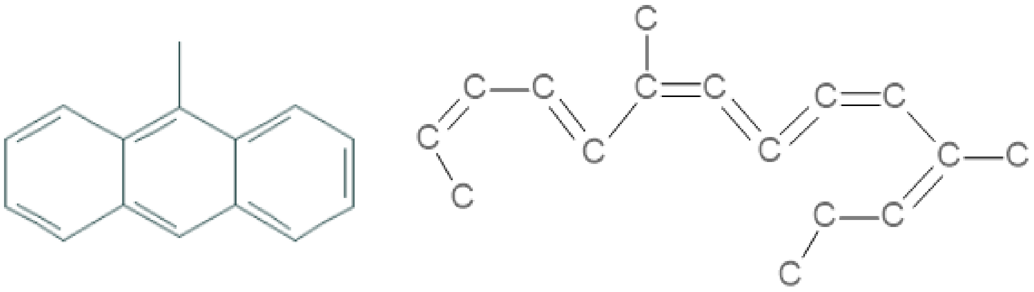


Figure 14: 9-Methylanthracene truth (left) [30] and prediction (right) created from using the greedy prediction method using the base truth structure.

Deep Q-Learning for Structure Prediction

As stated earlier, the greedy prediction method had a pair of flaws. It could only work with perfectly accurate predictions and it was not guaranteed to create a perfectly matching structure. To deal with these issues I turned to deep Q-learning. Q-learning uses greedy epsilon learning to make random predictions for the value of actions at various states. The algorithm's epsilon value controls the probability that the algorithm takes a random action or takes the action with the highest predicted value. A Q-table is updated at the end of each iteration. This table represents the future value of each action in each state. Deep Q-learning replaced the Q-table with deep learning model. This model is trained on the value of entering into new states based on the current state.

Deep Q-learning. Deep Q-learning is a form of unsupervised reinforcement learning which is based on Q-learning [25]. Q-learning involves the creation of a Q-table which gives the values for various states and actions. The Q-table acts as a model to dictate the actions of a process over a time series. This table could learn the actions required to solve a puzzle or be used in learning similar problems. In this process, epsilon-greedy learning is used to make take random actions and the Q-table learns the value of entering into these states. Q-learning decays the epsilon value over time. The epsilon value determines the rate at which the next state will be chosen at random versus choosing the next state by taking the action with the highest value from the Q-table. The Q-learning process seeks to maximize some evaluation function by estimating the future values of states as evaluated by this evaluation function. Q-learning learns not just the value of various states but also assigns values to each state based on the current value of the state and the possible maximum future value of the state. At each step in Q-learning, the value of each action at each state is updated using the formula:

$$Q(s_t, a_t) = \alpha[r_t + \gamma * \max\left(\mathrm{Q}(\mathrm{s}_{t+1}, a)\right) - Q(s_t, a_t)].$$

Deep Q-learning replaced the Q-table with a neural network model. Here the network uses its own predicted rewards for subsequent states to estimate the value of the current state. To stabilize the learning process a second copy of the model is used to predict future rewards. This second target model is updated with the main model's weights and biases on some regular number of training steps. The loss function used in deep Q-learning is:

$$loss = Huber\left(Q(s_t, a_t),\, r_t + \gamma * \max\left(\mathrm{Q}_{target}(\mathrm{s}_{t+1}, a)\right)\right)$$

Where Q-target is a copy of the Q-model which is updated on a less frequent interval. Huber loss is considered to provide for more stable convergence. Having the model estimate the weighted sum of the immediate state reward and its own prediction for the action's future value creates some instability in the learning process. As γ approaches 1 the training process becomes more unstable. Values for γ greater than or equal to 1 are inherently unstable and will not converge. One of the benefits of deep Q-learning is that it can generalize problems and be used to evaluate states that are independent from those seen during the training process. This also means a deep Q-model may have some ability to solve problems similar to those it was trained on.

The deep Q-learning training process often uses two models. One is the Q-model, and the other is Q-target which is a copy of the Q-model. In the training process, the Q-model is taught to learn the past rewards plus the future rewards which is based on its own predictions. So, the Q-model's predictions are dependent on the future rewards as predicted by a past version of itself. This means deep Q-learning needs to converge in two ways. First the Q and Q-target models need to converge with each other. Second the Q-learning process needs to explore the possibility space thoroughly enough that it can learn to solve the given problem. In this training process, the Q-model could be considered to be building its own dataset on the fly. At each step the Q-model predicts the best action based on its understanding of the problem. If the chosen action is in fact not the true best action, then the new state will either have a lower reward or lead to a lower reward in the future. By finding the situations where the model's expectations for future rewards does not reflect the true value of the action, the model is able to actively find flaws in its comprehension of the problem.

Deep Q-learning Prediction Method. My project required some reinterpretation of the Deep Q-learning method. The biggest obstacle was that my project required this model to be able to predict the structure of as yet unseen compounds. This means I would need to evaluate this model on sample from the testing set while training it on compounds in the training set. To do this I would need to give the model any information I could about the nature of the compound whose structure it was trying to reconstruct. I designed a model that would take three inputs: First it would take a matrix representing features of the predicted compound's structure. Second it would take the predictions of substructures within the structure. Third it would take the encoded spectra discussed in the previous section. The purpose for this design was to create a model that could interpret the relationship between a compound's spectrum, substructures, and structure.

This model was used to predict structures by piecing the structure together over multiple steps. Similar to how the greedy method worked, this new deep Q-learning method incrementally modified the predicted structure based on its predictions for the future value of each possible prediction. Each iteratively created prediction is evaluated and the network is trained using the following loss function:

$$loss = \left|Q_{model}(struct, predictions, spec\_encoding) - [\,reward(next\_struct) + gamma\ * Q_{target}(next\_struct, predictions, spec\_encoding)]\right|.$$

Q-learning predicts the future value of moving from one state to another. In this loss function the partially built structures are the states. Each partial structure that is built during the training process acts as a new training example for the model. In deep Q-learning, the model iteratively learns to output a value that is more similar to the current rewards for the state it left plus some percentage of the models' predictions for the future rewards of the new state. Here g*amma* is a parameter with a value between 0 and 1 which weights the importance of predicted future rewards to the immediate value of the new state.

In deep Q-learning the Q-model predicts the approximate future value of various states. My implementation of this method attempts to predict the future value of various states based on the networks understanding of the current problem. Each compound in the dataset can be thought of as a unique problem which the model is attempting to solve. The model needs to understand the how to build all the structures for these compounds and be able to generalize its understanding well enough to predict the structures of compounds it has never seen before. Furthermore, it needs to be able to do this while accounting for the noisy nature of FTIR spectra and the inaccuracy of substructure predictions.

With the goal of training a model which understands the potential inaccuracies this model would likely face in real world scenarios, I intentionally induced random false positive and false negatives into the substructure prediction based on the average rates from samples in the testing set by sample occurrence rate. Each time a structure is reset to its base state, new random inaccuracies are induced into the predictions. The substructure predictions and spectra encodings are also masked by Gaussian noise with a standard deviation of 0.1. The reason for these normalization techniques is to encourage the model to generalize its understanding of the problem by learning to compensate for the inaccuracies and noise which could appear in compounds.

Structure prediction network. The structures predicted in this process need to be represented in a form that can be understood by a neural network. I chose a vector which represents linear substructures of up to size 5. This algorithm creates a vector whose elements each represent a different unique permutation of carbon, oxygen, and nitrogen atoms and single, double, and triple bonds. An autoencoder was trained on the training set to reduce the dimensionality of the representations to 512 floating point values.

The structure prediction network, shown below in Figure 15, predicts the estimated value of entering new states based on the current state and game. Here the game is the structure it is trying to build. The game is represented by the substructure network predictions and the spectral encoding. The current state is represented by an encoding of the current structure, newest modification to the current structure, and the current structure's fingerprint. The substructure predictions and spectral encoding are there to inform the network about the possible structure features of the compound based on its FTIR spectrum. This network is trained using deep Q-learning, a method that is often used for learning in games such as chess. Here the game the networks plays is building a structure which best matches the spectrum encoding and substructure predictions.

The structure prediction network pieces the structure together one step at a time. The network is initially given a structure with only a single carbon atom, then it begins building out from this atom. Based on the current structure state, a simple algorithm permutes new possible structures by finding each new atom bond pair that could be added to the structure. Each of these possible modifications are evaluated by the network and the new state with the highest value is chosen. This process iteratively builds the structure of the compound until the network choses to stop.

Figure 15 shows the structure of the network. Each input to the network is given a separate multilayer perceptron (MLP) with two layers of size 1024 or size 512. The output of these sub-MLPs are concatenated and another three dense layers further interpret the inputs. The final layer of the network gives two outputs. One is the value of entering into the new state, which is referred to as the Deep Learning Based Score (DLBS). The other is how similar the network believes the new structure's size is to the size of the true structure, called the Deep Learning Completion Score (DLCS). When the DLCS value is greater than 1, the network stops building the structure. During the training process the network is trained on the true values of its predictions at each step, both the true SBSS and the TCS.

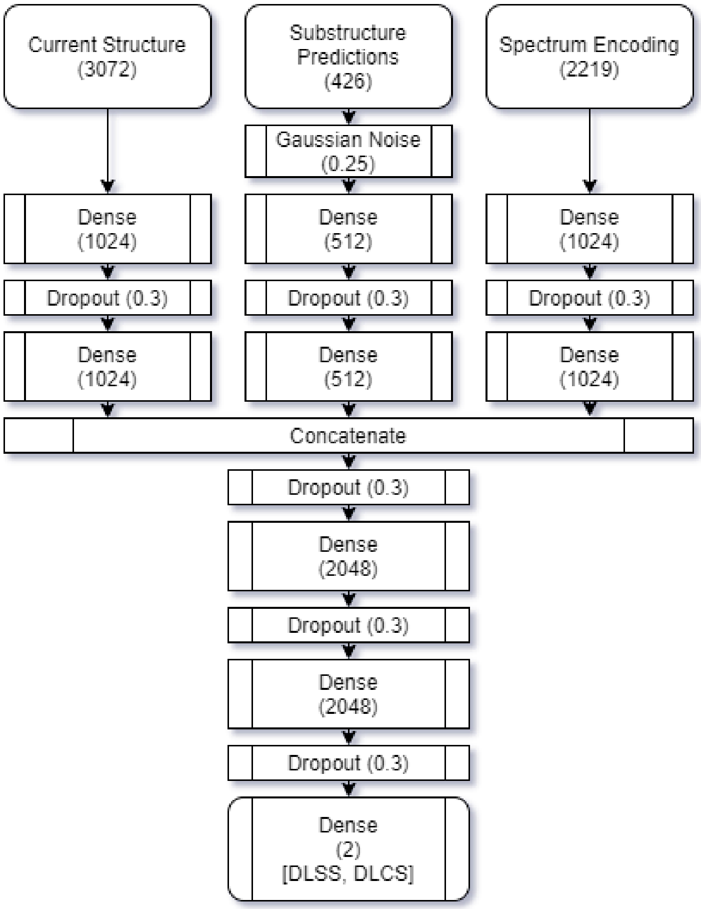


Figure 15: Topology for structure prediction network.

Training the structure predictions network. The structure prediction network was trained on the training set using greedy epsilon learning. The model was given a base truth dataset to learn from which contained the true structures of the compounds in the training set, pseudo-partial predictions created by removing atoms one at a time from true structures, and the last 350,000 partial structure predictions created by the model. Training used Huber loss with a learning rate of 10^{-4}. The primary Q-model would be updated every 4 frames and the target Q-model would be updated every 32 frames. Over the first 128 training iterations, random updates to structures were used. This is part of the greedy-epsilon Q-learning method. After these 128 iterations the epsilon value began to decay from 1.0 to 0.1 over the next 384 iterations. Figure 16 shows the decay of ε. The training process is stopped when loss reduction slows as shown in Figure 17.

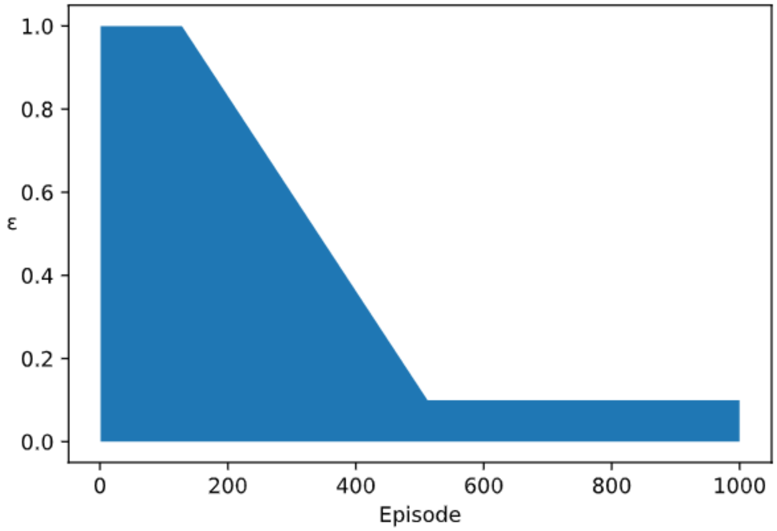


Figure 16: Decay curve of epsilon value.

In each training iteration, 303 different compounds were permuted and their new partial predictions evaluated. When a structure was completed, a new compound would replace it and begin the process from the initial state of a single carbon atom. Before each structure permutation and evaluation, a random number alpha whose value is between 0 and 1 would be created and this number would be compared to epsilon. If $\alpha < \varepsilon$ then a new random permutation would be chosen, if $\alpha > \varepsilon$ the new permutation would be chosen based on the model's evaluation using the DLBS values.

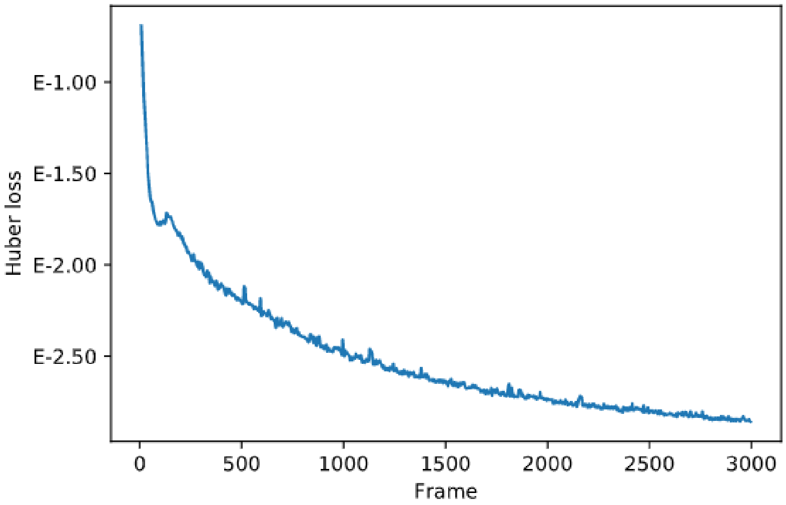


Figure 17: Loss value over training process.

Structure predictions algorithms

The *good_and_unique* algorithm uses two parameters, *n* and *q*. The *n* parameter determines the maximum number of candidate structures that well be retained at each iteration. The *q* parameter determines the weighting applied to the value of variation within these *n* predictions. The *n* predictions with the highest value are returned from the *good_and_unique* algorithm. These metrics were tuned by splitting the testing set in half and testing various values for *n* and *q* as shown in Figures 18 and 19. Overall *n*=13, *q*=1.0 worked well for both halves of the testing dataset. Given that there was little difference between the results on both halves of the testing set, the remainder of this thesis will give results on the full testing dataset.

```
$\boldsymbol{Algorithm}: \boldsymbol{good\_and\_unique}$
$inputs: \{prediction\}\ predictions, integer\ n, float\ q$
$output: A\ set\ of\ the\ best\ n\ predictions$

$new\_predictions = \{\ \}$
$\boldsymbol{while}\ |new\_predictions| < n\ \&\&\ |predictions| > 0\ \boldsymbol{do}$
    $best\_i = 0$
    $best = -\infty$
    $for\ i, p\ in\ enumerate(predictions) do$
        $d = average(stdev(new\_predictions + \{p\}))$
        $v = average(\{\_p.value\ for\ \_p\ in\ (new\_predictions + \{p\})\})$
        $s = v + q * d$
        $\boldsymbol{if}\ x > best\ \boldsymbol{then}$
            $best = s$
            $best\_i = i$
        $\boldsymbol{end\ if}$
    **end for**
    $new\_predictions = new\_predictions + \{predictions.pop(best\_i)\}$
**end while**
$\boldsymbol{return}\ new\_predictions$
```

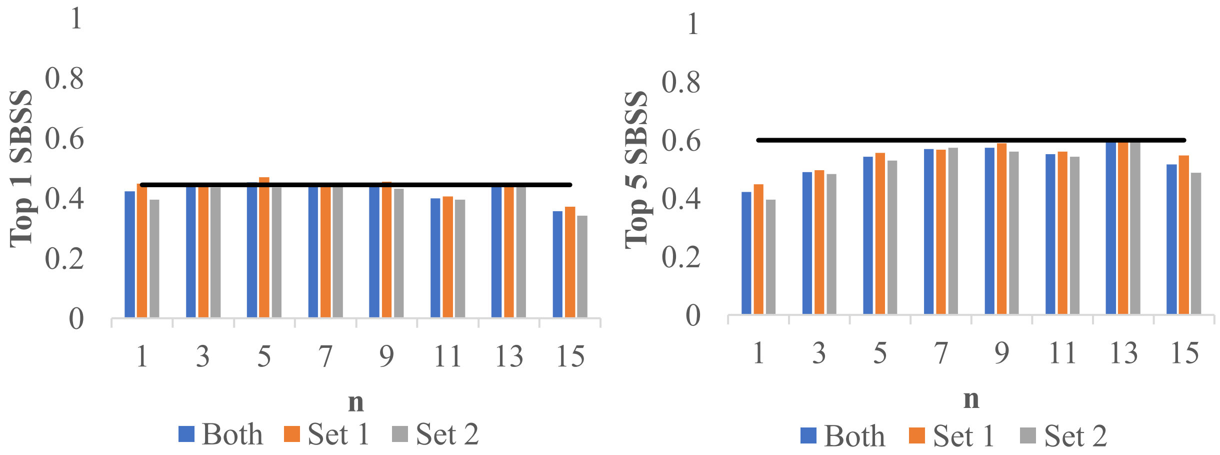


Figure 18: Top 1 scores (left) and best of top 5 scores (right) by q parameter value.

The structure prediction process can easily be modified to output multiple predictions by simply taking the top *n* predictions at each step. This modified process can be thought of as building a tree of possible new states. Branches from this tree are continued if they received scores high enough to put them in the top *n* predictions.

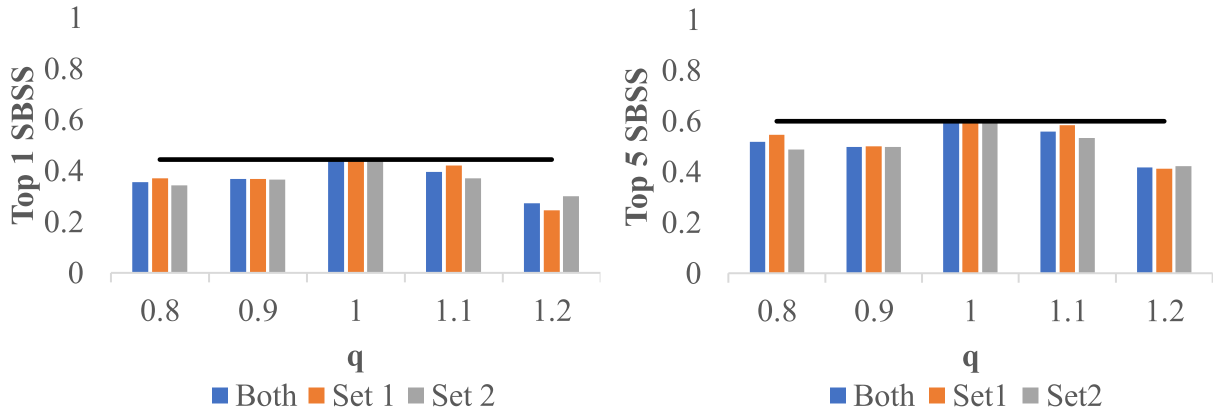


Figure 19: Top 1 scores (left) and best of top 5 scores (right) by n parameter value.

Any predictions with DLCS values greater than 1 are set aside into a list of finished predictions and new and unfinished predictions are placed into a separate list. At each step, the *good_and_unique* algorithm is used on both lists. This modified process ends when none of the new and unfinished predictions improve upon any of the previous best predictions. The finished and unfinished predictions are merged and *good_and_unique* algorithm is applied one more time. This list is sorted by the DLBS, and the top *n* results are returned.

Structure predictions results. The structure prediction network was next used to predict structures for the testing dataset. Here the system again started with a single carbon atom and modified the structure of the compound over multiple steps. This process uses two populations: finished and unfinished predictions. If a prediction has a completion value of >=1 then it is placed into the finished predictions population, otherwise it is appended to the unfinished predictions. This method uses good and unique algorithm for both populations at each step in this process. Once a step delivers no new improved predictions the prediction process ends. The finished and unfinished populations are then combined and the *good_and_unique* algorithm is applied one more time. These predictions are sorted by the model's scores and the results are output. Below are a set of these results. The rest of the results for the testing set can be found in Appendix C.

Figure 20 shows the results from two samples which the model was able to provide correct predictions within its top 5 predictions. Sample *42072-39-9* is an example of a compound which the model predicted correctly with its top result. The other predictions are similar alternatives to this prediction. Sample *30414-53-0* is an example of a structure which the model was able to predict, but not as its top result. The fifth prediction does match the truth, but there were four other predictions that did not match the original and were given higher scores by the model. In both these cases the model's score for the original structure is shown. Here we can see that the model gives scores less than 1.0 to both these true structures. This suggests the model know the truth but has low confidence in its prediction.

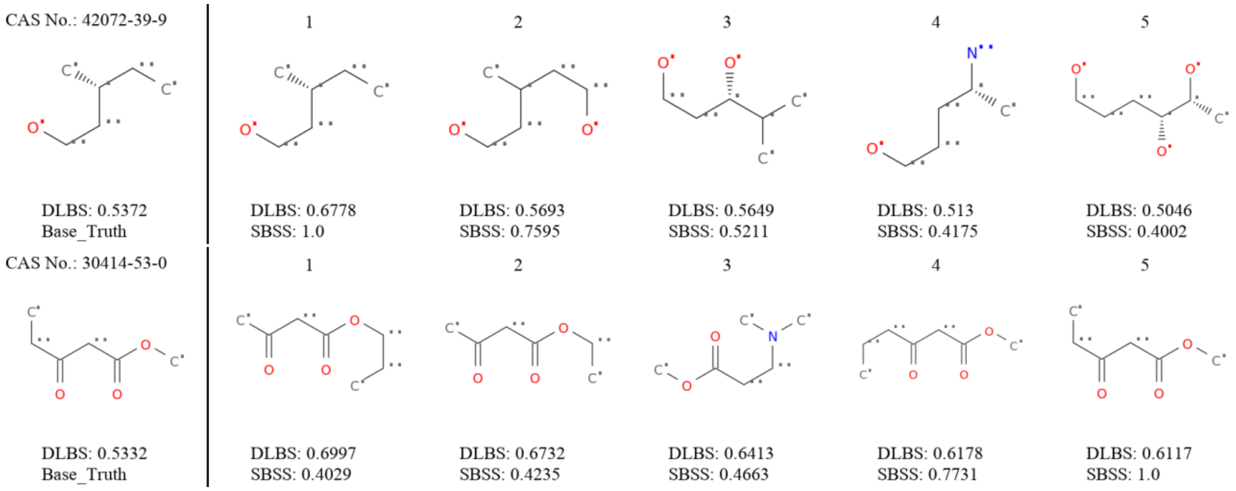


Figure 20: Results for 42072-39-9 (top) and 30414-53-0 (bottom). Shows true evaluation and model's predicted value for top 5 predictions and base truth.

Figure 21 shows the results from two samples which the model was able to predict to some degree, but not perfectly. The true structures were not in these predictions. Here the model identified elements of the true structure but was not able to recreate the true structure. The results for *87-91-2* shows that the network was able to identify elements the carboxylic (R−CO_2H) of the structure but missed the symmetry within the structure. The network produced results for *109-76-2* that formed chains of single bonded carbon atoms with nitrogen atoms. However, the network was unable to correctly predict the exact shape of the structure.

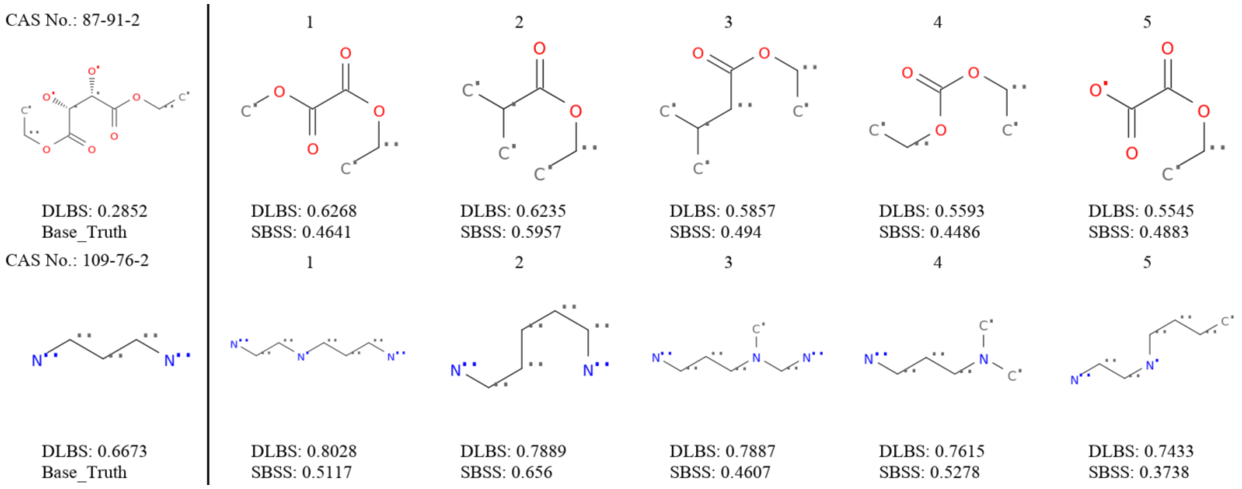


Figure 21: Results for 87-91-2 (top) and 109-76-2 (bottom). Shows true evaluation and model's predicted value for top 5 predictions and base truth.

Figure 22 shows two samples which were poorly predicted by the model. We can see that the model gave the true structures for these predictions low scores, thus suggesting that the model has not captured the relationship between spectrum and structure well for these compounds. Both structures contain carbon triple bonds which are absent from the model's predictions. The *628-36-4* results show that the network often misses the nitrogen in the structure, incorrectly identifies a C=C substructure, and tends to miss the symmetric nature of the compound. The *112-45-8* results miss some details about the compound's shape and exclude the C=C substructure present in the true structure.

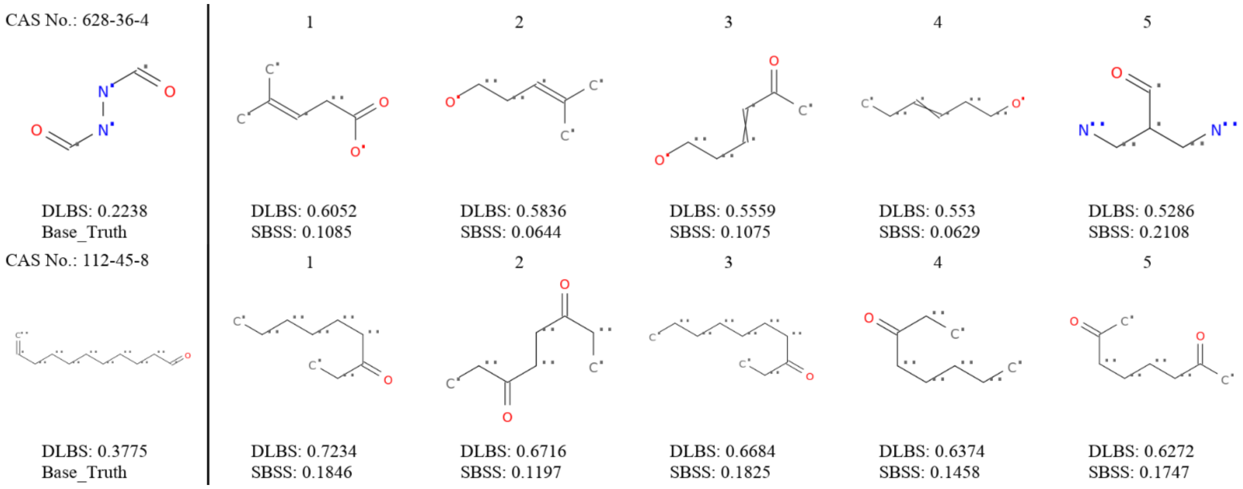


Figure 22: Results for 628-36-4 (top) and 112-45-8 (bottom). Shows true evaluation and model's predicted value for top 5 predictions and base truth.

Figure 23 shows a histogram of the SBSS for the model's highest DLBS for each sample in the testing set. Again, the evaluation scores used here are based on the harmonic mean of three metrics. The first metric is substructure similarity, the second was the fingerprint similarity, and the third was atom bond pair similarity. These metrics are useful for measuring the similarity of predictions shown here to the truth. Remember that in a previous section these metrics were shown to be useful for building a compound with the greedy structure building algorithm. Therefore, we case use this metric to judge how similar the predictions are to the truth.

The results of this method can also be judged by accuracy. The model produces 1 correct top prediction and 9 correct top 5 predictions from the 116 tested structures giving this method a 0.86% top 1 accuracy and an 7.76% top 5 accuracy. These results show that this method is still not capable of giving the exact structure of the compound. However, the structures it produces are often structurally similar to the truth as indicated in SPR4.

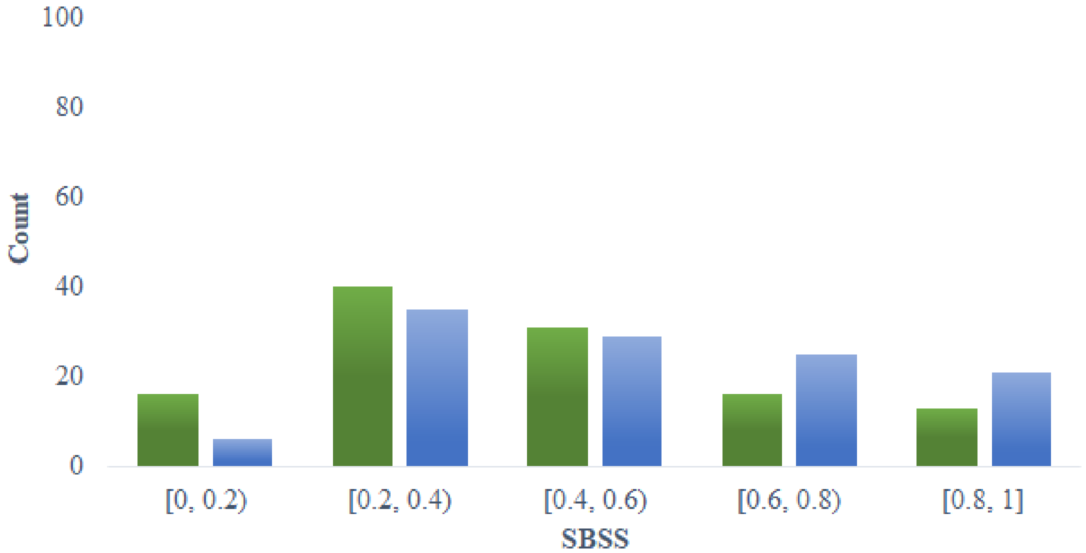


Figure 23: Histogram of SBSS for top 1 results (green) and best of top 5 results (blue).

Figure 24 shows a histogram of the individual metrics for the top 1 prediction for each testing sample. This shows that the model's DLBS correlates well with the substructure similarity and atom bond pairs metrics. However, the model has trouble predicting the fingerprint similarity metric. Therefore, the model tends to create structures that are of a similar size and contain similar substructures, but also differ from the true structure in meaningful ways.

Figure 25 shows a histogram of the individual metrics for the best SBSS from the top 5 DLBS for each testing sample. Moving to the top 5 predictions improves the top predictions scores based on the fingerprint metric, but at the cost of the substructure similarity and atom bond pairs metrics. This would indicate that the model is having a hard time generating structures which simultaneously satisfy each of these metrics.

Figure 26 shows a scatter plot of each prediction's SBSS vs the model's DLBS. The true similarity is the x-axis and the model's prediction for this value is the y-axis. This graph contains 5 predictions for each of the 106 testing samples. The correlation between the SBSS and DLBS values is 0.5433. Overall, this means the deep learning method's ability to model the SBSS is imperfect. The model can consistently produce structures that are similar to but is generally unable to completely predict the true structure.

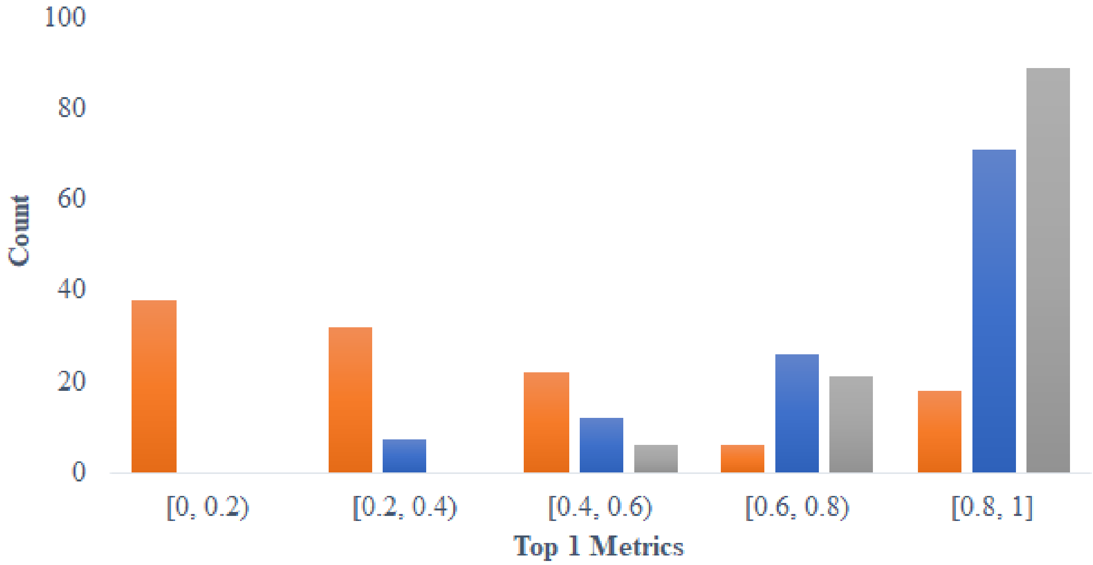


Figure 24: Histogram of top 1 predictions by fingerprint similarity (orange), substructure similarity (blue), and atom bond pairs (gray) metrics.

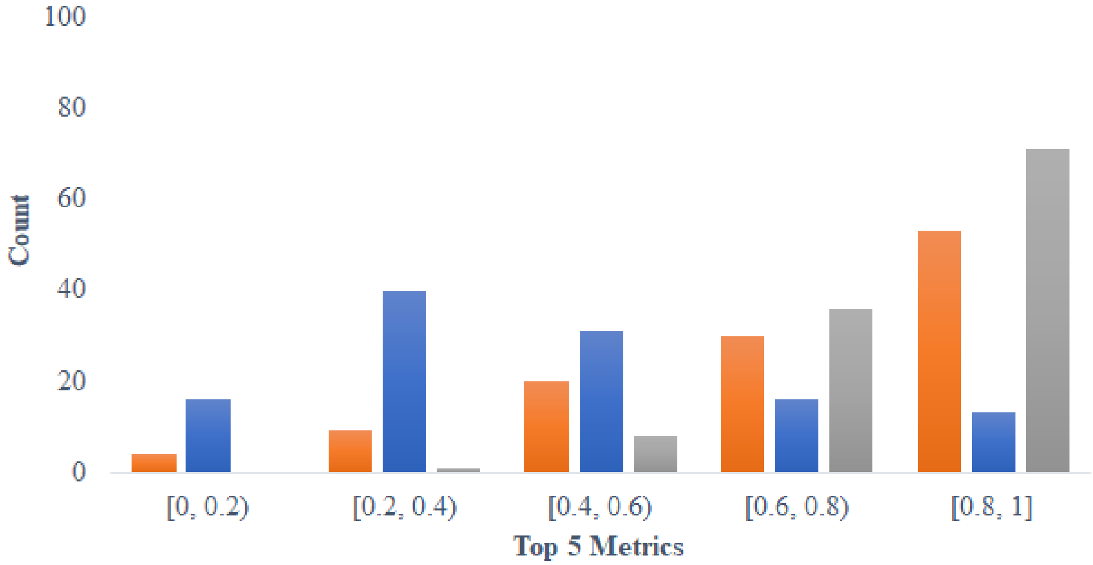


Figure 25: Histogram of best of top 5 predictions by fingerprint similarity (orange), substructure similarity (blue), and atom bond pairs (gray) metrics.

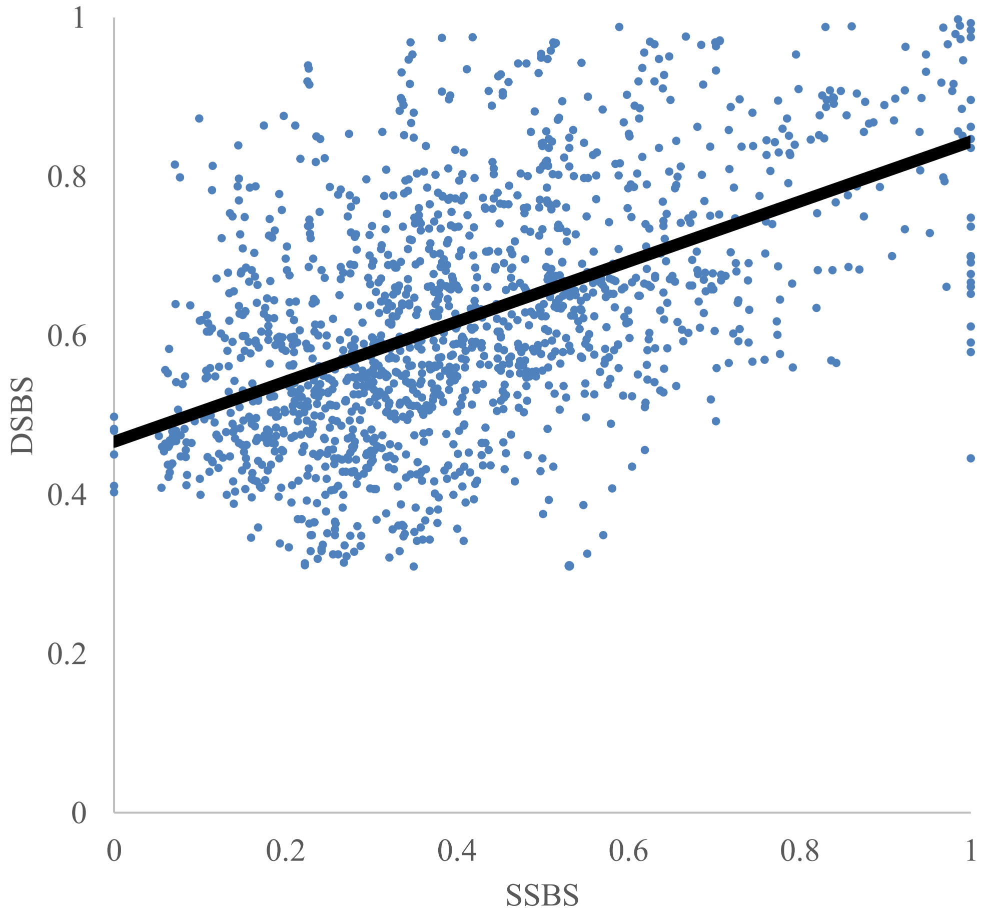


Figure 26: Scatter plot of the prediction's SBSS vs the model's DLBS.

CONCLUSIONS

The results presented in this paper show that the relationship between a compound's FTIR spectrum and its structure can be modeled to some degree. The first experiments conducted in this paper were based on using binary classifier networks to analyze an FTIR spectrum in order to predict whether a given substructure was present in a compound. These networks were optimized by changing from uniformly chosen hyperparameters to hyperparameter curves. The networks were optimized against a subset of the substructures present in the dataset using a genetic algorithm which maximized the average F1-score and minimized variance between replicates. These hyperparameter curves were used to train each of the substructure networks. Finally, substructures with too few examples to reliable be used in the network training process were instead predicted through an MLP which was trained on an PCA transformation of the outputs from the last layer of the networks with sufficient substructure counts. The results of these experiments showed that many substructures could be predicted with high accuracy with this method. However, this also revealed a fundamental limitation which was the limited dataset. A strong connection was shown between substructure occurrence rates and the F1-score of substructure predictions. My conclusion is that this method could potentially be used to predict a wide variety of substructures on the condition that a large enough dataset of spectra could be created. Specifically, this larger dataset would need to contain a variety of structures which provide a sufficient number of examples for even the rarer substructures.

The next investigation was into a method for iteratively building a compound's structure. In this section I created a structure evaluation method which could be used to iteratively build a compound's structure by listing possible additions to a structure and greedily choosing the best one. This process would stop when no possible new modification to the structure would increase the similarity by this evaluation metric. This metric was based on the harmonic mean of the fingerprint similarity, substructure similarity, and atom bond pairs similarity. Results showed this method had some issues with cyclic structures and symmetrical structures. However, I predicted that Q-learning would help overcome these issues to some degree by allowing this method to understand the future value of each possible modification and not just the immediate benefit. Hopefully, this would allow it to overcome the limitation of only greedily choosing the modification giving the highest value by the evaluation metric.

The third method explored was a deep Q-learning method which attempted to estimate the future rewards of the greedy structure building method by learning the relationship between the input FTIR spectrum, the current state of the structure prediction it is building, and the evaluation metric's value for the current version of the structure prediction. It was not clear if utilizing Q-learning would be sufficient to overcome the symmetry issue. In the results from this experiment, the deep Q-learning method used still had low performance when predicting the spectra-structure relationship of symmetric structures. However, it also had issues with non-symmetric structures.

Overall, this method was not as successful as I had hoped it would be. However, I believe the promise of these approaches. Even though the percentage of the cases where network predicted the true structure was relatively low, the predicted structures often showed similarities to the true structures. With further development of this method, it should be possible to push these results further. It may also be possible to generate the spectrum of a given structure and use spectral similarity to help improve this process. By comparing this current work and other works, it seems likely that predicting spectrum from structure may be considerably more effective than predicting structure from spectrum. The structure prediction method used in this paper might benefit from viewing the relationship from both directions. Additionally, improvements to the dataset, further development of the training process, development of the model, and refinement of the method could further improve results. For the dataset, it stands to reason that the limitation I saw in substructure predictions also applies to structure predictions.

The methods presented in this thesis showed varying levels of success. The substructure prediction networks proved to be quite capable for substructures with a sufficient number of examples to learn from. This work was able both increase the breadth of structural features being predicted and predict more specifically defined substructures than previously reported works. While the structure prediction method developed in this thesis work may not be perfect, these findings provided valuable insights toward the development of computational approaches that can predict a compound's structure solely based on its IR spectrum. Other structure prediction methods that employ quantum calculations do exist. However, these methods are often computationally highly expensive and therefore are not applicable in many applications. Therefore, while predicting the nature of a compound from its IR spectrum is a fundamentally challenging task, I believe the methods outlined in this thesis provide a meaningful and novel step forward in our ability to identify substances from their IR spectrum.

REFERENCES

[1] K. Varmuza, P. N. Penchev, and H. Scsibrany, “Maximum common substructures of organic compounds exhibiting similar infrared spectra,” *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 3, pp. 420–427, 1998.

[2] J. Li, D. B. Hibbert, S. Fuller, and G. Vaughn, “A comparative study of point-to-point algorithms for matching spectra,” *Chemometrics and Intelligent Laboratory Systems*, vol. 82, no. 1-2, pp. 50–58, 2006.

[3] *Compound Interest*, 2015. [Online]. Available: : https://www.compoundchem.com/wp-content/uploads/2015/02/Analytical-Chemistry-Infrared-Spectroscopy.pdf. [Accessed: 08-Jul-2021].

[4] K. Varmuza, P. N. Penchev, and H. Scsibrany, “Large and frequently occurring substructures in organic compounds obtained by library search of infrared spectra,” *Vibrational Spectroscopy*, vol. 19, no. 2, pp. 407–412, 1999.

[5] J. B. Lambert, E. P. Mazzola, and C. D. Ridge, “Introductory Experimental Methods,” in *Nuclear magnetic resonance spectroscopy an introduction to principles, applications, and experimental methods*, Hoboken, NJ: Wiley & Sons, pp. 39–41, 2019.

[6] J. H. Gross, “What is Mass Spectrometry?,” in *Mass spectrometry: a textbook*, Berlin: Springer Science & Business Media, 2018, ch. 1.2, pp. 2–6.

[7] J. Haas and B. Mizaikoff, “Advances in mid-infrared spectroscopy for chemical analysis,” *Annual Review of Analytical Chemistry*, vol. 9, no. 1, 2016, pp. 45–68.

[8] K. Varmuza, M. Karlovits, and W. Demuth, “Spectral similarity versus structural similarity: infrared spectroscopy,” *Analytica Chimica Acta*, vol. 490, no. 1-2, pp. 313–324, 2003.

[9] P. Larkin, *Infrared and Raman Spectroscopy: Principles and Spectral Interpretation, Elsevier*, Amsterdam, 2011.

[10] J. J. Kelly, C. H. Barlow, T. M. Jinguji, and J. B. Callis, “Prediction of gasoline octane numbers from near-infrared spectral features in the range 660-1215 nm,” *Analytical Chemistry*, vol. 61, no. 4, pp. 313–320, 1989.

[11] J. M. Soriano-Disla, L. J. Janik, R. A. Viscarra Rossel, L. M. Macdonald, and M. J. McLaughlin, “The performance of visible, near-, and mid-infrared reflectance spectroscopy for prediction of soil physical, chemical, and biological properties,” *Applied Spectroscopy Reviews*, vol. 49, no. 2, pp. 139–186, 2013.

[12] S. Gosav, M. Praisler, J. Van Bocxlaer, A. P. De Leenheer, and D. L. Massart, “Class identity assignment for amphetamines using neural networks and GC–FTIR data,” *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 64, no. 5, pp. 1110–1117, 2006.

[13] M. Novic and J. Zupan, “Investigation of infrared spectra-structure correlation using Kohonen and counterpropagation neural network,” *Journal of Chemical Information and Computer Sciences*, vol. 35, no. 3, pp. 454–466, 1995.

[14] L. Hong, M. Chun, L. Fu, S. Nie, X. Feng, and M. Long, “Substructure prediction from infrared spectra by using support vector machines,” *Chinese Chemical Letters*, vol. 16, no. 10, pp. 1354–1356, 2005.

[15] M. C. Hemmer and J. Gasteiger, “Prediction of three-dimensional molecular structures using information from infrared spectra,” *Analytica Chimica Acta*, vol. 420, no. 2, pp. 145–154, 2000.

[16] J. A. Fine, A. A. Rajasekar, K. P. Jethava, and G. Chopra, “Spectral deep learning for prediction and prospective validation of functional groups,” *Chemical Science*, 13-Mar-2020. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2020/sc/c9sc06240h. [Accessed: 13-Jul-2021].

[17] J. Neugebauer, M. Reiher, C. Kind, and B. A. Hess, “Quantum chemical calculation of vibrational spectra of large molecules–Raman and IR spectra for Buckminsterfullerene,” *Journal of Computational Chemistry*, vol. 23, no. 9, pp. 895–910, 2002.

[18] K. Ghosh, A. Stuke, M. Todorović, P. B. Jørgensen, M. N. Schmidt, A. Vehtari, and P. Rinke, “Machine Learning: Deep Learning Spectroscopy: Neural Networks for Molecular

Excitation Spectra (Adv. Sci. 9/2019)," *Advanced Science*, vol. 6, no. 1801367, pp. 1-7, 2019.

[19] S. Ye, K. Zhong, J. Zhang, W. Hu, J. Hirst, G. Zhang, and J. Jiang, "A Machine Learning Protocol for Predicting Protein Infrared Spectra," *Journal of the American Chemical Society*, vol. 142, no. 45, pp. 19071–19077, 2020.

[20] P. Kovács, X. Zhu, J. Carrete, G. K. Madsen, and Z. Wang, "Machine-learning Prediction of Infrared Spectra of Interstellar Polycyclic Aromatic Hydrocarbons," *The Astrophysical Journal*, vol. 902, no. 2, pp. 100, 2020.

[21] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of Chemical Information and Modeling*, vol. 28, no. 1, pp. 31–36, 1988.

[22] K. Varmuza and H. Scsibrany, "Substructure Isomorphism Matrix.," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 2, pp. 308-313, 2000

[23] M. Hirohara, Y. Saito, Y. Koda, K. Sato, and Y. Sakakibara, "Convolutional neural network based on SMILES representation of compounds for detecting chemical motif," *BMC Bioinformatics*, vol. 19, no. 19, pp. 83-94, 2018.

[24] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 173–187, 1996.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, D. Hassabis, "Human-level control through deep reinforcement learning," 2015. [Online]. Available: https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassibis15NatureControlDeepRL.pdf. [Accessed: 08-Jul-2021].

[26] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," *Proceedings of the wWorkshop on Machine Learning in High-Performance Computing Environments*, pp. 1-5, 2015.

[27] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464-472, 2017.

[28] D. Rogers, and M. Hahn, “Extended-connectivity fingerprints,” *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742-754. 2010.

[29] D. Bajusz, A. Rácz, and K. Héberger, “Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?” *Journal of Cheminformatics*, vol. 7, no. 1, pp. 1-13, 2015.

[30] “PubChem,” *National Center for Biotechnology Information. PubChem Compound Database*. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/. [Accessed: 08-Jul-2021].

APPENDICES

Appendix A. Substructure hyperparameter curves.

$\mathrm{Epochs} = [0.047(x - 0.9745)^1 - 0.2116(x - 0.2728)^2 + 0.0528(x + 0.7725)^3 + 0.2041] * 300 + 100$
$\mathrm{Patience} = [-0.0214(x - 0.298)^1 - 0.4379(x + 0.6145)^2 + 0.069(x + 0.3208)^3 + 0.9842] * 99 + 1$
$\mathrm{Patience\ start} = [-0.0267(x - 0.8995)^1 - 0.0506(x - 0.6714)^2 + 0.0119(x - 0.5949)^3 + 0.5772] * 100$
$\mathrm{Batch\ size} = [0.8833(x + 0.4823)^1 + 0.0496(x + 0.7182)^2 - 0.379(x - 0.0411)^3 - 1.1391] * 336 + 64$
$\mathrm{Class\ weight} = -0.1882(x - 0.683)^1 + 0.0977(x + 0.2613)^2 - 0.1961(x + 0.5303)^3 + 0.5073$
$\mathrm{Momentum} = -0.1575(x + 0.2917)^1 - 0.5004(x - 0.6315)^2 - 0.2746(x + 0.5987)^3 + 2.214$
$\mathrm{LRDecay} = 0.3394(x - 0.4892)^1 + 0.1081(x + 0.8084)^2 + 0.0558(x - 0.3281)^3 + 0.2503$
$\mathrm{LRShortCycle} = 10\hat{}(-10 * [0.2064(x - 0.5378)^1 - 0.1252(y - 0.4827)^1 + 0.2273(x - 0.228)^2 + 0.1541(y + 0.3901)^2 + 0.1186(x + 0.6983)^3 - 0.0048(y + 0.1242)^3 - 1.1597])$
$\mathrm{LRShortCycle} = 10\hat{}(-10 * [0.2529(x - 0.0924)^1 - 0.1458(y - 0.3639)^1 - 0.1981(x - 0.228)^2 + 0.1954(y + 0.2633)^2 + 0.0603(x - 0.4494)^3 + 0.0533(y + 0.5467)^3 - 0.0525])$
$LRShortCycleLen = 23$
$\mathrm{Conv2DSize} = [53, 140, 23, 160]$
$\mathrm{DenseSize} = [95]$
$\mathrm{LRAlpha} = 10\hat{}(-3.5839)$

Appendix B. Structure prediction results.

Substructure	**Count**	**Precision**	**Recall**	**F1-score**
0C-1C	5189	0.98473	0.99422	0.98945
0C-1C(0-2C)	4252	0.8802	0.85919	0.86957
O	4119	0.94417	0.94189	0.94303
0C=1C	3562	0.94302	0.91944	0.93108
0C=1C(0-2C)	3515	0.9422	0.91831	0.9301
0C=1C-2C(0-3C)	3251	0.94921	0.90881	0.92857
0O-1C	3051	0.9085	0.9085	0.9085
0C-1O(0-2C)	3011	0.91176	0.92079	0.91626

0C-1C=2C(0=3C)	2909	0.94483	0.93836	0.94158
0C=1C-2C=3C(0-4C)	2877	0.9338	0.92096	0.92734
1?=0?-5?=4?-3?=2?-1	2840	0.94366	0.93056	0.93706
0C-1C-2C(0-3C)	2817	0.81887	0.77778	0.79779
0O=1C	2719	0.96139	0.94677	0.95402
0C-1C=2C(0-3C)	2676	0.85338	0.85019	0.85178
1C=0C-5C=4C-3C=2C-1	2658	0.89209	0.91176	0.90182
0C=1O(0-2C)	2592	0.92941	0.948	0.93861
0C-1C=2C-3C(0-4C)	2552	0.85039	0.84375	0.84706
0C-1C(0=2C)(0-3C)	2467	0.832	0.8595	0.84553
0C=1C-2C(1-3C)(0-4C)	2368	0.85714	0.86076	0.85895
0C=1C(0-2C=3C)(0-4C)	2258	0.84681	0.88444	0.86522
0C-1C-2O(0-3C)	2086	0.79902	0.79126	0.79512
0C-1C=2O(0-3C)	2061	0.82927	0.89005	0.85859
0C-1C-2C-3C(0-4C)	1829	0.79355	0.66848	0.72566
N	1772	0.85987	0.77143	0.81325
0N-1C	1600	0.84564	0.80255	0.82353
0O-1C(0-2C)	1598	0.89796	0.83019	0.86275
0C-1O-2C(0-3C)	1576	0.85417	0.77848	0.81457
0C=1O(0-2O)	1557	0.88235	0.90604	0.89404
0C-1C(0-2C=3C)(0=4C)	1552	0.72549	0.76552	0.74497
0C-1C-2O(0=3C)	1536	0.81169	0.81169	0.81169
0C-1O(0=2O)(0-3C)	1501	0.89286	0.88028	0.88652
0C-1C-2C=3C(0-4C)	1465	0.6993	0.67114	0.68493
0C=1C-2C-3O(0-4C)	1424	0.82734	0.8042	0.8156
0C-1N(0-2C)	1393	0.8189	0.76471	0.79087
0C-1C-2C(1=3C)(0-4C)	1355	0.67391	0.66429	0.66906
0C-1C-2C-3O(0-4C)	1341	0.74627	0.76336	0.75472
0C-1C=2O(0=3C)	1303	0.84328	0.86923	0.85606
0C=1C-2C=3O(0-4C)	1188	0.808	0.84167	0.82449
0C-1C-2C=3O(0-4C)	1160	0.73737	0.68868	0.7122
0C-1C-2O(1=3O)(0-4C)	1129	0.77679	0.82857	0.80184
0C-1C-2C=3O(0=4C)	1016	0.77064	0.80769	0.78873
0C-1C(0-2C-3C)(0=4C)	1010	0.53097	0.54545	0.53812
0C-1C-2O-3C(0-4C)	1010	0.72632	0.71134	0.71875
0C-1O-2C-3C(0-4C)	1008	0.73913	0.69388	0.71579
0C-1O(0=2C)	992	0.78	0.78788	0.78392
0C=1C(0-2C=3O)(0-4C)	990	0.74286	0.82105	0.78
0C-1N(0=2C)	977	0.7957	0.73267	0.76289

0C=1C-2O(0-3C)	970	0.8	0.81633	0.80808
0C-1C(0-2C)(0-3C)	959	0.70455	0.62	0.65957
0C=1C(0-2O)(0-3C)	955	0.83158	0.84043	0.83598
0C=1C-2C(1-3O)(0-4C)	946	0.82796	0.81915	0.82353
0C=1C-2N(0-3C)	924	0.76344	0.73958	0.75132
0C-1C=2C-3O(0=4C)	921	0.82418	0.79787	0.81081
0O-1C=2O(0-3C)	907	0.90426	0.93407	0.91892
0C-1C=2C(1-3O)(0=4C)	896	0.86517	0.84615	0.85556
0C-1C-2N(0=3C)	867	0.73973	0.60674	0.66667
0O-1C-2C=3C(0-4C)	838	0.78205	0.75309	0.7673
0O-1C-2C(1=3O)(0-4C)	837	0.90123	0.87952	0.89024
0C=1C-2C-3N(0-4C)	821	0.72857	0.61446	0.66667
0C-1C-2C(1-3C)(0-4C)	817	0.69565	0.56471	0.62338
0C-1C=2C-3N(0=4C)	788	0.81159	0.70886	0.75676
0N-1C(0-2C)	783	0.725	0.75325	0.73885
0C=1C(0-2N)(0-3C)	779	0.79452	0.6988	0.74359
0C-1C-2C-3O(0=4C)	777	0.65854	0.72	0.6879
0C-1C-2C=3C(0=4C)	751	0.625	0.64286	0.6338
0C=1C(0-2C-3O)(0-4C)	746	0.67123	0.68056	0.67586
0C-1N-2C(0-3C)	741	0.72857	0.68	0.70345
0C-1C-2N(0-3C)	740	0.60563	0.59722	0.6014
0C=1C-2C(1-3N)(0-4C)	737	0.73846	0.61538	0.67133
0C-1C(0-2C-3C)(0-4C)	730	0.66129	0.54667	0.59854
0C-1C(0-2C=3O)(0=4C)	729	0.63636	0.71014	0.67123
Cl	719	0.54412	0.52857	0.53623
0C-1O-2C=3O(0-4C)	717	0.88889	0.91429	0.90141
0Cl-1C	684	0.53846	0.51471	0.52632
0C-1C(0-2O)(0-3C)	667	0.75385	0.66216	0.70504
0C-1C=2C(1-3N)(0=4C)	662	0.84746	0.73529	0.7874
0C-1C(0=2O)(0-3C)	645	0.72727	0.7619	0.74419
0C-1Cl(0-2C)	638	0.53571	0.5	0.51724
0C-1C-2C(1=3O)(0-4C)	599	0.7963	0.76786	0.78182
0C-1C(0-2C-3O)(0=4C)	591	0.68966	0.70175	0.69565
0O-1C=2C(0-3C)	576	0.83333	0.71429	0.76923
0C=1C-2O-3C(0-4C)	558	0.85714	0.76364	0.80769
0O-1C-2C(1=3C)(0-4C)	544	0.83673	0.77358	0.80392
0C-1C-2O(1=3O)(0=4C)	537	0.88	0.83019	0.85437
0C-1C-2C(1-3O)(0-4C)	523	0.72727	0.68966	0.70796
0N=1C	505	0.725	0.54717	0.62366

0C-1N-2C-3C(0-4C)	476	0.63636	0.65116	0.64368
0C-1C(0-2C-3C)(0=4O)	475	0.58696	0.72973	0.6506
0C-1N(0=2O)	473	0.78431	0.8	0.79208
0C-1C-2Cl(0=3C)	467	0.4	0.43902	0.4186
0C=1C-2C-3Cl(0-4C)	448	0.41463	0.44737	0.43038
0N-1C-2C=3C(0-4C)	438	0.67568	0.55556	0.60976
0C-1C(0-2C-3C)(0-4O)	433	0.63462	0.71739	0.67347
0C-1C-2N-3C(0-4C)	431	0.5814	0.5814	0.5814
0C-1C-2C-3N(0-4C)	430	0.54054	0.47619	0.50633
0C-1Cl(0=2C)	428	0.45455	0.38462	0.41667
0C=1C-2Cl(0-3C)	420	0.45455	0.39474	0.42254
0N-1C=2C(0-3C)	415	0.58537	0.57143	0.57831
S	412	0.64516	0.4878	0.55556
0C=1C(0-2Cl)(0-3C)	403	0.42105	0.45714	0.43836
0N=1C(0-2C)	400	0.73333	0.5641	0.63768
0C=1C-2C(1-3Cl)(0-4C)	397	0.45714	0.45714	0.45714
0C-1C=2C-3Cl(0=4C)	395	0.44118	0.44118	0.44118
0C-1C=2C(1-3Cl)(0=4C)	388	0.42857	0.46875	0.44776
0C=1C-2N-3C(0-4C)	386	0.61111	0.5641	0.58667
0N-1C=2O(0-3C)	377	0.77273	0.80952	0.7907
0C=1N(0-2C)	376	0.64286	0.5	0.5625
0C-1C-2O(0-3O)	371	0.62162	0.65714	0.63889
0C=1O(0-2N)(0-3C)	368	0.70455	0.79487	0.74699
0N-1C-2C(1=3C)(0-4C)	364	0.67742	0.56757	0.61765
0S-1C	364	0.66667	0.44444	0.53333
0C=1C(0-2C-3N)(0-4C)	350	0.58065	0.51429	0.54545
0C-1N=2C(0=3C)	349	0.58621	0.51515	0.54839
0C-1N-2C=3O(0-4C)	339	0.72973	0.81818	0.77143
0C-1C-2C-3N(0=4C)	337	0.625	0.45455	0.52632
0C-1C(0-2C=3C)(0=4O)	323	0.625	0.55556	0.58824
0C-1C(0-2C)(0-3C)(0-4C)	310	0.95455	0.55263	0.7
0C-1O-2C=3C(0-4C)	309	0.75	0.58065	0.65455
0C-1C=2N(0=3C)	305	0.63636	0.5	0.56
0C=1C-2N=3C(0-4C)	303	0.68421	0.43333	0.53061
1C-0C-5C-4C-3C-2C-1	300	0.66667	0.46154	0.54545
0C-1S(0-2C)	296	0.77778	0.48276	0.59574
Br	294	0.57143	0.34286	0.42857
0Br-1C	293	0.55	0.31429	0.4
0C-1C(0-2C-3O)(0-4C)	292	0.64	0.53333	0.58182

0N-1C-2C(1=3O)(0-4C)	290	0.76667	0.67647	0.71875
0C-1C(0-2C=3C)(0-4C)	287	0.85	0.51515	0.64151
0C-1N-2C=3C(0-4C)	286	0.75	0.62069	0.67925
0C-1C-2N(0-3O)	284	0.66667	0.62069	0.64286
0O-1C-2C-3O(0-4C)	282	0.48148	0.5	0.49057
0C=1N-2C(0-3C)	279	0.47368	0.3913	0.42857
0C-1Br(0-2C)	276	0.47826	0.31429	0.37931
0C-1C-2C-3O(0-4O)	275	0.6	0.67742	0.63636
0C-1C(0-2O-3C)(0-4C)	264	0.65	0.48148	0.55319
0N-1C(0-2C)(0-3C)	259	0.66667	0.63636	0.65116
0C=1C-2C=3N(0-4C)	258	0.7	0.30435	0.42424
0C-1S(0=2C)	250	0.68421	0.59091	0.63415
F	249	0.86957	0.76923	0.81633
0F-1C	248	0.86364	0.73077	0.79167
0C-1C-2C=3O(0-4O)	248	0.56	0.6087	0.58333
0C-1C=2C-3O(0-4C)	247	0.63158	0.5	0.55814
0C-1N-2C(1-3C)(0-4C)	246	0.72222	0.61905	0.66667
1?=0?-5?-4?-3?-2?-1	243	0.44444	0.17391	0.25
0C-1F(0-2C)	242	0.85	0.68	0.75556
0N=1C-2C=3C(0-4C)	240	0.57143	0.4	0.47059
0C-1C-2Cl(0-3C)	237	0.5	0.25926	0.34146
0C-1C(0-2C-3N)(0=4C)	236	0.36842	0.28	0.31818
0C=1C-2S(0-3C)	234	0.73333	0.57895	0.64706
0C-1C=2O(1-3N)(0-4C)	226	0.55	0.52381	0.53659
0C-1O(0-2C-3O)(0-4C)	223	0.5	0.54545	0.52174
0C-1C-2N(0=3O)	219	0.56522	0.68421	0.61905
0C-1C=2C-3S(0=4C)	219	0.73333	0.61111	0.66667
1?=0?-5?-4?-3?=2?-1	218	0.57143	0.30769	0.4
0C=1N-2C=3C(0-4C)	216	0.5	0.35294	0.41379
0C-1C(0-2C=3O)(0-4C)	214	0.6	0.42857	0.5
0C-1C=2C(1-3O)(0-4C)	213	0.33333	0.22222	0.26667
0C-1C=2C(1-3N)(0-4C)	211	0.52941	0.36	0.42857
1?=0N-5?=4?-3?=2?-1	211	0.66667	0.52632	0.58824
0C-1C-2F(0=3C)	205	0.76471	0.61905	0.68421
0C-1C=2O(0-3O)	203	0.6087	0.56	0.58333
0C=1C(0-2S)(0-3C)	203	0.78571	0.61111	0.6875
0C-1C(0-2N)(0-3C)	199	0.64286	0.45	0.52941
0C=1C-2C-3F(0-4C)	198	0.75	0.6	0.66667
0C-1N=2C(0-3C)	195	0.7	0.35	0.46667

0C-1C-2S(0=3C)	195	0.8	0.6	0.68571
0C=1C-2C(1-3S)(0-4C)	195	0.71429	0.55556	0.625
0C-1C(0-2C-3O)(0-4O)	194	0.44	0.55	0.48889
0C-1C(0=2C-3N)(0-4C)	192	0.42857	0.15	0.22222
0C=1C-2C-3S(0-4C)	191	0.86667	0.65	0.74286
0S-1C(0-2C)	190	0.88889	0.42105	0.57143
0C-1C(0-2C)(0-3O)(0-4C)	181	0.90476	0.82609	0.86364
0C-1N(0-2N)	180	0.58824	0.52632	0.55556
0C=1N(0-2N)	179	0.53846	0.36842	0.4375
0N-1C-2C-3O(0-4C)	176	0.73333	0.61111	0.66667
0C-1C=2C(1-3S)(0=4C)	176	0.76923	0.625	0.68966
0C-1C(0=2C-3O)(0-4C)	170	0.6	0.40909	0.48649
0C-1C=2C-3N(0-4C)	170	0.57143	0.21053	0.30769
0C-1N-2C=3O(0=4C)	169	0.61111	0.61111	0.61111
0C-1C=2O(1-3N)(0=4C)	168	0.4375	0.35	0.38889
0C=1C(0-2N=3C)(0-4C)	167	0.55556	0.3125	0.4
1C-0C-4C-3C-2C-1	162	0.85714	0.4	0.54545
0C=1O(0-2C-3N)(0-4O)	162	0.6875	0.73333	0.70968
0C-1F(0=2C)	161	0.85714	0.70588	0.77419
0C=1C(0-2F)(0-3C)	159	0.92308	0.70588	0.8
0C=1C-2F(0-3C)	159	0.8	0.70588	0.75
0N=1C-2N(0-3C)	158	0.58333	0.41176	0.48276
0C=1C-2C(1-3F)(0-4C)	157	0.91667	0.64706	0.75862
0C-1C=2C-3F(0=4C)	157	0.85714	0.70588	0.77419
1C=0C-5C-4C-3C-2C-1	157	0.57143	0.30769	0.4
0C-1C=2C(1-3F)(0=4C)	156	0.86667	0.76471	0.8125
0C-1C-2Br(0=3C)	153	0.55556	0.2381	0.33333
0C-1C-2C(1-3N)(0-4C)	152	0.8	0.47059	0.59259
0C=1C-2C-3Br(0-4C)	151	0.6	0.3	0.4
1C=0C-5C-4C-3C=2C-1	150	0.6	0.35294	0.44444
0S=1O	147	0.91667	0.78571	0.84615
0N#1C	144	1	0.73333	0.84615
0S=1O(0=2O)	141	0.92308	0.85714	0.88889
0S=1O(0-2C)	140	0.92308	0.85714	0.88889
0C-1Br(0=2C)	139	0.625	0.3125	0.41667
0S=1O(0=2O)(0-3C)	138	0.91667	0.78571	0.84615
0C#1N(0-2C)	137	1	0.84615	0.91667
0C-1S-2C(0-3C)	135	1	0.21429	0.35294
0C=1C-2Br(0-3C)	133	0.66667	0.25	0.36364

0C=1C(0-2Br)(0-3C)	132	0.625	0.3125	0.41667
0C-1C=2C-3Br(0=4C)	130	0.66667	0.25	0.36364
0N-1C-2C=3O(0-4C)	130	0.72727	0.72727	0.72727
0C-1C-2Br(0-3C)	129	0.71429	0.33333	0.45455
0C=1C(0-2C-3Cl)(0-4C)	129	0.5	0.14286	0.22222
0C-1C=2N(0-3C)	129	1	0.23077	0.375
0S-1C=2C(0-3C)	129	0.85714	0.54545	0.66667
0C-1C=2C-3O(0-4O)	128	0.375	0.27273	0.31579
0C=1C-2C(1-3Br)(0-4C)	127	0.83333	0.3125	0.45455
0C-1C=2C-3N(0=4O)	127	0.33333	0.13333	0.19048
0C-1S=2O(0-3C)	126	0.91667	0.84615	0.88
0C-1N=2C-3N(0=4C)	126	0.5	0.30769	0.38095
0C-1S=2O(1=3O)(0-4C)	125	0.91667	0.84615	0.88
0C-1C=2C(1-3Br)(0=4C)	124	0.66667	0.25	0.36364
0N-1N	123	0.6	0.21429	0.31579
0C-1C(0=2N)(0-3C)	121	0.6	0.1875	0.28571
0C-1O(0-2O)	120	0.875	0.5	0.63636
0C-1O(0-2C=3O)(0-4C)	120	0.41667	0.45455	0.43478
0C-1C-2S=3O(0=4C)	119	0.91667	0.91667	0.91667
0C-1C#2N(0-3C)	119	0.9	0.81818	0.85714
0N-1C-2N(0-3C)	118	0.4	0.28571	0.33333
0C-1S=2O(0=3C)	118	0.83333	1	0.90909
0C-1N=2C-3C(0-4C)	118	0.5	0.2	0.28571
0O-1C-2O(0-3C)	117	0.88889	0.61538	0.72727
0C=1C-2S=3O(0-4C)	117	0.83333	1	0.90909
0C-1S=2O(1=3O)(0=4C)	117	0.81818	0.9	0.85714
0C=1C(0-2S=3O)(0-4C)	115	0.83333	1	0.90909
0C-1C-2N=3C(0=4C)	114	0.75	0.25	0.375
0C=1C-2S-3C(0-4C)	114	0.5	0.22222	0.30769
0C-1C-2N(0-3N)	113	0.44444	0.5	0.47059
0C-1C-2S(0-3C)	112	0.66667	0.18182	0.28571
0C=1C-2O(0-3O)	111	0.75	0.375	0.5
0O-1C-2O-3C(0-4C)	110	1	0.8	0.88889
0C-1C(0-2C-3Cl)(0=4C)	109	0.6	0.25	0.35294
0C-1O(0=2C-3O)(0-4C)	109	1	0.28571	0.44444
0C-1C(0-2C=3C)(0-4O)	105	0.875	0.46667	0.6087
0O-1C-2C=3O(0-4C)	105	0.85714	0.46154	0.6
0N-1N(0-2C)	105	0.42857	0.25	0.31579
0C-1C(0-2C-3O)(0-4N)	104	0.63636	0.63636	0.63636

0C-1C-2F(0-3C)	102	0.85714	0.5	0.63158
0C=1C-2C=3O(0-4O)	102	0.42857	0.27273	0.33333
0C-1C-2C=3O(0=4O)	101	0.875	0.58333	0.7
0C-1C(0-2N-3C)(0-4C)	100	0.66667	0.36364	0.47059
0C-1C=2C-3Cl(0-4Cl)	100	1	0.22222	0.36364
0C-1N-2C-3N(0-4C)	99	0.57143	0.30769	0.4
0N-1C(0-2C=3C)(0-4C)	99	0.75	0.25	0.375
0C-1C-2C-3Cl(0-4C)	97	1	0.18182	0.30769
0C-1C(0-2C=3O)(0-4N)	97	0.625	0.5	0.55556
0C-1C-2C-3Cl(0=4C)	95	0.5	0.27273	0.35294
0C-1C-2O(1=3O)(0-4O)	95	0.77778	0.63636	0.7
0C-1O(0-2O)(0-3C)	94	0.83333	0.41667	0.55556
0O-1C-2C(1-3O)(0-4C)	91	0.85714	0.54545	0.66667
0C-1C=2C-3N(0-4O)	91	1	0.44444	0.61538
0C-1N-2C=3O(0=4O)	91	0.5	0.57143	0.53333
0C-1C-2C-3N(0-4O)	90	0.33333	0.09091	0.14286
0C-1N-2C=3C(0=4C)	90	1	0.33333	0.5
0N-1C-2C-3N(0-4C)	89	0.5	0.16667	0.25
0C-1O-2C-3O(0-4C)	88	0.875	0.58333	0.7
0O-1C=2C-3O(0-4C)	87	1	0.28571	0.44444
0C-1O-2C=3C(0=4C)	86	1	0.30769	0.47059
0S-1C-2C(1=3C)(0-4C)	85	1	0.25	0.4
0C#1C	82	1	0.75	0.85714
0C#1C(0-2C)	82	0.9	0.75	0.81818
0C=1C-2C=3N(0-4N)	82	0.5	0.33333	0.4
0C-1N(0-2N)(0=3O)	82	0.71429	0.71429	0.71429
0C-1C-2C-3N(0=4O)	81	0.6	0.3	0.4
0C-1C=2C(1-3Cl)(0-4C)	79	0	0	0
0O-1C-2C-3N(0-4C)	79	1	0.25	0.4
0C=1C(0-2C=3N)(0-4C)	79	Nan	0	0
0C=1C(0-2C-3F)(0-4C)	77	1	0.375	0.54545
0C-1C-2C#3N(0=4C)	77	0.71429	0.625	0.66667
0C-1F(0-2F)	77	1	0.85714	0.92308
0C-1N(0-2O)	76	1	0.5	0.66667
0N-1C=2N(0-3C)	76	Nan	0	0
0C-1C=2C-3Cl(0-4C)	76	1	0.25	0.4
0N-1C=2O(1-3N)(0-4C)	76	0.71429	0.71429	0.71429
0C=1C-2N(0-3N)	75	0.5	0.25	0.33333
0C-1C(0=2C-3Cl)(0-4C)	75	1	0.16667	0.28571

0C-1N-2N(0-3C)	74	0.5	0.22222	0.30769
0N-1C=2N-3C(0-4C)	73	0	0	0
0C-1C-2C-3Br(0-4C)	73	0.66667	0.28571	0.4
0C-1C(0=2N-3C)(0-4C)	73	1	0.28571	0.44444
0N-1C-2N-3C(0-4C)	72	0.83333	0.71429	0.76923
0C-1F(0-2F)(0-3C)	72	1	1	1
0C-1C(0-2C-3O)(0=4O)	71	0.75	0.33333	0.46154
0C-1C#2N(0=3C)	71	0.875	0.875	0.875
0C-1O-2C=3O(0=4C)	71	0.71429	0.625	0.66667
0C-1Cl(0=2O)	71	0.66667	0.66667	0.66667
0C-1S-2C=3C(0=4C)	71	1	0.25	0.4
0S-1C-2C=3C(0-4C)	70	1	0.11111	0.2
0C-1C-2N=3C(0-4C)	70	1	0.125	0.22222
0N-1C(0-2C=3O)(0-4C)	70	0.2	0.2	0.2
0N-1N(0=2C)	69	0	0	0
0O-1C-2N(0-3C)	69	1	0.625	0.76923
0N-1C=2N(0=3C)	69	1	0.375	0.54545
0C-1C-2O(1-3O)(0-4C)	68	0.5	0.28571	0.36364
0C-1C(0-2C=3O)(0-4O)	68	0.66667	0.57143	0.61538
0C-1F(0-2F)(0-3F)	68	1	1	1
0C-1C-2C-3F(0=4C)	67	1	0.75	0.85714
0C-1C(0-2C-3C)(0-4N)	66	0.66667	0.22222	0.33333
0C-1C=2O(0=3O)	66	0.66667	0.25	0.36364
0N-1O	65	0.66667	0.18182	0.28571
0S-1O	65	1	0.8	0.88889
0S=1O(0-2O)	65	1	0.8	0.88889
0C-1C=2N-3C(0-4C)	65	1	0.4	0.57143
0C-1C=2C-3N(0-4N)	64	0.66667	0.2	0.30769
0C-1F(0-2F)(0-3F)(0-4C)	64	1	1	1
0C-1N=2C-3N(0-4C)	63	Nan	0	0
0S=1O(0=2O)(0-3O)	63	1	0.8	0.88889
0C-1C-2C=3N(0=4C)	63	Nan	0	0
0N-1C-2N(0=3C)	62	0.5	0.33333	0.4
0C-1C-2C(1=3N)(0-4C)	62	0.66667	0.25	0.36364
0C-1C-2Cl(0-3O)	62	0.5	0.14286	0.22222
0N-1C=2C-3N(0=4C)	62	1	0.33333	0.5
0C=1C(0-2C#3N)(0-4C)	61	0.75	0.85714	0.8
0N=1C-2N=3C(0-4C)	61	1	0.42857	0.6
P	61	0.83333	1	0.90909

0S-1O(0-2C)	61	1	0.8	0.88889
0S-1O(0=2O)(0-3C)	61	1	0.8	0.88889
0S-1O(0=2O)(0=3O)(0-4C)	61	1	0.8	0.88889
0C-1N-2C=3N(0-4C)	60	0	0	0
0C-1S-2C-3C(0-4C)	60	Nan	0	0
0C-1S-2C=3C(0-4C)	60	Nan	0	0
0C=1O(0-2Cl)(0-3C)	60	0.6	0.75	0.66667
0C-1O(0-2C-3O)(0=4C)	60	0.66667	0.66667	0.66667
0C-1C#2C(0-3C)	59	1	0.8	0.88889
0C-1C-2F(1-3F)(0-4C)	58	0.8	0.8	0.8
0C-1C(0-2C-3F)(0=4C)	58	1	0.6	0.75
0C-1S-2O(0-3C)	58	1	1	1
0C-1S-2O(1=3O)(0-4C)	58	1	1	1
0C-1N-2N(0=3C)	57	0.25	0.14286	0.18182
0C=1C-2N-3N(0-4C)	57	0	0	0
0C=1C-2C#3N(0-4C)	57	0.57143	0.66667	0.61538
0C-1N-2C=3N(0=4C)	57	0	0	0
0C=1N-2N(0-3C)	56	Nan	0	0
0C-1C(0-2C#3N)(0=4C)	56	0.625	0.83333	0.71429
0C-1S-2O(0=3C)	56	1	1	1
0C=1C(0-2S-3O)(0-4C)	56	1	1	1
0C-1C-2S-3O(0=4C)	56	1	1	1
0C-1S-2O(1=3O)(0=4C)	56	1	1	1
0C-1S(0=2N)	55	0.6	0.6	0.6
0C=1N-2C=3N(0-4C)	55	0.75	0.6	0.66667
0C=1C-2S-3O(0-4C)	55	1	1	1
0C-1N-2C-3N(0=4C)	54	0.2	0.11111	0.14286
0C-1C=2C-3Cl(0=4O)	54	0	0	0
I	54	0.4	0.4	0.4
0S-1C=2N(0-3C)	54	0.6	0.6	0.6
0C-1C-2C#3N(0-4C)	54	0	0	0
0C-1C-2C-3S(0-4C)	53	0.5	0.14286	0.22222
0C-1N-2C-3N(0=4O)	53	0.71429	0.71429	0.71429
0N=1N	53	0.66667	0.33333	0.44444
0N=1N(0-2C)	53	1	0.5	0.66667
0C=1C(0-2C-3S)(0-4C)	53	1	0.5	0.66667
0I-1C	53	0.33333	0.4	0.36364
0C-1N-2N(0=3O)	53	0.66667	0.5	0.57143
0C-1C-2N-3N(0=4C)	52	0.66667	0.28571	0.4

0N-1N=2C(0-3C)	52	Nan	0	0
0C-1C-2S-3C(0-4C)	52	1	0.2	0.33333
0C-1C=2O(1-3Cl)(0-4C)	52	0.5	0.66667	0.57143
0C-1O-2C-3N(0-4C)	51	1	0.57143	0.72727
0C=1C-2Cl(0-3Cl)	50	0	0	0
0C-1I(0-2C)	50	0.4	0.4	0.4
0C-1C-2Cl(0=3O)	50	0	0	0
0C-1C(0-2C-3N)(0-4C)	50	Nan	0	0
0C-1C=2C(1-3S)(0-4C)	50	0.33333	0.2	0.25
0C-1N(0=2N)(0-3C)	50	Nan	0	0
0C-1N(0-2C-3N)(0=4O)	50	0	0	0
0C-1C(0-2C-3C)(0=4N)	49	0.5	0.16667	0.25
0C-1S(0-2N)	48	1	0.14286	0.25
0C-1C-2O(0#3C)	48	1	0.83333	0.90909
0C-1N(0=2C-3N)(0-4C)	48	0	0	0
0C-1N=2N(0-3C)	48	1	0.5	0.66667
0N=1C-2S(0-3C)	48	0.75	0.75	0.75
0C=1C-2S(0-3N)	48	0.5	0.25	0.33333
0C=1C(0-2C-3Br)(0-4C)	48	0.33333	0.25	0.28571
0C-1Cl(0=2C-3Cl)(0-4C)	47	0	0	0
0C-1C(0-2C=3C)(0=4N)	47	Nan	0	0
0C-1N=2N(0=3C)	47	1	0.4	0.57143
0C=1O(0-2C=3O)(0-4C)	47	1	0.2	0.33333
0N=1N-2C(0-3C)	47	1	0.5	0.66667
0N=1C-2S-3C(0-4C)	47	0.75	0.75	0.75
0C-1C-2C-3Br(0=4C)	47	0.33333	0.33333	0.33333
0C-1Cl(0=2N)	46	0.66667	0.28571	0.4
0S-1C-2N(0-3C)	46	0.5	0.16667	0.25
0C-1N(0-2N)(0=3C)	46	0.33333	0.16667	0.22222
0N-1C-2O(0-3C)	46	1	0.4	0.57143
0C-1O-2C=3O(0=4O)	46	0.71429	1	0.83333
0P-1O	46	0.8	1	0.88889
0P=1O	46	0.8	1	0.88889
0N-1C=2C-3N(0-4C)	46	0	0	0
0S=1C	45	1	0.28571	0.44444
0C=1C(0-2N-3N)(0-4C)	45	0.5	0.14286	0.22222
0C=1C-2N(1-3N)(0-4C)	45	0.5	0.16667	0.25
0C=1N-2N-3C(0-4C)	45	0	0	0
0C=1C(0-2N=3N)(0-4C)	45	1	0.5	0.66667

0C-1C-2F(1-3F)(0=4C)	45	1	1	1
0C-1C=2C-3F(0-4F)	44	1	0.57143	0.72727
0P-1O(0-2O)	44	1	1	1
0C-1N=2N-3C(0-4C)	44	1	0.5	0.66667
0C-1C-2N=3N(0=4C)	44	1	0.5	0.66667
0C-1N=2C-3S(0=4C)	44	1	1	1
0C=1C-2C-3Cl(0-4N)	44	0.33333	0.33333	0.33333
0N=1C-2Cl(0-3C)	43	0.66667	0.28571	0.4
0C-1C-2Br(0-3O)	43	0	0	0
0O-1C-2N-3C(0-4C)	43	1	0.4	0.57143
0C-1N(0=2N)(0-3N)	43	0.66667	0.5	0.57143
0N=1N-2C=3C(0-4C)	43	1	0.5	0.66667
0C=1C-2N=3N(0-4C)	43	1	0.5	0.66667
0C-1C(0-2C-3Br)(0=4C)	43	0.33333	0.33333	0.33333
0C-1Cl(0-2Cl)	42	0	0	0
0C=1O(0-2N)(0-3O)	42	1	0.83333	0.90909
0O-1C=2O(1-3N)(0-4C)	42	1	0.83333	0.90909
0N=1C-2N(1-3N)(0-4C)	42	0.5	0.66667	0.57143
0C-1N-2N=3C(0=4C)	41	0.5	0.2	0.28571
0C=1C-2N(0-3O)	41	0.5	0.5	0.5
0C-1O(0=2C-3N)(0-4C)	41	0.5	0.5	0.5
0C-1C(0-2C=3N)(0=4C)	41	Nan	0	0
0C-1C-2C#3C(0-4C)	40	0.85714	0.85714	0.85714
0S-1N	40	1	0.5	0.66667
0N-1O(0=2C)	40	1	0.16667	0.28571
0C=1N-2O(0-3C)	40	1	0.16667	0.28571
0C-1C-2Br(0=3O)	40	0	0	0
0C=1O(0-2N-3N)(0-4C)	40	1	0.66667	0.8
0C-1N(0-2C-3N)(0=4C)	40	Nan	0	0
0C-1C-2Cl(0-3Cl)	39	0	0	0
0C-1C-2F(0-3F)	39	0.8	0.66667	0.72727
0C-1C-2C-3N(0-4N)	39	1	0.2	0.33333
0O-1P(0-2C)	39	1	1	1
0P=1O(0-2O)	39	0.8	1	0.88889
0C-1C-2C-3Cl(0=4O)	39	0	0	0
0C-1C-2S=3O(0-4C)	38	0.75	0.6	0.66667
0C=1N-2C-3N(0-4C)	38	0.66667	0.4	0.5
0C-1O(0-2C-3N)(0-4C)	38	0.66667	0.4	0.5
0O-1P-2O(0-3C)	38	1	0.75	0.85714

0P-1O(0=2O)(0-3O)	38	0.8	1	0.88889
0S-1N(0-2C)	37	1	0.5	0.66667
0C-1Cl(0-2N)	37	1	0.16667	0.28571
0C=1S(0-2N)	37	Nan	0	0
0C-1Cl(0-2Cl)(0-3C)	37	0	0	0
0C-1N=2C-3N(0-4N)	37	0.25	0.25	0.25
0P-1C	37	0.75	1	0.85714
0N=1C-2C(1-3N)(0-4C)	37	Nan	0	0
0S-1C=2C-3N(0-4C)	37	0.66667	0.66667	0.66667
0C-1S-2C=3N(0=4C)	37	0.66667	0.66667	0.66667
0N-1C(0-2N)(0-3C)	36	1	0.2	0.33333
0C-1C=2N-3N(0=4C)	36	Nan	0	0
0C-1C-2S(0=3O)	36	1	0.25	0.4
0C=1C-2Cl(0-3O)	36	1	0.33333	0.5
0C-1C=2C-3S(0-4C)	36	Nan	0	0
0C-1O(0=2C-3Cl)(0-4C)	36	Nan	0	0
0C-1C-2N(1=3N)(0=4C)	36	Nan	0	0
0C-1C-2C=3N(0=4O)	36	1	0.33333	0.5
0C-1O-2P(0-3C)	35	1	1	1
0O-1P-2O-3C(0-4C)	35	1	0.75	0.85714
0C-1C-2N-3N(0-4C)	35	1	1	1
0C-1O(0-2C-3N)(0=4C)	34	1	0.33333	0.5
0C-1C=2C(1-3F)(0-4C)	34	0	0	0
0C-1N-2N=3C(0-4C)	34	1	0.2	0.33333
0N-1C=2C(1-3N)(0=4C)	34	0.33333	0.2	0.25
0C-1C-2S(0-3N)	34	Nan	0	0
0C-1O-2P-3O(0-4C)	34	1	0.75	0.85714
0C-1C-2C-3S(0-4O)	34	1	0.33333	0.5
0C-1C=2N(0=3N)	33	1	0.33333	0.5
0S-1N(0=2O)	33	1	0.6	0.75
0S=1O(0-2N)(0=3O)	33	1	0.6	0.75
0C-1C-2C-3F(0-4C)	33	1	0.6	0.75
0C=1N(0-2S)(0-3N)	33	0	0	0
0C-1C=2C-3Br(0-4C)	33	Nan	0	0
0C-1C=2O(1-3Cl)(0=4C)	33	0.33333	0.5	0.4
0S=1O(0-2N)(0-3C)	32	1	0.6	0.75
0S=1O(0=2O)(0-3N)(0-4C)	32	1	0.6	0.75
0C-1C-2C-3Cl(0-4O)	32	Nan	0	0
0C-1C(0-2C=3O)(0=4O)	32	0.5	0.2	0.28571

1C=0C-4C-3C-2C-1	32	0	0	0
0C-1C(0-2Br)(0-3C)	32	1	0.25	0.4
0C#1C-2C(0-3C)	32	1	0.25	0.4
0O-1P=2O(0-3C)	32	1	1	1
0C=1C-2Cl(0-3N)	32	Nan	0	0
0C-1C(0=2C-3F)(0-4C)	32	0	0	0
0O-1P-2O(1=3O)(0-4C)	32	1	1	1
0N-1C-2O(1=3O)(0-4C)	32	1	0.5	0.66667
0C-1C=2C-3Cl(0-4O)	32	Nan	0	0
0C-1C=2N-3N(0-4C)	32	Nan	0	0
0C-1N-2C-3O(0-4C)	32	1	0.66667	0.8
0S-1C-2N(1=3N)(0-4C)	32	0	0	0
0C-1N(0=2C-3S)(0-4C)	32	1	0.5	0.66667
0C-1C(0-2C-3S)(0=4C)	32	0.5	0.5	0.5
0N-1C=2C-3S(0=4C)	32	0.66667	1	0.8
0C=1N-2C=3N(0-4N)	31	Nan	0	0
0C=1C-2F(0-3F)	31	1	0.5	0.66667
0C-1F(0=2C-3F)(0-4C)	31	1	0.75	0.85714
0C-1N=2C-3Cl(0=4C)	31	0	0	0
0C=1N(0-2O)	31	1	0.33333	0.5
0S-1C-2C=3O(0-4C)	31	1	0.33333	0.5
0C-1C-2C=3N(0-4C)	30	1	0.16667	0.28571
0C-1S-2N(0=3C)	30	0.66667	0.5	0.57143
0C-1P(0-2C)	30	0.75	1	0.85714
0C-1C(0-2Cl)(0-3C)	30	0	0	0
0O-1C=2N(0-3C)	30	1	0.33333	0.5
0N-1C(0-2C-3N)(0-4C)	30	0.33333	0.33333	0.33333
0C-1O-2P=3O(0-4C)	29	1	1	1
0C-1I(0=2C)	29	0	0	0
0C=1C(0-2I)(0-3C)	29	0	0	0
0C-1C-2Cl(0-3N)	29	Nan	0	0
0C-1N=2C-3S(0-4C)	29	1	0.33333	0.5
0C=1C-2S-3N(0-4C)	29	0.66667	0.66667	0.66667
0C-1C=2C(1-3Br)(0-4C)	29	Nan	0	0
0C-1O(0-2C=3O)(0=4C)	29	1	0.5	0.66667
0C-1N-2C(1-3N)(0-4C)	28	1	0.2	0.33333
0C-1C=2N-3O(0-4C)	28	1	0.2	0.33333
0C-1C=2N(0-3N)	28	Nan	0	0
0C-1F(0-2C-3F)(0=4C)	28	1	0.75	0.85714

0C=1N(0-2Cl)(0-3C)	28	0.5	0.33333	0.4
0C-1C-2N=3N(0-4C)	28	0.66667	0.66667	0.66667
0O-1C-2C-3Cl(0-4C)	28	Nan	0	0
0P=1O(0-2C)	28	0.66667	1	0.8
0C-1C=2N(0=3O)	28	Nan	0	0
0C-1N-2N(0-3N)	28	Nan	0	0
0C-1C-2C-3S(0=4C)	28	0	0	0
0C-1S-2N(0-3C)	27	1	0.6	0.75
0C-1C(0-2C)(0-3N)(0-4C)	27	1	0.2	0.33333
0C-1S=2O(1-3N)(0-4C)	27	1	0.6	0.75
0N-1C-2Cl(0=3C)	27	1	0.25	0.4
0N=1C-2O(0-3C)	27	0.5	0.33333	0.4
0C-1C-2I(0=3C)	27	0	0	0
0C-1C-2F(0-3O)	27	1	1	1
0C=1C-2C-3I(0-4C)	27	0	0	0
0C-1C=2C(1-3I)(0=4C)	27	0	0	0
0C-1C-2C-3S(0=4O)	27	0.5	0.33333	0.4
0C-1C-2S(0-3O)	26	Nan	0	0
0C-1C-2S-3N(0=4C)	26	1	0.75	0.85714
0O-1S(0-2C)	26	1	0.33333	0.5
0O-1S=2O(0-3C)	26	1	0.33333	0.5
0N-1C(0-2N=3C)(0-4C)	26	1	0.33333	0.5
0O-1C=2N-3C(0-4C)	26	1	0.66667	0.8
0N=1C-2C(1-3Cl)(0-4C)	26	0.5	0.33333	0.4
0N=1C-2C=3N(0-4C)	26	1	0.66667	0.8
0C-1C=2N-3O(0=4C)	26	Nan	0	0
0C-1S=2O(1-3N)(0=4C)	26	0.66667	0.66667	0.66667
0C-1C(0-2C-3N)(0-4O)	26	0.66667	0.66667	0.66667
0S-1Cl	26	0.66667	1	0.8
0C=1C-2I(0-3C)	26	0	0	0
0C=1C-2C(1-3I)(0-4C)	26	0	0	0
0N=1C-2N(1-3S)(0-4C)	26	0	0	0
0C-1C=2C-3I(0=4C)	26	0	0	0
0N-1C-2C-3N(0=4C)	26	Nan	0	0
0C=1N(0-2C=3N)(0-4C)	25	1	0.4	0.57143
0C-1O(0=2N)(0-3C)	25	1	0.33333	0.5
0C=1C(0-2S-3N)(0-4C)	25	0.66667	0.66667	0.66667
0C-1S-2C-3N(0=4C)	25	1	0.33333	0.5
0C-1C=2N(1-3Cl)(0=4C)	25	0.5	0.33333	0.4

0N-1C-2N(1=3N)(0=4C)	25	1	0.66667	0.8
0S-1Cl(0=2O)	25	0.66667	1	0.8
0C-1Cl(0-2Cl)(0-3Cl)	25	0.5	0.5	0.5
0C-1C(0=2C-3S)(0-4C)	25	Nan	0	0
0C=1O(0-2C=3O)(0-4O)	24	1	0.16667	0.28571
0C-1C(0=2N-3N)(0-4C)	24	1	0.2	0.33333
0N-1C=2S(0-3C)	24	Nan	0	0
0C-1S-2C-3N(0-4C)	24	1	0.25	0.4
0C-1Cl(0-2C-3Cl)(0=4C)	24	Nan	0	0
0C-1C-2O(1-3O)(0-4O)	24	0.5	0.25	0.33333
0C-1N(0=2S)(0-3N)	24	Nan	0	0
0O-1C-2C(1=3N)(0-4C)	24	1	0.33333	0.5
0C-1S-2C=3N(0-4C)	24	0.5	0.33333	0.4
0O-1S=2O(1=3O)(0-4C)	24	1	0.33333	0.5
0C-1N(0=2C-3Cl)(0-4C)	24	0	0	0
0S-1Cl(0-2C)	24	0.66667	1	0.8
0C-1C-2I(0-3C)	24	0.66667	1	0.8
0S=1O(0-2Cl)(0=3O)	24	0.66667	1	0.8
0O-1C-2C-3Br(0-4C)	24	0	0	0
0C-1N(0-2C-3N)(0-4C)	24	1	1	1
0C-1C=2O(1-3N)(0=4O)	24	1	1	1
0N-1C-2N=3C(0-4C)	23	0	0	0
0O-1N(0-2C)	23	Nan	0	0
0O-1S-2C(0-3C)	23	1	0.33333	0.5
0C-1O-2S(0-3C)	23	1	0.33333	0.5
0C-1C-2P(0=3C)	23	0.75	1	0.85714
0C-1C#2C-3C(0-4C)	23	1	0.33333	0.5
0O-1S-2C(1=3O)(0-4C)	23	1	0.33333	0.5
0C-1O-2S=3O(0-4C)	23	1	0.33333	0.5
0C=1C-2C-3P(0-4C)	23	0.66667	0.66667	0.66667
0P-1O(0-2C)	23	0.66667	1	0.8
0S=1O(0-2Cl)(0-3C)	23	0.66667	1	0.8
0C-1C=2C-3F(0-4C)	23	Nan	0	0
0N-1C(0-2C=3N)(0-4C)	23	0	0	0
0S=1O(0=2O)(0-3Cl)(0-4C)	23	0.66667	1	0.8
0C-1Cl(0-2Cl)(0-3Cl)(0-4C)	23	0	0	0
0C-1C(0=2N-3O)(0-4C)	22	0	0	0
0C-1O-2N(0-3C)	22	Nan	0	0

0C=1O(0-2C-3Cl)(0-4C)	22	1	0.33333	0.5
0S-1C-2C-3N(0-4C)	22	Nan	0	0
0C-1C(0-2C-3Cl)(0=4O)	22	Nan	0	0
0C-1C(0-2C=3N)(0=4N)	22	1	0.66667	0.8
0C-1P(0=2C)	22	0.5	1	0.66667
0C-1S-2Cl(0-3C)	22	0.66667	1	0.8
0P-1O(0=2O)(0-3C)	22	0.66667	1	0.8
0C-1S-2Cl(0=3C)	22	0.66667	1	0.8
0C=1N-2N(0-3N)	22	Nan	0	0
0C-1C-2S-3O(0-4C)	22	0.5	0.5	0.5
0C-1C(0-2O)(0-3O)(0-4C)	22	1	0.5	0.66667
0C-1C-2N(1=3N)(0-4C)	22	Nan	0	0
0C=1C-2S-3Cl(0-4C)	22	0.66667	1	0.8
0N-1C-2C-3O(0=4C)	22	Nan	0	0
0C-1C(0-2C-3N)(0-4N)	22	1	1	1
0N-1C(0-2C=3C)(0-4N)	22	1	0.5	0.66667
0C-1O-2S-3C(0-4C)	21	1	0.33333	0.5
0N=1C-2C(1-3O)(0-4C)	21	1	0.33333	0.5
0C-1P=2O(0-3C)	21	0.66667	1	0.8
0S-1C(0=2O)(0-3C)	21	Nan	0	0
0P-1O(0-2O)(0-3C)	21	1	1	1
0C-1S=2O(1-3Cl)(0-4C)	21	0.66667	1	0.8
0C=1N-2C-3Cl(0-4C)	21	0	0	0
0P-1O(0-2O)(0=3O)(0-4C)	21	1	1	1
0C-1C-2S-3Cl(0=4C)	21	0.66667	1	0.8
0C-1S=2O(1-3Cl)(0=4C)	21	0.66667	1	0.8
0C-1N-2C=3S(0-4C)	20	Nan	0	0
0C-1N(0-2Cl)(0=3C)	20	Nan	0	0
0C-1N-2C-3N(0-4N)	20	1	0.33333	0.5
0C=1C(0-2P)(0-3C)	20	0.66667	1	0.8
0C=1C-2P(0-3C)	20	0.66667	1	0.8
0C-1S-2O-3C(0-4C)	20	1	0.5	0.66667
0C=1C-2C(1-3P)(0-4C)	20	0.66667	1	0.8
0O-1S-2C=3C(0-4C)	20	1	0.5	0.66667
0C=1C(0-2S-3Cl)(0-4C)	20	0.66667	1	0.8
0C=1O(0-2C-3N)(0-4C)	20	1	1	1
0C-1C=2C(1-3P)(0=4C)	20	0.66667	1	0.8
0C-1C=2C-3P(0=4C)	20	0.66667	1	0.8

0C-1N-2C-3O(0=4C)	20	Nan	0	0
0N-1O(0-2C)	19	Nan	0	0
0C-1C(0-2C#3C)(0-4O)	19	1	0.75	0.85714
0N-1C-2O(0=3C)	19	0.66667	0.66667	0.66667
0C-1C-2C(1-3Cl)(0-4C)	19	Nan	0	0
0C-1C=2N(0-3O)	19	Nan	0	0
0S-1C(0=2O)(0=3O)(0-4C)	19	Nan	0	0
0C#1C-2C-3O(0-4C)	19	0	0	0
0O-1C-2C#3C(0-4C)	19	1	1	1
0C-1C(0-2F)(0-3C)	18	0.75	0.75	0.75
0C=1C-2N(1-3Cl)(0-4C)	18	Nan	0	0
0N-1C-2N(1=3S)(0-4C)	18	Nan	0	0
0N-1N-2C(0-3C)	18	Nan	0	0
0C-1S-2C(1=3O)(0-4C)	18	1	0.5	0.66667
0S-1C-2C-3O(0-4C)	18	Nan	0	0
0N=1C-2C-3N(0-4C)	18	Nan	0	0
0C=1C-2C=3N(0-4O)	18	Nan	0	0
0C-1C-2C=3N(0-4N)	18	1	0.5	0.66667
0C-1C-2C-3Cl(0-4Cl)	18	Nan	0	0
0N-1C-2C=3O(0=4C)	18	Nan	0	0
0C-1C=2C-3S(0=4O)	18	Nan	0	0
0C-1C-2C(1-3F)(0-4C)	17	0.75	0.75	0.75
0N-1C=2C(1-3N)(0-4C)	17	0	0	0
0C-1C(0-2C-3C)(0-4F)	17	0.8	1	0.88889
0C-1C-2C(1-3Br)(0-4C)	17	1	0.33333	0.5
0C-1C-2O-3S(0-4C)	17	1	0.33333	0.5
0N-1C=2C(1-3Cl)(0=4C)	17	0	0	0
0C=1O(0-2C-3Br)(0-4O)	17	0	0	0
0C-1N=2C-3Cl(0=4N)	17	1	0.66667	0.8
0P-1O(0-2O)(0-3O)	17	1	0.5	0.66667
0C=1N(0-2Cl)(0-3N)	17	1	0.5	0.66667
0O-1P-2O(1-3O)(0-4C)	17	1	0.5	0.66667
0N-1C-2C-3Cl(0-4C)	17	Nan	0	0
0C-1C-2O(1=3N)(0=4C)	17	1	0.5	0.66667
0C-1N=2C-3O(0=4C)	17	0	0	0
0C-1C(0=2C-3Br)(0-4C)	17	Nan	0	0
0N-1N-2C=3O(0=4C)	17	Nan	0	0
0C=1O(0-2C-3Cl)(0-4O)	17	0	0	0
0C-1N-2O(0-3C)	16	Nan	0	0

0C-1C(0-2C#3C)(0-4C)	16	1	0.5	0.66667
0O-1P-2C(0-3C)	16	1	1	1
0C-1P-2O(0-3C)	16	0.66667	1	0.8
0N-1C-2S(0-3C)	16	Nan	0	0
0N-1C-2S-3C(0-4C)	16	Nan	0	0
0C-1C-2Cl(1-3Cl)(0-4C)	16	Nan	0	0
0N=1C-2N(1-3Cl)(0-4C)	16	1	0.5	0.66667
0C=1C-2C-3N(0-4O)	16	Nan	0	0
0C=1C-2N(1-3N)(0-4N)	16	Nan	0	0
0C-1C(0-2C-3C)(0-4Cl)	16	0	0	0
Si	16	1	1	1
0S-1S	16	0	0	0
0S-1S(0-2C)	16	0	0	0
0S-1S-2C(0-3C)	16	0	0	0
0O-1C-2N=3C(0-4C)	16	1	1	1
0C-1C-2C-3F(0-4O)	16	0	0	0
0C=1O(0-2C-3S)(0-4O)	16	0	0	0
0C-1C=2C-3Br(0-4Br)	16	Nan	0	0
0C-1N-2C=3S(0=4C)	15	Nan	0	0
0C-1N-2C=3N(0=4O)	15	0	0	0
0C-1O(0-2C-3Cl)(0-4C)	15	Nan	0	0
0C-1N(0-2C=3N)(0=4C)	15	Nan	0	0
0C-1C-2C-3F(0=4O)	15	Nan	0	0
0C-1C-2F(0-3N)	15	Nan	0	0
0O-1P-2C(1-3O)(0-4C)	15	1	1	1
0O-1P-2C(1=3O)(0-4C)	15	1	1	1
0C-1P-2O(1=3O)(0-4C)	15	0.66667	1	0.8
0C-1N=2C-3Cl(0-4C)	15	Nan	0	0
0C=1O(0-2C-3Br)(0-4C)	15	0	0	0
0C=1C-2C#3N(0-4N)	15	0	0	0
0N-1N=2C(0=3C)	15	Nan	0	0
0N-1C-2N(1=3N)(0-4C)	15	0	0	0
0C-1O(0-2C-3Cl)(0=4C)	15	1	1	1
0C-1O(0-2C-3Br)(0=4C)	15	Nan	0	0
0C-1C-2C=3N(0-4O)	15	Nan	0	0
0C=1C-2C-3Cl(0-4F)	15	Nan	0	0
0C-1O(0-2N)(0=3C)	14	Nan	0	0
0C-1C-2Br(0-3Br)	14	Nan	0	0
0C=1C-2O(1-3N)(0-4C)	14	1	0.33333	0.5

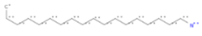
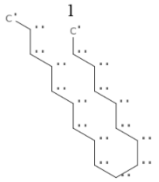
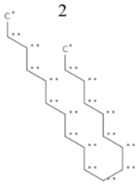
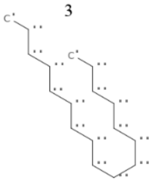
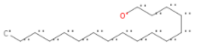
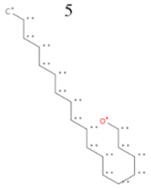
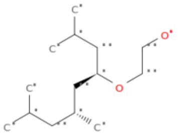
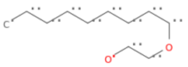
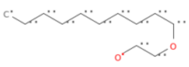
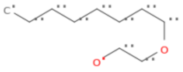
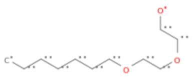
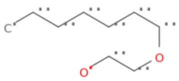
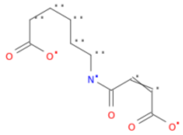
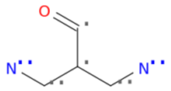
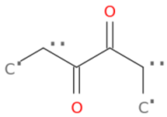
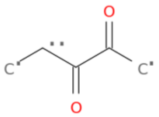
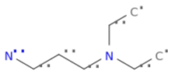
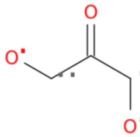
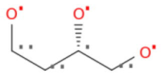
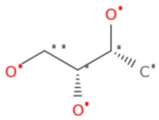
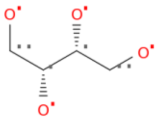
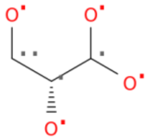
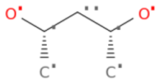
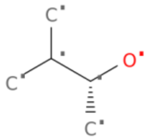
0C=1C-2C-3F(0-4N)	14	Nan	0	0
0C-1C-2C-3I(0-4C)	14	1	1	1
0C-1O-2P-3C(0-4C)	14	1	1	1
0C-1P-2O(1-3O)(0-4C)	14	1	1	1
0N-1N-2C=3O(0-4C)	14	Nan	0	0
0C-1C-2P=3O(0=4C)	14	0.5	0.5	0.5
0C-1N(0-2C-3S)(0=4C)	14	1	0.5	0.66667
0C-1C(0-2C-3Br)(0=4O)	14	Nan	0	0
0C-1C=2C-3F(0=4O)	14	0	0	0
0C-1C-2C-3F(0-4F)	14	0.5	0.5	0.5
0Si-1C	14	1	1	1
0Si-1C(0-2C)	14	0.5	1	0.66667
0P-1C(0-2C)	14	0.5	1	0.66667
0C-1P-2C(0-3C)	14	0.5	1	0.66667
0C-1C(0-2S)(0-3C)	14	Nan	0	0
0C-1N-2N-3C(0-4C)	14	Nan	0	0
0N-1C-2C=3N(0-4C)	14	Nan	0	0
0C=1C-2N=3N(0-4O)	14	0.33333	1	0.5
0O-1N=2C(0-3C)	13	Nan	0	0
0C-1C=2O(0#3C)	13	1	0.5	0.66667
0N-1C=2O(0-3O)	13	Nan	0	0
0C-1C(0-2C-3F)(0-4C)	13	1	1	1
0C-1C(0-2O-3S)(0-4C)	13	1	0.5	0.66667
0C=1N-2O-3C(0-4C)	13	Nan	0	0
0C-1N-2C-3S(0-4C)	13	Nan	0	0
0C-1C-2O-3S(0=4C)	13	1	0.5	0.66667
0C-1C(0-2C-3Cl)(0-4O)	13	Nan	0	0
0C-1C-2C-3Br(0-4O)	13	Nan	0	0
0P-1O(0-2O)(0=3O)(0-4O)	13	1	0.5	0.66667
0C-1S-2S(0-3C)	13	0	0	0
0P-1C=2C(0-3C)	13	0.5	1	0.66667
0C-1P=2O(0=3C)	13	0.33333	1	0.5
0C-1P-2C-3C(0-4C)	13	0.5	1	0.66667
0C-1S-2S-3C(0-4C)	13	0	0	0
0P-1C-2C(1=3C)(0-4C)	13	0.5	1	0.66667
0C=1C-2P-3C(0-4C)	13	0.5	1	0.66667
0C-1P-2C=3C(0-4C)	13	0.5	1	0.66667
0P-1C-2C=3C(0-4C)	13	0.5	1	0.66667
0N=1C-2C=3O(0-4C)	13	Nan	0	0

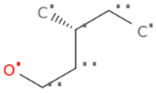
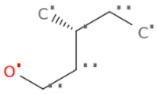
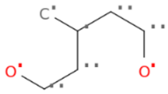
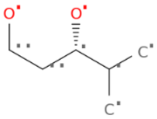
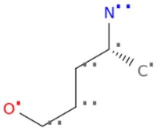
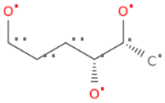
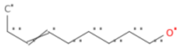
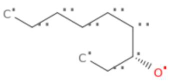
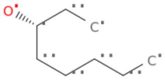
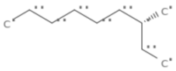
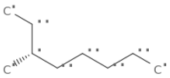
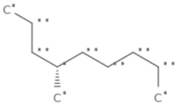
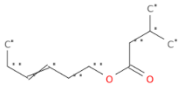
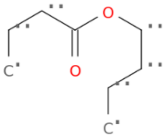
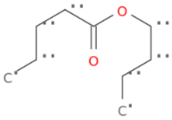
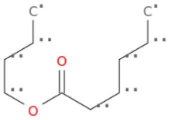
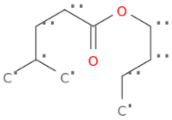
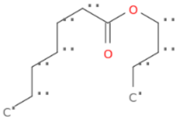
0N-1C-2C-3S(0-4C)	13	0	0	0
0C-1P-2C=3C(0=4C)	13	0.5	1	0.66667
0N-1C=2N(1-3Cl)(0=4C)	13	1	1	1
0C-1C=2C-3F(0-4O)	13	0	0	0
0C-1C-2F(1-3F)(0-4O)	13	1	1	1
0C=1O(0-2C=3N)(0-4O)	13	Nan	0	0
0C-1C-2C#3N(0=4O)	13	Nan	0	0
0S-1C(0-2C=3C)(0=4O)	13	0	0	0
0N-1C=2C(1-3O)(0=4C)	12	Nan	0	0
0C-1N=2C-3N(0-4O)	12	1	0.33333	0.5
0C-1C(0-2C-3Cl)(0-4C)	12	Nan	0	0
0C-1C-2N-3O(0-4C)	12	Nan	0	0
0C-1O-2N=3C(0-4C)	12	Nan	0	0
0C-1C-2C#3C(0=4C)	12	Nan	0	0
0C-1N(0-2C-3F)(0=4C)	12	Nan	0	0
0C-1C(0-2C-3N)(0=4O)	12	1	1	1
0C-1C-2O(1=3O)(0#4C)	12	1	1	1
0C-1C-2Cl(1-3Cl)(0-4O)	12	Nan	0	0
0C-1C-2Cl(1-3Cl)(0=4O)	12	Nan	0	0
0C-1C(0-2C=3O)(0=4N)	12	Nan	0	0
0N-1S(0-2C)	11	Nan	0	0
0N-1C(0-2O)(0-3C)	11	Nan	0	0
0N-1S-2C(0-3C)	11	Nan	0	0
0N-1S=2O(0-3C)	11	1	0.33333	0.5
0C-1N-2C(1-3O)(0-4C)	11	Nan	0	0
0N-1S-2C(1=3O)(0-4C)	11	Nan	0	0
0N-1S=2O(1=3O)(0-4C)	11	Nan	0	0
0C-1Cl(0-2O)	11	1	0.5	0.66667
0C-1S(0-2S)	11	0	0	0
0O-1C-2Cl(0-3C)	11	1	0.5	0.66667
0S-1C-2S(0-3C)	11	Nan	0	0
0C-1P-2O-3C(0-4C)	11	1	1	1
0C-1O(0-2C-3F)(0-4C)	11	1	1	1
0C-1C-2P-3O(0=4C)	11	0.5	0.5	0.5
0C-1C(0-2C-3F)(0-4O)	11	1	1	1
0C-1C=2O(1-3N)(0-4O)	11	0.66667	1	0.8
0C-1C-2C-3Br(0=4O)	11	Nan	0	0
0C-1C-2F(1-3F)(0-4F)	11	0.66667	1	0.8
0C-1C-2Cl(1-3Cl)(0-4Cl)	11	0	0	0

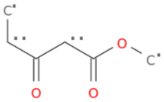
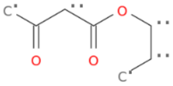
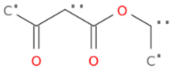
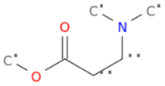
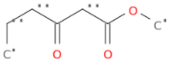
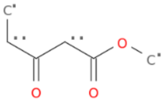
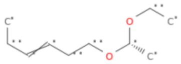
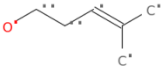
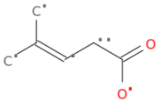
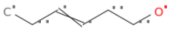
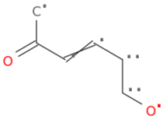
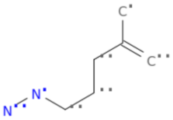
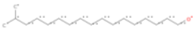
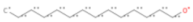
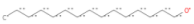
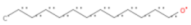
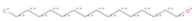
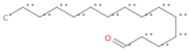
0C=1N(0=2O)	11	1	1	1
0P-1C(0-2C)(0-3C)	11	1	1	1
0N-1C=2N(0-3N)	11	Nan	0	0
0C-1P-2C(1-3C)(0-4C)	11	1	1	1
0C-1C-2C(1-3S)(0-4C)	11	Nan	0	0
0C=1C(0-2P=3O)(0-4C)	11	0.33333	1	0.5
0C=1C-2P=3O(0-4C)	11	0.5	1	0.66667
0C#1C-2C=3O(0-4C)	11	0	0	0
0N-1N=2C-3N(0-4C)	11	Nan	0	0
0C=1N-2C=3N(0-4O)	11	1	1	1
0N=1C-2C-3O(0-4C)	10	Nan	0	0
0C-1O-2C-3O(0-4O)	10	0	0	0
0C-1C-2C-3S(0-4N)	10	Nan	0	0
0C-1C(0-2C-3O)(0-4F)	10	1	1	1
0C-1N=2C-3Cl(0-4Cl)	10	1	0.5	0.66667
0C-1C(0-2C-3C)(0-4Br)	10	1	0.5	0.66667
0C-1O(0-2N)(0-3C)	10	Nan	0	0
0C-1O-2N(0=3C)	10	Nan	0	0
0C-1P-2O(0=3C)	10	0.33333	1	0.5
0P-1C(0-2C=3C)(0-4C)	10	1	1	1
0O-1C-2C(1-3N)(0-4C)	10	Nan	0	0
0N-1C-2C(1-3O)(0-4C)	10	Nan	0	0
0C-1C-2O(1-3N)(0-4C)	10	Nan	0	0
0C=1C-2O-3N(0-4C)	10	Nan	0	0
0C-1N(0-2C-3S)(0-4C)	10	0	0	0
0N-1N-2C-3N(0-4C)	10	Nan	0	0
0N=1C-2N-3N(0-4C)	10	Nan	0	0
0C-1O-2N=3C(0=4C)	10	Nan	0	0
0C-1N-2C-3N(0-4O)	10	Nan	0	0
0C-1N(0-2C-3Cl)(0=4O)	10	Nan	0	0
0C-1C-2C#3N(0-4N)	10	Nan	0	0
0N=1C=2O(0-3C)	9	1	1	1
0C-1C#2C(0=3C)	9	Nan	0	0
0C-1O(0=2O)(0-3O)	9	1	1	1
0C=1O(0-2Cl)(0-3O)	9	1	1	1
0C=1C(0-2O-3N)(0-4C)	9	Nan	0	0
0C=1N-2N=3C(0-4C)	9	Nan	0	0
0O-1C=2C(1-3N)(0-4C)	9	Nan	0	0
0O-1C-2C-3F(0-4C)	9	Nan	0	0

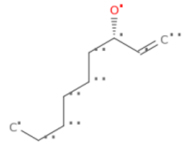
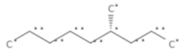
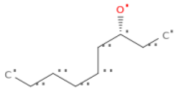
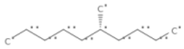
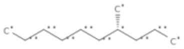
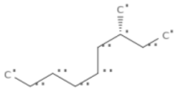
0C=1O(0-2C-3S)(0-4C)	9	1	1	1
0C-1P-2O(1=3O)(0=4C)	9	0.33333	1	0.5
0N-1C=2N-3N(0=4C)	9	Nan	0	0
0C=1N-2C-3C(0=4O)	9	1	1	1
0C-1C=2N-3N(0=4O)	9	Nan	0	0
0C=1C-2C-3S(0-4N)	9	0	0	0
0C-1C(0-2C-3O)(0=4N)	9	Nan	0	0
0C-1C=2C-3Cl(0=4N)	9	Nan	0	0
0C-1C(0-2C-3C)(0-4S)	9	Nan	0	0

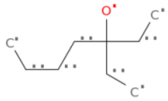
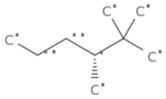
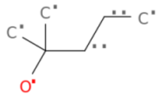
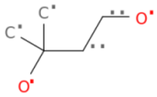
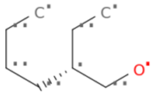
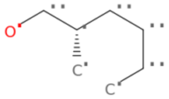
Appendix C. Structure prediction results.

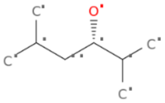
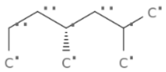
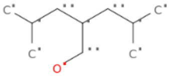
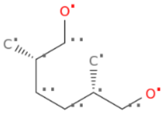
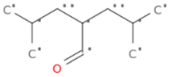
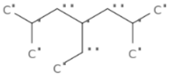
CAS No.: 61788-45-2	1	2	3	4	5
					
DLBS: 0.5831 Base_Truth	DLBS: 0.9715 SBSS: 0.7069	DLBS: 0.9701 SBSS: 0.7071	DLBS: 0.9691 SBSS: 0.7015	DLBS: 0.9688 SBSS: 0.5129	DLBS: 0.9678 SBSS: 0.5157
CAS No.: 60828-78-6	1	2	3	4	5
					
DLBS: 0.2106 Base_Truth	DLBS: 0.9685 SBSS: 0.346	DLBS: 0.9531 SBSS: 0.348	DLBS: 0.9474 SBSS: 0.3438	DLBS: 0.9308 SBSS: 0.3358	DLBS: 0.9164 SBSS: 0.3455
CAS No.: 57079-14-8	1	2	3	4	5
					
DLBS: 0.4282 Base_Truth	DLBS: 0.5308 SBSS: 0.1499	DLBS: 0.498 SBSS: 0.1578	DLBS: 0.4825 SBSS: 0.1415	DLBS: 0.4744 SBSS: 0.078	DLBS: 0.4373 SBSS: 0.1829
CAS No.: 42890-76-6	1	2	3	4	5
					
DLBS: 0.3855 Base_Truth	DLBS: 0.9348 SBSS: 0.4118	DLBS: 0.7767 SBSS: 0.471	DLBS: 0.59 SBSS: 0.4212	DLBS: 0.548 SBSS: 0.2786	DLBS: 0.5277 SBSS: 0.2072

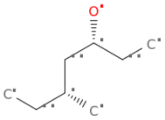
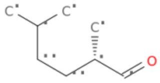
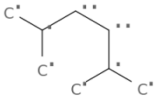
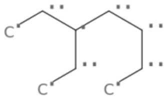
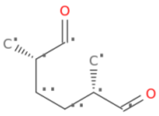
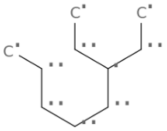
CAS No.: 42072-39-9	1	2	3	4	5
					
DLBS: 0.5372 Base_Truth	DLBS: 0.6778 SBSS: 1.0	DLBS: 0.5693 SBSS: 0.7595	DLBS: 0.5649 SBSS: 0.5211	DLBS: 0.513 SBSS: 0.4175	DLBS: 0.5046 SBSS: 0.4002
CAS No.: 35854-86-5	1	2	3	4	5
					
DLBS: 0.5354 Base_Truth	DLBS: 0.6937 SBSS: 0.303	DLBS: 0.6786 SBSS: 0.2756	DLBS: 0.6425 SBSS: 0.2351	DLBS: 0.6191 SBSS: 0.2099	DLBS: 0.6142 SBSS: 0.2414
CAS No.: 35154-45-1	1	2	3	4	5
					
DLBS: 0.3427 Base_Truth	DLBS: 0.6963 SBSS: 0.3852	DLBS: 0.6915 SBSS: 0.3827	DLBS: 0.6899 SBSS: 0.3854	DLBS: 0.6624 SBSS: 0.4649	DLBS: 0.6618 SBSS: 0.3702

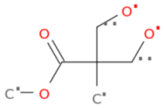
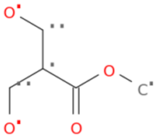
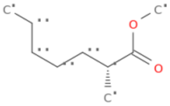
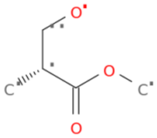
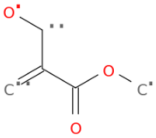
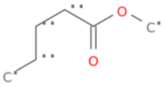
CAS No.: 30414-53-0	1	2	3	4	5
					
DLBS: 0.5332 Base_Truth	DLBS: 0.6997 SBSS: 0.4029	DLBS: 0.6732 SBSS: 0.4235	DLBS: 0.6413 SBSS: 0.4663	DLBS: 0.6178 SBSS: 0.7731	DLBS: 0.6117 SBSS: 1.0
CAS No.: 28069-74-1	1	2	3	4	5
					
DLBS: 0.1912 Base_Truth	DLBS: 0.5842 SBSS: 0.2294	DLBS: 0.581 SBSS: 0.1326	DLBS: 0.5467 SBSS: 0.4814	DLBS: 0.5364 SBSS: 0.2039	DLBS: 0.475 SBSS: 0.1197
CAS No.: 27458-93-1	1	2	3	4	5
					
DLBS: 0.3609 Base_Truth	DLBS: 0.9694 SBSS: 0.6253	DLBS: 0.9664 SBSS: 0.631	DLBS: 0.9562 SBSS: 0.6189	DLBS: 0.9475 SBSS: 0.636	DLBS: 0.9396 SBSS: 0.2263

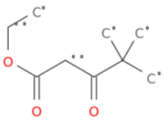
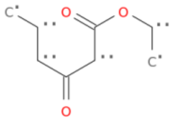
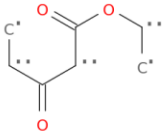
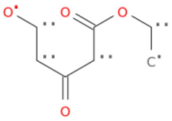
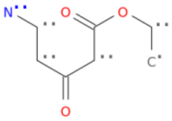
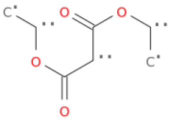
CAS No.: 21964-44-3	1	2	3	4	5
					
DLBS: 0.4394	DLBS: 0.83	DLBS: 0.8248	DLBS: 0.7706	DLBS: 0.7613	DLBS: 0.7598
Base_Truth	SBSS: 0.4078	SBSS: 0.6163	SBSS: 0.3903	SBSS: 0.4328	SBSS: 0.4331

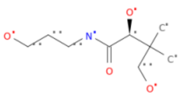
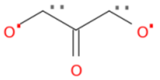
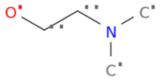
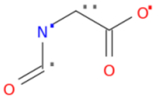
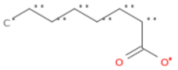
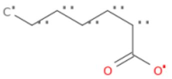
CAS No.: 19780-41-7	1	2	3	4	5
					
DLBS: 0.282	DLBS: 0.5711	DLBS: 0.5559	DLBS: 0.5252	DLBS: 0.522	DLBS: 0.5121
Base_Truth	SBSS: 0.266	SBSS: 0.451	SBSS: 0.3153	SBSS: 0.4068	SBSS: 0.4053

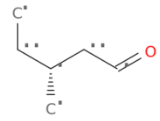
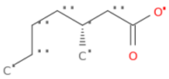
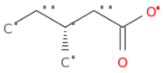
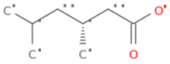
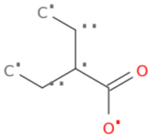
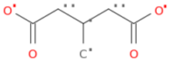
CAS No.: 19550-07-3	1	2	3	4	5
					
DLBS: 0.3432	DLBS: 0.7981	DLBS: 0.6916	DLBS: 0.655	DLBS: 0.6362	DLBS: 0.6176
Base_Truth	SBSS: 0.4205	SBSS: 0.5232	SBSS: 0.3056	SBSS: 0.4217	SBSS: 0.4545

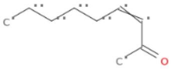
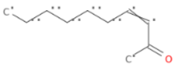
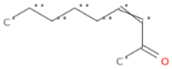
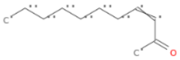
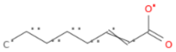
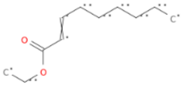
CAS No.: 18720-65-5	1	2	3	4	5
					
DLBS: 0.3517	DLBS: 0.8473	DLBS: 0.776	DLBS: 0.6894	DLBS: 0.671	DLBS: 0.6709
Base_Truth	SBSS: 0.2408	SBSS: 0.2752	SBSS: 0.3517	SBSS: 0.2648	SBSS: 0.3432

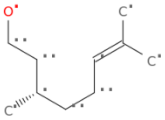
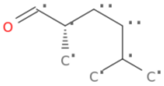
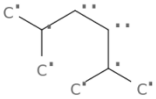
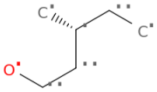
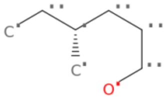
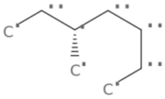
CAS No.: 17872-55-8	1	2	3	4	5
					
DLBS: 0.2543	DLBS: 0.4751	DLBS: 0.4447	DLBS: 0.4367	DLBS: 0.4364	DLBS: 0.4348
Base_Truth	SBSS: 0.4635	SBSS: 0.3275	SBSS: 0.4395	SBSS: 0.4307	SBSS: 0.409

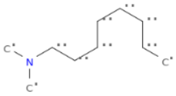
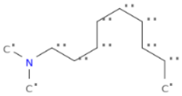
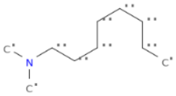
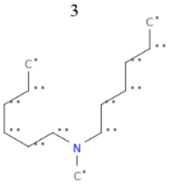
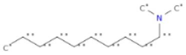
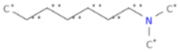
CAS No.: 17094-34-7	1	2	3	4	5
					
DLBS: 0.4138	DLBS: 0.7381	DLBS: 0.6676	DLBS: 0.6319	DLBS: 0.6317	DLBS: 0.6282
Base_Truth	SBSS: 0.6409	SBSS: 0.6544	SBSS: 0.6373	SBSS: 0.5998	SBSS: 0.6885

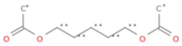
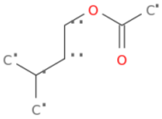
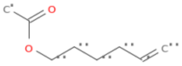
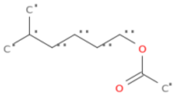
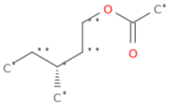
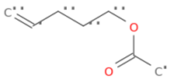
CAS No.: 16485-10-2	1	2	3	4	5
					
DLBS: 0.274	DLBS: 0.4207	DLBS: 0.3926	DLBS: 0.3895	DLBS: 0.3653	DLBS: 0.365
Base_Truth	SBSS: 0.235	SBSS: 0.2249	SBSS: 0.237	SBSS: 0.2332	SBSS: 0.2342

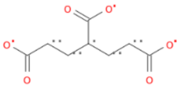
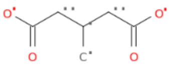
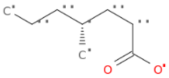
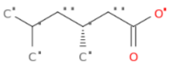
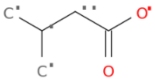
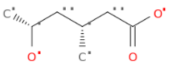
CAS No.: 15877-57-3	1	2	3	4	5
					
DLBS: 0.3802 Base_Truth	DLBS: 0.5715 SBSS: 0.3285	DLBS: 0.5599 SBSS: 0.5063	DLBS: 0.5528 SBSS: 0.3155	DLBS: 0.5527 SBSS: 0.3596	DLBS: 0.5428 SBSS: 0.3586

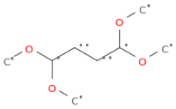
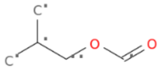
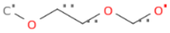
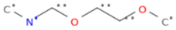
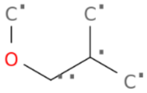
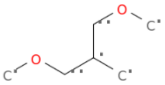
CAS No.: 14309-57-0	1	2	3	4	5
					
DLBS: 0.6284 Base_Truth	DLBS: 0.9635 SBSS: 0.9233	DLBS: 0.8964 SBSS: 1.0	DLBS: 0.8902 SBSS: 0.899	DLBS: 0.8379 SBSS: 0.6961	DLBS: 0.816 SBSS: 0.5852

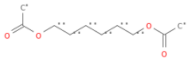
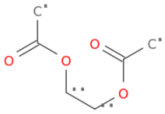
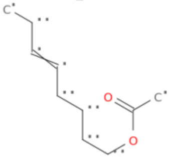
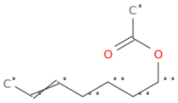
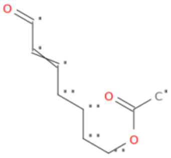
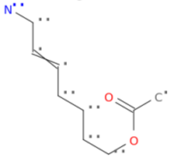
CAS No.: 7540-51-4	1	2	3	4	5
					
DLBS: 0.3285 Base_Truth	DLBS: 0.7616 SBSS: 0.2767	DLBS: 0.7296 SBSS: 0.2768	DLBS: 0.5716 SBSS: 0.5463	DLBS: 0.5629 SBSS: 0.4368	DLBS: 0.5557 SBSS: 0.2903

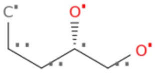
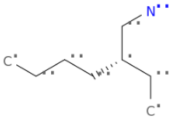
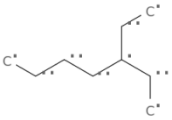
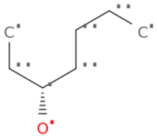
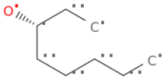
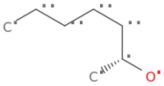
CAS No.: 7378-99-6	1	2	3	4	5
					
DLBS: 0.559 Base_Truth	DLBS: 0.8387 SBSS: 0.9836	DLBS: 0.8359 SBSS: 1.0	DLBS: 0.8019 SBSS: 0.6861	DLBS: 0.7988 SBSS: 0.9677	DLBS: 0.7939 SBSS: 0.9691

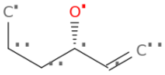
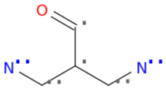
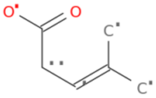
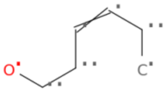
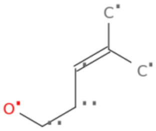
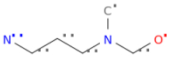
CAS No.: 6963-44-6	1	2	3	4	5
					
DLBS: 0.5248 Base_Truth	DLBS: 0.617 SBSS: 0.6573	DLBS: 0.606 SBSS: 0.7008	DLBS: 0.6009 SBSS: 0.774	DLBS: 0.597 SBSS: 0.6013	DLBS: 0.5928 SBSS: 0.6645

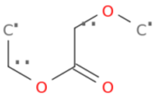
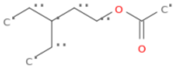
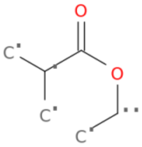
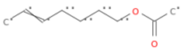
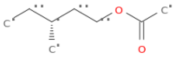
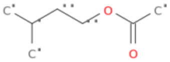
CAS No.: 6940-58-5	1	2	3	4	5
					
DLBS: 0.3785 Base_Truth	DLBS: 0.6842 SBSS: 0.4634	DLBS: 0.6604 SBSS: 0.5954	DLBS: 0.6395 SBSS: 0.4459	DLBS: 0.6318 SBSS: 0.4289	DLBS: 0.6279 SBSS: 0.3985

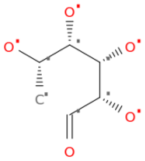
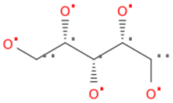
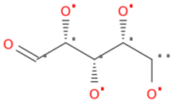
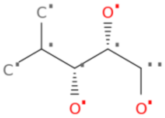
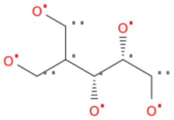
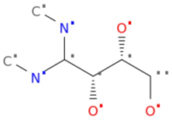
CAS No.: 6922-39-0	1	2	3	4	5
					
DLBS: 0.2367 Base_Truth	DLBS: 0.4 SBSS: 0.2466	DLBS: 0.3786 SBSS: 0.2474	DLBS: 0.3693 SBSS: 0.2143	DLBS: 0.3613 SBSS: 0.3315	DLBS: 0.3599 SBSS: 0.3381

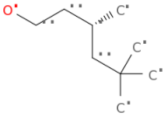
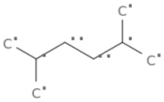
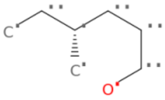
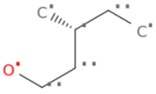
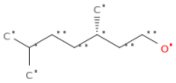
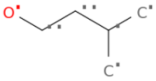
CAS No.: 6222-17-9	1	2	3	4	5
					
DLBS: 0.5885 Base_Truth	DLBS: 0.7864 SBSS: 0.7234	DLBS: 0.7237 SBSS: 0.6845	DLBS: 0.7045 SBSS: 0.7196	DLBS: 0.6785 SBSS: 0.6989	DLBS: 0.6771 SBSS: 0.6819

CAS No.: 5343-92-0	1	2	3	4	5
					
DLBS: 0.5259 Base_Truth	DLBS: 0.7773 SBSS: 0.2612	DLBS: 0.7089 SBSS: 0.3429	DLBS: 0.6605 SBSS: 0.4792	DLBS: 0.6333 SBSS: 0.4282	DLBS: 0.6229 SBSS: 0.43

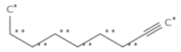
CAS No.: 4798-44-1	1	2	3	4	5
					
DLBS: 0.346 Base_Truth	DLBS: 0.5429 SBSS: 0.1407	DLBS: 0.4988 SBSS: 0.1913	DLBS: 0.4941 SBSS: 0.2545	DLBS: 0.4923 SBSS: 0.2148	DLBS: 0.4785 SBSS: 0.1311

CAS No.: 3938-96-3	1	2	3	4	5
					
DLBS: 0.4301 Base_Truth	DLBS: 0.5928 SBSS: 0.3106	DLBS: 0.5552 SBSS: 0.4819	DLBS: 0.5482 SBSS: 0.2926	DLBS: 0.5472 SBSS: 0.2735	DLBS: 0.5436 SBSS: 0.2969

CAS No.: 3615-37-0	1	2	3	4	5
					
DLBS: 0.2116 Base_Truth	DLBS: 0.7872 SBSS: 0.2517	DLBS: 0.6958 SBSS: 0.6049	DLBS: 0.688 SBSS: 0.301	DLBS: 0.6865 SBSS: 0.227	DLBS: 0.6136 SBSS: 0.2345

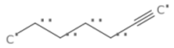
CAS No.: 3452-97-9	1	2	3	4	5
					
DLBS: 0.3133 Base_Truth	DLBS: 0.6901 SBSS: 0.3256	DLBS: 0.5556 SBSS: 0.4998	DLBS: 0.541 SBSS: 0.6306	DLBS: 0.5369 SBSS: 0.6559	DLBS: 0.5364 SBSS: 0.5243

CAS No.: 3452-09-3



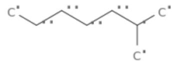
DLBS: 0.5577
Base_Truth

1



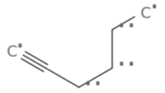
DLBS: 0.9888
SBSS: 0.8609

2



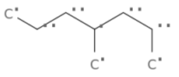
DLBS: 0.8894
SBSS: 0.441

3



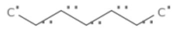
DLBS: 0.8387
SBSS: 0.7459

4



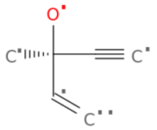
DLBS: 0.8083
SBSS: 0.3177

5



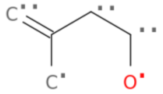
DLBS: 0.804
SBSS: 0.6105

CAS No.: 3230-69-1



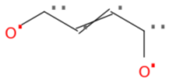
DLBS: 0.3438
Base_Truth

1



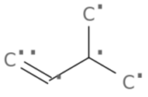
DLBS: 0.7862
SBSS: 0.1586

2



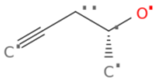
DLBS: 0.7537
SBSS: 0.1352

3



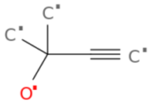
DLBS: 0.7279
SBSS: 0.2286

4



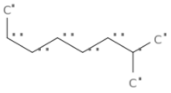
DLBS: 0.7081
SBSS: 0.283

5



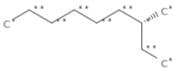
DLBS: 0.6255
SBSS: 0.49

CAS No.: 3221-61-2



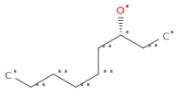
DLBS: 0.6023
Base_Truth

1



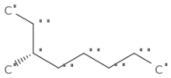
DLBS: 0.9653
SBSS: 0.6853

2



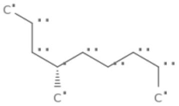
DLBS: 0.9432
SBSS: 0.5458

3



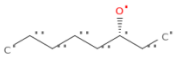
DLBS: 0.9367
SBSS: 0.6165

4



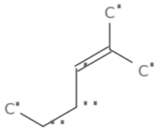
DLBS: 0.9281
SBSS: 0.6416

5



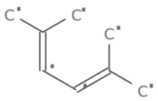
DLBS: 0.8816
SBSS: 0.4909

CAS No.: 2738-19-4



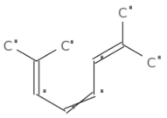
DLBS: 0.5494
Base_Truth

1



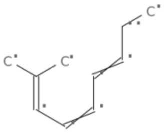
DLBS: 0.9752
SBSS: 0.4183

2



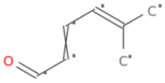
DLBS: 0.9745
SBSS: 0.3827

3



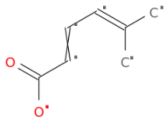
DLBS: 0.9065
SBSS: 0.3823

4



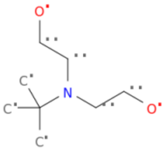
DLBS: 0.8564
SBSS: 0.3134

5



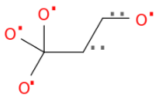
DLBS: 0.8539
SBSS: 0.2741

CAS No.: 2160-93-2



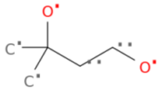
DLBS: 0.2636
Base_Truth

1



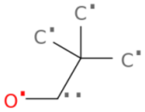
DLBS: 0.4339
SBSS: 0.2944

2



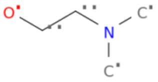
DLBS: 0.4254
SBSS: 0.3458

3



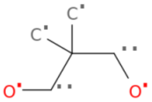
DLBS: 0.4102
SBSS: 0.3313

4



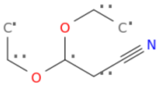
DLBS: 0.3935
SBSS: 0.5077

5



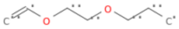
DLBS: 0.3876
SBSS: 0.3347

CAS No.: 2032-34-0



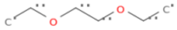
DLBS: 0.2129
Base_Truth

1



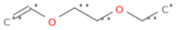
DLBS: 0.5235
SBSS: 0.1869

2



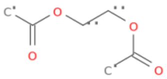
DLBS: 0.4936
SBSS: 0.3419

3



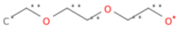
DLBS: 0.4924
SBSS: 0.2603

4



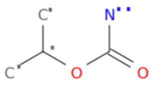
DLBS: 0.4862
SBSS: 0.1645

5



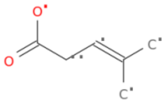
DLBS: 0.482
SBSS: 0.2787

CAS No.: 1746-77-6



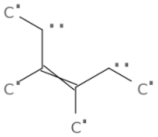
DLBS: 0.282
Base_Truth

1



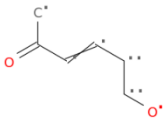
DLBS: 0.6441
SBSS: 0.1869

2



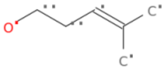
DLBS: 0.6258
SBSS: 0.1082

3



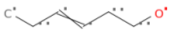
DLBS: 0.6034
SBSS: 0.1797

4



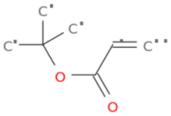
DLBS: 0.5923
SBSS: 0.1571

5



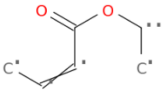
DLBS: 0.5488
SBSS: 0.1091

CAS No.: 1663-39-4



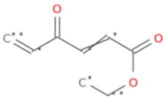
DLBS: 0.3611
Base_Truth

1



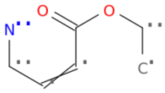
DLBS: 0.689
SBSS: 0.3889

2



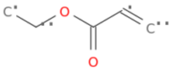
DLBS: 0.6861
SBSS: 0.4451

3



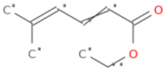
DLBS: 0.6855
SBSS: 0.3282

4



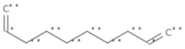
DLBS: 0.6658
SBSS: 0.5677

5



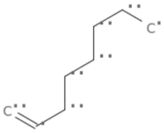
DLBS: 0.6605
SBSS: 0.3284

CAS No.: 1647-16-1



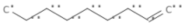
DLBS: 0.6662
Base_Truth

1



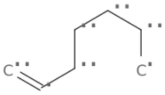
DLBS: 0.9537
SBSS: 0.7962

2



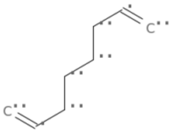
DLBS: 0.9084
SBSS: 0.8352

3



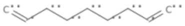
DLBS: 0.8971
SBSS: 0.7304

4



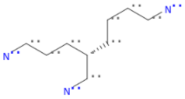
DLBS: 0.8558
SBSS: 0.875

5



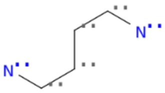
DLBS: 0.8078
SBSS: 0.9412

CAS No.: 1572-55-0



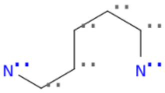
DLBS: 0.326
Base_Truth

1



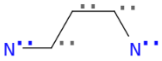
DLBS: 0.6839
SBSS: 0.5107

2



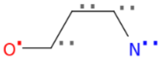
DLBS: 0.6751
SBSS: 0.5156

3



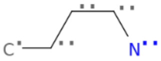
DLBS: 0.6577
SBSS: 0.4019

4



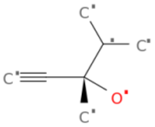
DLBS: 0.631
SBSS: 0.3329

5



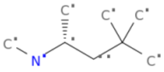
DLBS: 0.6241
SBSS: 0.3738

CAS No.: 1482-15-1



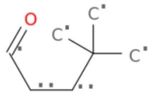
DLBS: 0.2085
Base_Truth

1



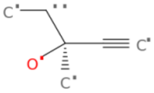
DLBS: 0.5998
SBSS: 0.2482

2



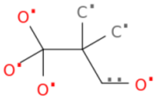
DLBS: 0.5483
SBSS: 0.118

3



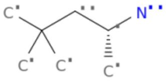
DLBS: 0.5392
SBSS: 0.4304

4



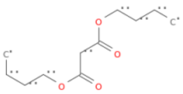
DLBS: 0.5123
SBSS: 0.2171

5



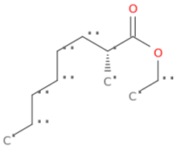
DLBS: 0.4812
SBSS: 0.1973

CAS No.: 1190-39-2



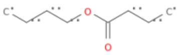
DLBS: 0.4751
Base_Truth

1



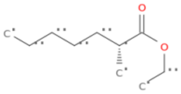
DLBS: 0.7063
SBSS: 0.4376

2



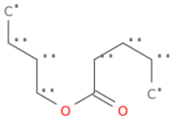
DLBS: 0.6908
SBSS: 0.7257

3



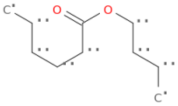
DLBS: 0.6827
SBSS: 0.4449

4



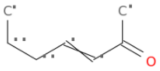
DLBS: 0.6806
SBSS: 0.7267

5



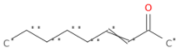
DLBS: 0.675
SBSS: 0.7134

CAS No.: 1119-44-4



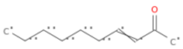
DLBS: 0.3939
Base_Truth

1



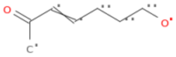
DLBS: 0.7028
SBSS: 0.7598

2



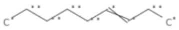
DLBS: 0.6079
SBSS: 0.7298

3



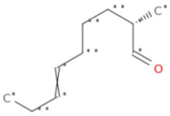
DLBS: 0.6037
SBSS: 0.6704

4



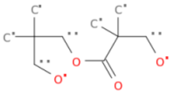
DLBS: 0.5939
SBSS: 0.3588

5



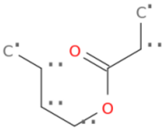
DLBS: 0.578
SBSS: 0.2797

CAS No.: 1115-20-4



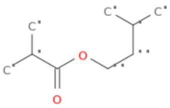
DLBS: 0.2406
Base_Truth

1



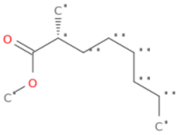
DLBS: 0.4715
SBSS: 0.304

2



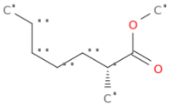
DLBS: 0.4576
SBSS: 0.279

3



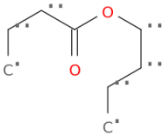
DLBS: 0.4559
SBSS: 0.2501

4

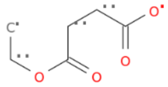
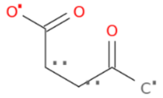
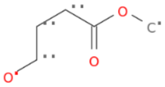
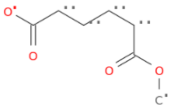
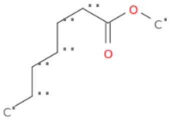
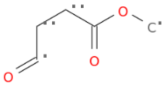
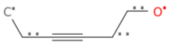
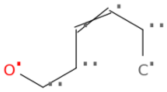
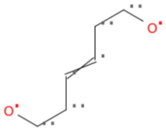
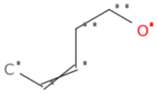
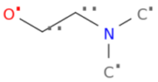
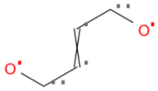
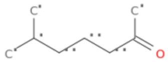
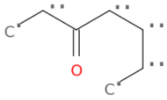
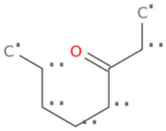
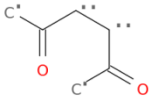
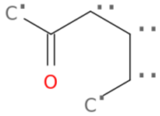
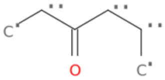
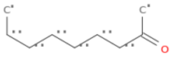
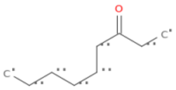
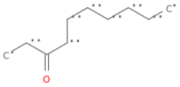
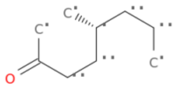
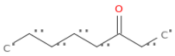
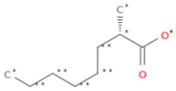
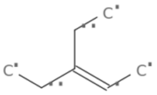
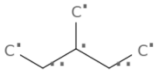
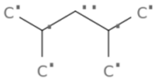
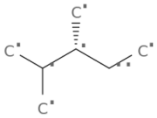
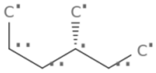
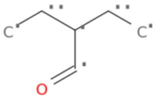
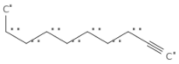
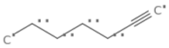
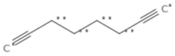
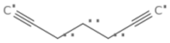
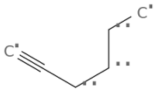
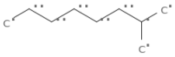
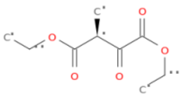
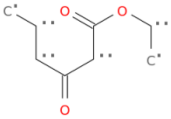
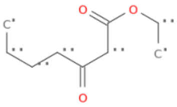
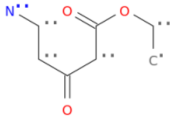
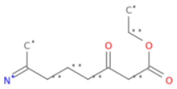
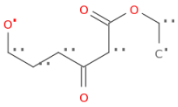


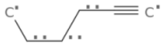
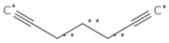
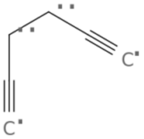
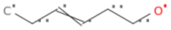
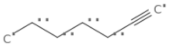
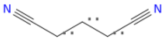
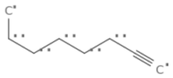
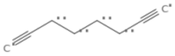
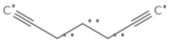
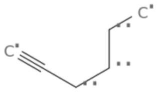
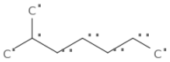
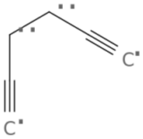
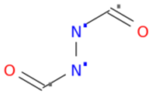
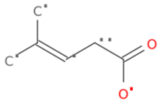
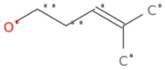
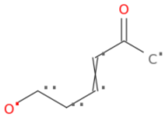
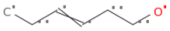
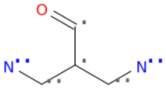
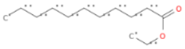
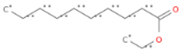
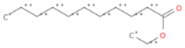
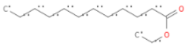
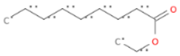
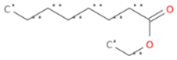
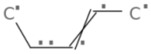
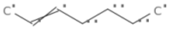
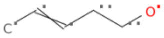
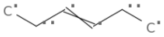
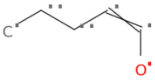
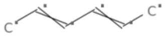
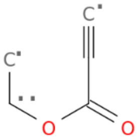
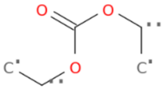
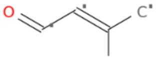
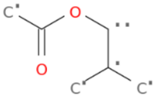
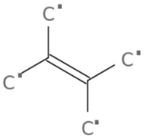
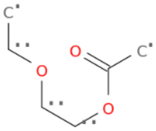
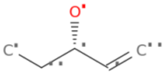
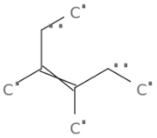
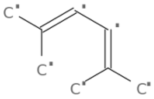
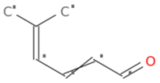
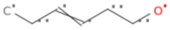
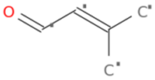
DLBS: 0.4509
SBSS: 0.2489

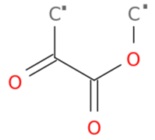
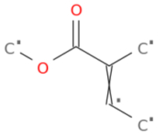
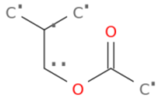
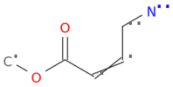
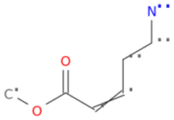
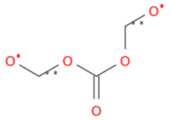
5

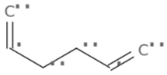
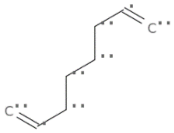
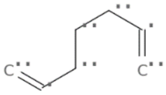
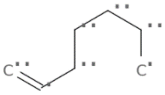
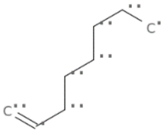
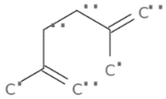


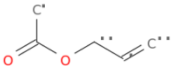
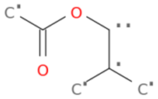
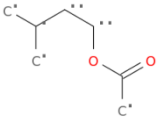
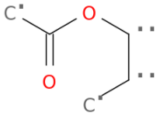
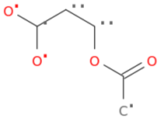
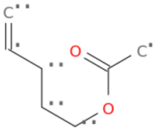
DLBS: 0.4475
SBSS: 0.2727

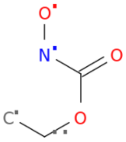
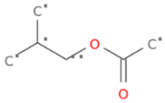
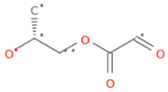
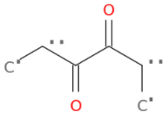
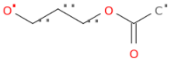
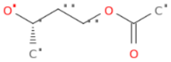
CAS No.: 1070-34-4	1	2	3	4	5
					
DLBS: 0.4889 Base_Truth	DLBS: 0.6735 SBSS: 0.568	DLBS: 0.6708 SBSS: 0.3952	DLBS: 0.6686 SBSS: 0.5848	DLBS: 0.6661 SBSS: 0.381	DLBS: 0.6615 SBSS: 0.3662
CAS No.: 1002-28-4	1	2	3	4	5
					
DLBS: 0.3261 Base_Truth	DLBS: 0.6928 SBSS: 0.3344	DLBS: 0.5359 SBSS: 0.2996	DLBS: 0.5353 SBSS: 0.3013	DLBS: 0.5257 SBSS: 0.2797	DLBS: 0.5035 SBSS: 0.1992
CAS No.: 928-68-7	1	2	3	4	5
					
DLBS: 0.4434 Base_Truth	DLBS: 0.8385 SBSS: 0.3636	DLBS: 0.7606 SBSS: 0.3559	DLBS: 0.7403 SBSS: 0.6199	DLBS: 0.7217 SBSS: 0.6424	DLBS: 0.7201 SBSS: 0.3476
CAS No.: 821-55-6	1	2	3	4	5
					
DLBS: 0.5942 Base_Truth	DLBS: 0.9507 SBSS: 0.6476	DLBS: 0.9155 SBSS: 0.6876	DLBS: 0.8728 SBSS: 0.6131	DLBS: 0.868 SBSS: 0.5952	DLBS: 0.8559 SBSS: 0.4862
CAS No.: 816-79-5	1	2	3	4	5
					
DLBS: 0.2875 Base_Truth	DLBS: 0.8223 SBSS: 0.2171	DLBS: 0.8183 SBSS: 0.2355	DLBS: 0.7778 SBSS: 0.1925	DLBS: 0.7258 SBSS: 0.1809	DLBS: 0.7221 SBSS: 0.2292
CAS No.: 764-93-2	1	2	3	4	5
					
DLBS: 0.5485 Base_Truth	DLBS: 0.9878 SBSS: 0.8302	DLBS: 0.8629 SBSS: 0.6811	DLBS: 0.817 SBSS: 0.5902	DLBS: 0.8096 SBSS: 0.7181	DLBS: 0.8003 SBSS: 0.483
CAS No.: 759-65-9	1	2	3	4	5
					
DLBS: 0.3698 Base_Truth	DLBS: 0.7796 SBSS: 0.3781	DLBS: 0.7653 SBSS: 0.3682	DLBS: 0.6936 SBSS: 0.3574	DLBS: 0.6668 SBSS: 0.3088	DLBS: 0.6633 SBSS: 0.3323

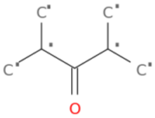
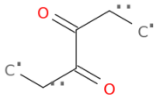
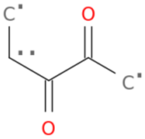
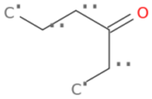
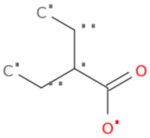
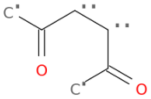
CAS No.: 693-02-7	1	2	3	4	5
					
DLBS: 0.5793 Base_Truth	DLBS: 0.8581 SBSS: 0.7171	DLBS: 0.8014 SBSS: 0.5886	DLBS: 0.7696 SBSS: 0.1456	DLBS: 0.754 SBSS: 0.8207	DLBS: 0.7091 SBSS: 0.3118
CAS No.: 629-05-0	1	2	3	4	5
					
DLBS: 0.5622 Base_Truth	DLBS: 0.8379 SBSS: 0.7326	DLBS: 0.8358 SBSS: 0.636	DLBS: 0.8307 SBSS: 0.7759	DLBS: 0.8185 SBSS: 0.4413	DLBS: 0.7353 SBSS: 0.518
CAS No.: 628-36-4	1	2	3	4	5
					
DLBS: 0.2238 Base_Truth	DLBS: 0.6052 SBSS: 0.1085	DLBS: 0.5836 SBSS: 0.0644	DLBS: 0.5559 SBSS: 0.1075	DLBS: 0.553 SBSS: 0.0629	DLBS: 0.5286 SBSS: 0.2108
CAS No.: 627-90-7	1	2	3	4	5
					
DLBS: 0.7791 Base_Truth	DLBS: 0.9897 SBSS: 0.9873	DLBS: 0.9844 SBSS: 1.0	DLBS: 0.9729 SBSS: 0.9882	DLBS: 0.9662 SBSS: 0.973	DLBS: 0.9319 SBSS: 0.9476
CAS No.: 627-20-3	1	2	3	4	5
					
DLBS: 0.7542 Base_Truth	DLBS: 0.849 SBSS: 0.5314	DLBS: 0.7583 SBSS: 0.5103	DLBS: 0.7416 SBSS: 0.6729	DLBS: 0.7379 SBSS: 0.3013	DLBS: 0.736 SBSS: 0.3976
CAS No.: 623-47-2	1	2	3	4	5
					
DLBS: 0.5065 Base_Truth	DLBS: 0.686 SBSS: 0.5043	DLBS: 0.6309 SBSS: 0.1532	DLBS: 0.6239 SBSS: 0.3251	DLBS: 0.6183 SBSS: 0.1003	DLBS: 0.5984 SBSS: 0.4066
CAS No.: 616-25-1	1	2	3	4	5
					
DLBS: 0.4101 Base_Truth	DLBS: 0.8766 SBSS: 0.1981	DLBS: 0.8391 SBSS: 0.1447	DLBS: 0.7827 SBSS: 0.1144	DLBS: 0.7495 SBSS: 0.273	DLBS: 0.7227 SBSS: 0.1253

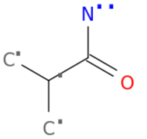
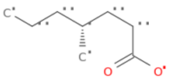
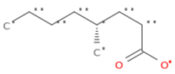
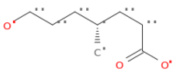
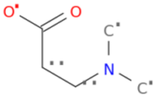
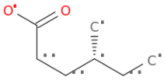
CAS No.: 600-22-6	1	2	3	4	5
					
DLBS: 0.3624	DLBS: 0.5734	DLBS: 0.5624	DLBS: 0.5285	DLBS: 0.5126	DLBS: 0.5118
Base_Truth	SBSS: 0.5105	SBSS: 0.3088	SBSS: 0.3763	SBSS: 0.3644	SBSS: 0.2212

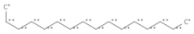
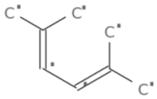
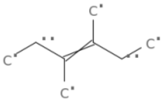
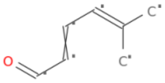
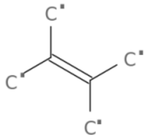
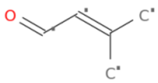
CAS No.: 592-42-7	1	2	3	4	5
					
DLBS: 0.6073	DLBS: 0.933	DLBS: 0.8875	DLBS: 0.7717	DLBS: 0.7182	DLBS: 0.7121
Base_Truth	SBSS: 0.7027	SBSS: 0.7196	SBSS: 0.5625	SBSS: 0.5282	SBSS: 0.2014

CAS No.: 591-87-7	1	2	3	4	5
					
DLBS: 0.6256	DLBS: 0.7498	DLBS: 0.6704	DLBS: 0.6579	DLBS: 0.6535	DLBS: 0.6408
Base_Truth	SBSS: 0.5512	SBSS: 0.5058	SBSS: 0.573	SBSS: 0.4997	SBSS: 0.6453

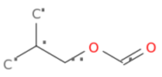
CAS No.: 589-41-3	1	2	3	4	5
					
DLBS: 0.3904	DLBS: 0.5979	DLBS: 0.5362	DLBS: 0.5125	DLBS: 0.495	DLBS: 0.4703
Base_Truth	SBSS: 0.3012	SBSS: 0.2908	SBSS: 0.2184	SBSS: 0.329	SBSS: 0.2949

CAS No.: 565-80-0	1	2	3	4	5
					
DLBS: 0.2686	DLBS: 0.7032	DLBS: 0.6005	DLBS: 0.5789	DLBS: 0.5559	DLBS: 0.5507
Base_Truth	SBSS: 0.3805	SBSS: 0.3119	SBSS: 0.2777	SBSS: 0.3121	SBSS: 0.3174

CAS No.: 563-83-7	1	2	3	4	5
					
DLBS: 0.4024	DLBS: 0.6722	DLBS: 0.5979	DLBS: 0.5531	DLBS: 0.5481	DLBS: 0.5328
Base_Truth	SBSS: 0.2456	SBSS: 0.2061	SBSS: 0.237	SBSS: 0.2203	SBSS: 0.2549

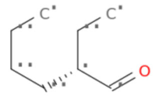
CAS No.: 544-76-3	1	2	3	4	5
					
DLBS: 0.5001	DLBS: 0.8732	DLBS: 0.8504	DLBS: 0.8151	DLBS: 0.8137	DLBS: 0.7988
Base_Truth	SBSS: 0.0993	SBSS: 0.236	SBSS: 0.0706	SBSS: 0.1147	SBSS: 0.0773

CAS No.: 542-55-2



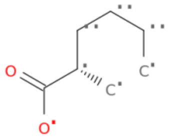
DLBS: 0.4384
Base_Truth

1



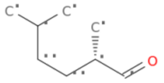
DLBS: 0.8231
SBSS: 0.246

2



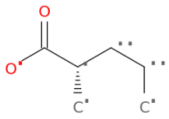
DLBS: 0.7839
SBSS: 0.2646

3



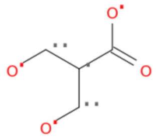
DLBS: 0.7761
SBSS: 0.3258

4



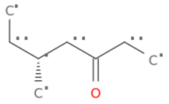
DLBS: 0.7555
SBSS: 0.2397

5



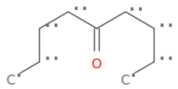
DLBS: 0.746
SBSS: 0.2276

CAS No.: 541-85-5



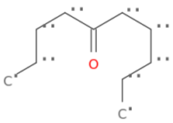
DLBS: 0.3569
Base_Truth

1



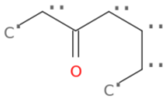
DLBS: 0.8974
SBSS: 0.3906

2



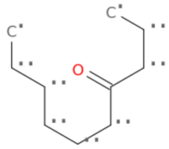
DLBS: 0.804
SBSS: 0.3532

3



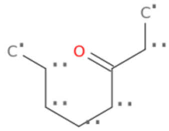
DLBS: 0.7858
SBSS: 0.5051

4



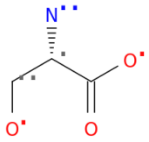
DLBS: 0.782
SBSS: 0.3431

5



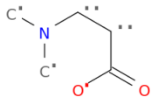
DLBS: 0.7709
SBSS: 0.4971

CAS No.: 302-84-1



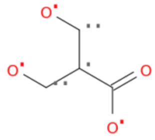
DLBS: 0.3774
Base_Truth

1



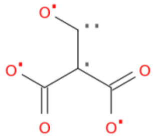
DLBS: 0.6159
SBSS: 0.2899

2



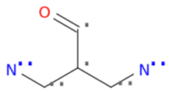
DLBS: 0.5543
SBSS: 0.567

3



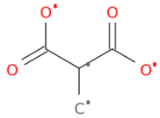
DLBS: 0.5272
SBSS: 0.5491

4



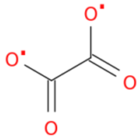
DLBS: 0.523
SBSS: 0.2187

5



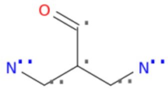
DLBS: 0.5055
SBSS: 0.4468

CAS No.: 144-62-7



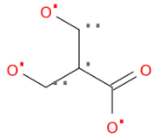
DLBS: 0.4225
Base_Truth

1



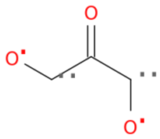
DLBS: 0.5484
SBSS: 0.0826

2



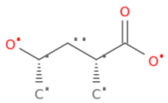
DLBS: 0.4984
SBSS: 0.3225

3



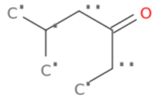
DLBS: 0.4611
SBSS: 0.3321

4



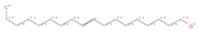
DLBS: 0.4437
SBSS: 0.2657

5



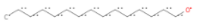
DLBS: 0.4306
SBSS: 0.1796

CAS No.: 143-28-2



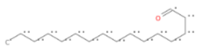
DLBS: 0.5353
Base_Truth

1



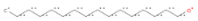
DLBS: 0.8599
SBSS: 0.7801

2



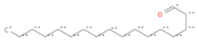
DLBS: 0.8566
SBSS: 0.5042

3



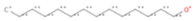
DLBS: 0.8516
SBSS: 0.7878

4



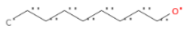
DLBS: 0.8472
SBSS: 0.5073

5



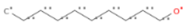
DLBS: 0.8432
SBSS: 0.7715

CAS No.: 143-08-8



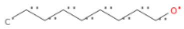
DLBS: 0.7652
Base_Truth

1



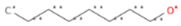
DLBS: 0.9792
SBSS: 0.9818

2



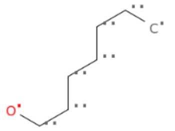
DLBS: 0.9753
SBSS: 1.0

3



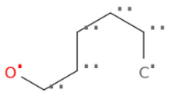
DLBS: 0.9164
SBSS: 0.9796

4

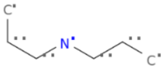
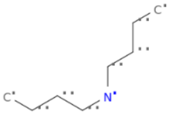
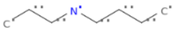
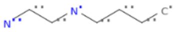
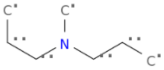
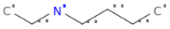
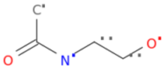
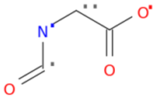
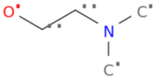
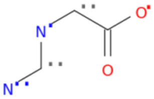
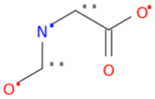
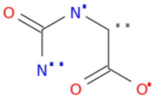
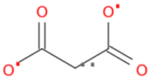
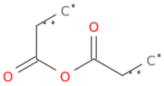
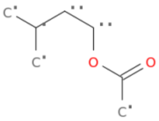
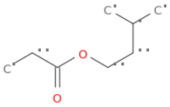
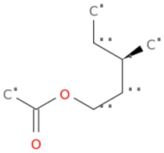
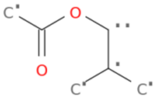
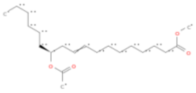
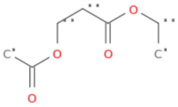
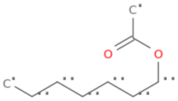
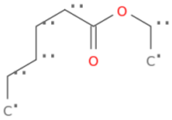
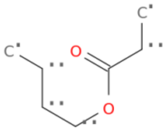
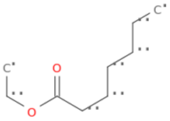
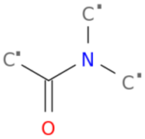
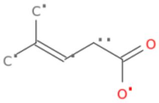
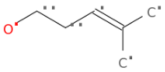
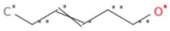
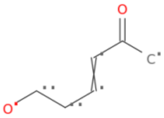
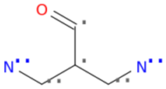
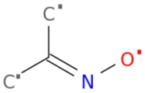
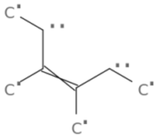
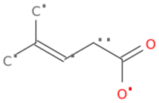
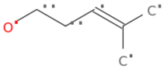
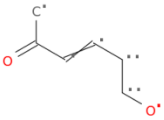
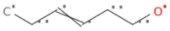
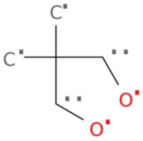
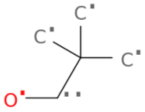
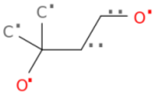
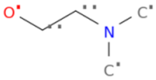
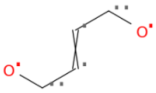
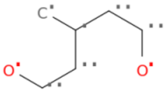


DLBS: 0.856
SBSS: 0.9403

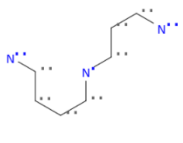
5



DLBS: 0.7874
SBSS: 0.8669

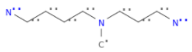
CAS No.: 142-84-7	1	2	3	4	5
					
DLBS: 0.7047 Base_Truth	DLBS: 0.9031 SBSS: 0.5986	DLBS: 0.8952 SBSS: 0.775	DLBS: 0.8583 SBSS: 0.5357	DLBS: 0.803 SBSS: 0.4066	DLBS: 0.7979 SBSS: 0.5484
CAS No.: 142-26-7	1	2	3	4	5
					
DLBS: 0.3651 Base_Truth	DLBS: 0.4052 SBSS: 0.2532	DLBS: 0.3814 SBSS: 0.378	DLBS: 0.3657 SBSS: 0.258	DLBS: 0.3619 SBSS: 0.3366	DLBS: 0.3603 SBSS: 0.2838
CAS No.: 141-82-2	1	2	3	4	5
					
DLBS: 0.367 Base_Truth	DLBS: 0.7272 SBSS: 0.2828	DLBS: 0.5949 SBSS: 0.2637	DLBS: 0.5796 SBSS: 0.2098	DLBS: 0.5618 SBSS: 0.2389	DLBS: 0.5524 SBSS: 0.2411
CAS No.: 140-03-4	1	2	3	4	5
					
DLBS: 0.3452 Base_Truth	DLBS: 0.5811 SBSS: 0.3343	DLBS: 0.581 SBSS: 0.4099	DLBS: 0.5742 SBSS: 0.4317	DLBS: 0.5734 SBSS: 0.3235	DLBS: 0.5715 SBSS: 0.4611
CAS No.: 127-19-5	1	2	3	4	5
					
DLBS: 0.5776 Base_Truth	DLBS: 0.6046 SBSS: 0.234	DLBS: 0.592 SBSS: 0.138	DLBS: 0.5563 SBSS: 0.0594	DLBS: 0.5432 SBSS: 0.2097	DLBS: 0.5411 SBSS: 0.0725
CAS No.: 127-06-0	1	2	3	4	5
					
DLBS: 0.5096 Base_Truth	DLBS: 0.6351 SBSS: 0.1761	DLBS: 0.6125 SBSS: 0.1802	DLBS: 0.5983 SBSS: 0.2097	DLBS: 0.562 SBSS: 0.2085	DLBS: 0.5604 SBSS: 0.1225
CAS No.: 126-30-7	1	2	3	4	5
					
DLBS: 0.3931 Base_Truth	DLBS: 0.5592 SBSS: 0.589	DLBS: 0.5414 SBSS: 0.3924	DLBS: 0.4842 SBSS: 0.2516	DLBS: 0.4572 SBSS: 0.2554	DLBS: 0.4554 SBSS: 0.3031

CAS No.: 124-20-9



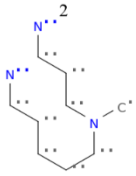
DLBS: 0.6921
Base_Truth

1



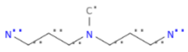
DLBS: 0.9418
SBSS: 0.4816

2



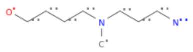
DLBS: 0.9187
SBSS: 0.4601

3



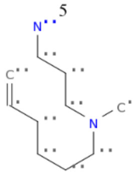
DLBS: 0.9078
SBSS: 0.437

4



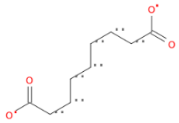
DLBS: 0.8061
SBSS: 0.3616

5



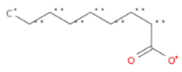
DLBS: 0.7966
SBSS: 0.3438

CAS No.: 123-99-9



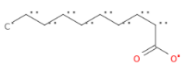
DLBS: 0.628
Base_Truth

1



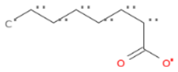
DLBS: 0.8992
SBSS: 0.8389

2



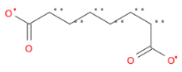
DLBS: 0.8912
SBSS: 0.84

3



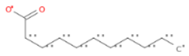
DLBS: 0.877
SBSS: 0.8235

4



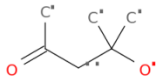
DLBS: 0.8566
SBSS: 0.9851

5



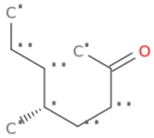
DLBS: 0.8484
SBSS: 0.8289

CAS No.: 123-42-2



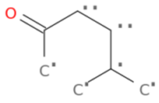
DLBS: 0.3153
Base_Truth

1



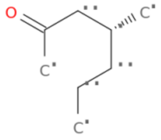
DLBS: 0.4632
SBSS: 0.3957

2



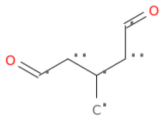
DLBS: 0.4322
SBSS: 0.4561

3



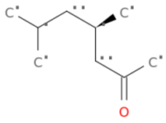
DLBS: 0.4201
SBSS: 0.3883

4



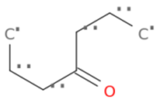
DLBS: 0.4186
SBSS: 0.1767

5



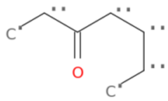
DLBS: 0.4173
SBSS: 0.4237

CAS No.: 123-19-3



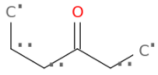
DLBS: 0.5613
Base_Truth

1



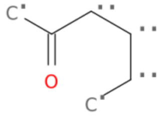
DLBS: 0.9881
SBSS: 0.5893

2



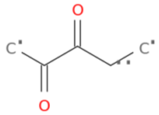
DLBS: 0.8454
SBSS: 0.7613

3



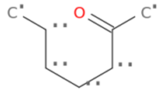
DLBS: 0.7654
SBSS: 0.508

4



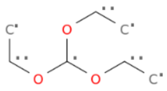
DLBS: 0.7325
SBSS: 0.358

5



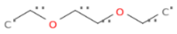
DLBS: 0.7259
SBSS: 0.4963

CAS No.: 122-51-0



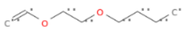
DLBS: 0.2789
Base_Truth

1



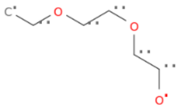
DLBS: 0.6333
SBSS: 0.4462

2



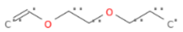
DLBS: 0.564
SBSS: 0.2089

3



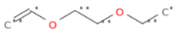
DLBS: 0.5621
SBSS: 0.3471

4



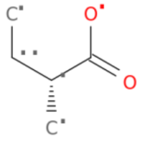
DLBS: 0.5501
SBSS: 0.217

5



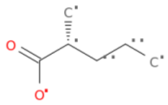
DLBS: 0.5498
SBSS: 0.3191

CAS No.: 116-53-0



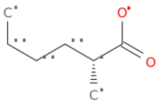
DLBS: 0.6251
Base_Truth

1



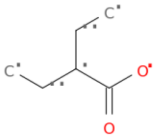
DLBS: 0.976
SBSS: 0.6674

2



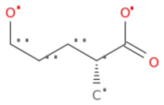
DLBS: 0.9198
SBSS: 0.6309

3



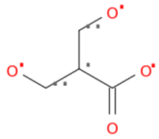
DLBS: 0.8125
SBSS: 0.6616

4

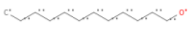
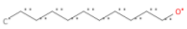
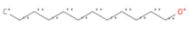
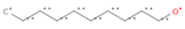
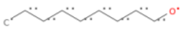
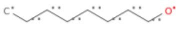


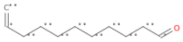
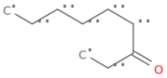
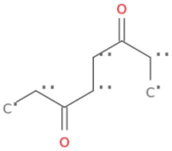
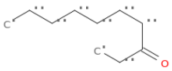
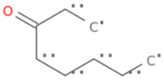
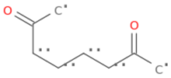
DLBS: 0.79
SBSS: 0.6034

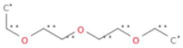
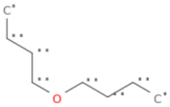
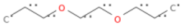
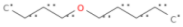
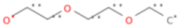
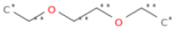
5

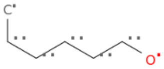
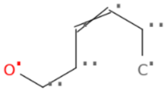
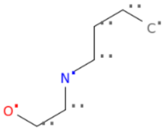
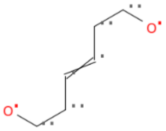
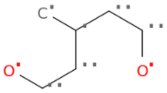
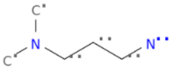


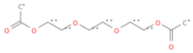
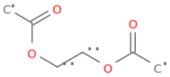
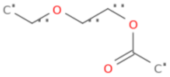
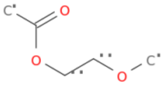
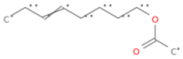
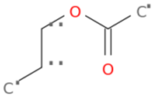
DLBS: 0.7392
SBSS: 0.5187

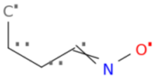
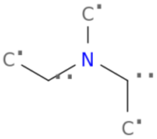
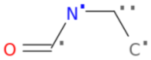
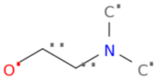
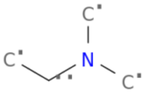
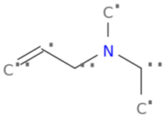
CAS No.: 112-53-8	1	2	3	4	5
					
DLBS: 0.7439 Base_Truth	DLBS: 0.9977 SBSS: 0.9851	DLBS: 0.9931 SBSS: 1.0	DLBS: 0.9875 SBSS: 0.9677	DLBS: 0.9534 SBSS: 0.9474	DLBS: 0.9086 SBSS: 0.9231

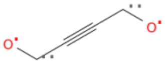
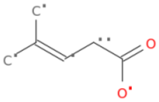
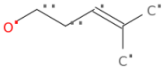
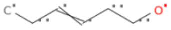
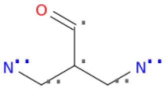
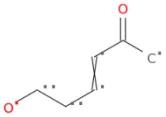
CAS No.: 112-45-8	1	2	3	4	5
					
DLBS: 0.3775 Base_Truth	DLBS: 0.7234 SBSS: 0.1846	DLBS: 0.6716 SBSS: 0.1197	DLBS: 0.6684 SBSS: 0.1825	DLBS: 0.6374 SBSS: 0.1458	DLBS: 0.6272 SBSS: 0.1747

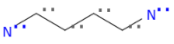
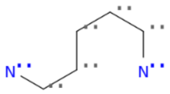
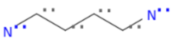
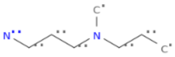
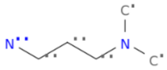
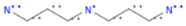
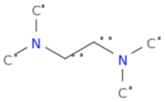
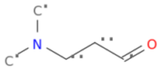
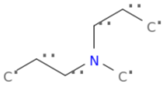
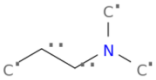
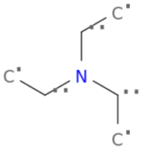
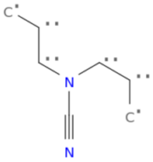
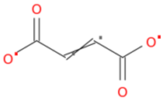
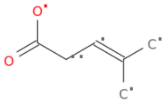
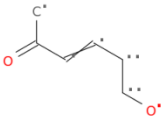
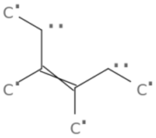
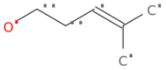
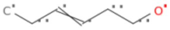
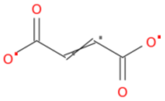
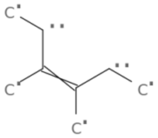
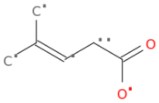
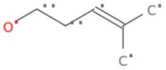
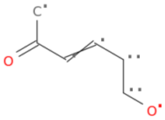
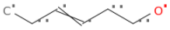
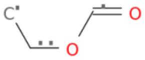
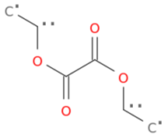
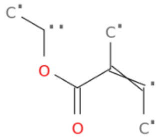
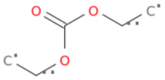
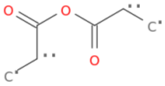
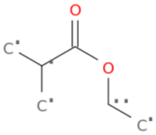
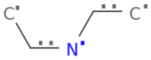
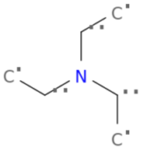
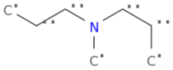
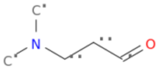
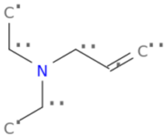
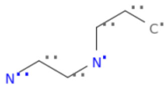
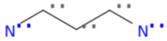
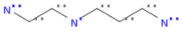
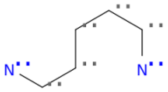
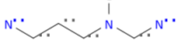
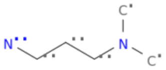
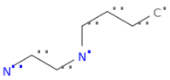
CAS No.: 112-36-7	1	2	3	4	5
					
DLBS: 0.3923 Base_Truth	DLBS: 0.9016 SBSS: 0.4539	DLBS: 0.8502 SBSS: 0.6014	DLBS: 0.7951 SBSS: 0.4028	DLBS: 0.7919 SBSS: 0.7869	DLBS: 0.7502 SBSS: 0.875

CAS No.: 111-27-3	1	2	3	4	5
					
DLBS: 0.511 Base_Truth	DLBS: 0.6159 SBSS: 0.3238	DLBS: 0.5981 SBSS: 0.4515	DLBS: 0.5956 SBSS: 0.2813	DLBS: 0.5892 SBSS: 0.3962	DLBS: 0.5809 SBSS: 0.1523

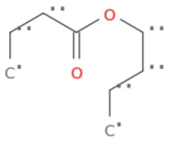
CAS No.: 111-21-7	1	2	3	4	5
					
DLBS: 0.4235 Base_Truth	DLBS: 0.91 SBSS: 0.7991	DLBS: 0.7757 SBSS: 0.7532	DLBS: 0.7057 SBSS: 0.6797	DLBS: 0.6778 SBSS: 0.5089	DLBS: 0.6726 SBSS: 0.6088

CAS No.: 110-69-0	1	2	3	4	5
					
DLBS: 0.4699 Base_Truth	DLBS: 0.7879 SBSS: 0.1651	DLBS: 0.6969 SBSS: 0.2002	DLBS: 0.674 SBSS: 0.1644	DLBS: 0.6636 SBSS: 0.1673	DLBS: 0.5775 SBSS: 0.1589

CAS No.: 110-65-6	1	2	3	4	5
					
DLBS: 0.4056 Base_Truth	DLBS: 0.5966 SBSS: 0.1296	DLBS: 0.5879 SBSS: 0.2087	DLBS: 0.5451 SBSS: 0.2022	DLBS: 0.5387 SBSS: 0.0797	DLBS: 0.5324 SBSS: 0.2173

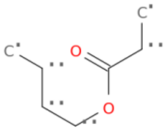
CAS No.: 110-60-1	1	2	3	4	5
					
DLBS: 0.6912 Base_Truth	DLBS: 0.7673 SBSS: 0.8425	DLBS: 0.737 SBSS: 1.0	DLBS: 0.6853 SBSS: 0.4211	DLBS: 0.6565 SBSS: 0.5121	DLBS: 0.6544 SBSS: 0.5593
CAS No.: 110-18-9	1	2	3	4	5
					
DLBS: 0.5147 Base_Truth	DLBS: 0.5067 SBSS: 0.5199	DLBS: 0.4803 SBSS: 0.4031	DLBS: 0.4559 SBSS: 0.6193	DLBS: 0.4343 SBSS: 0.3203	DLBS: 0.4259 SBSS: 0.2916
CAS No.: 110-17-8	1	2	3	4	5
					
DLBS: 0.3893 Base_Truth	DLBS: 0.6856 SBSS: 0.3295	DLBS: 0.6142 SBSS: 0.3236	DLBS: 0.5995 SBSS: 0.1533	DLBS: 0.5833 SBSS: 0.1967	DLBS: 0.5294 SBSS: 0.1338
CAS No.: 110-16-7	1	2	3	4	5
					
DLBS: 0.3268 Base_Truth	DLBS: 0.6296 SBSS: 0.1533	DLBS: 0.6241 SBSS: 0.3295	DLBS: 0.598 SBSS: 0.1967	DLBS: 0.5719 SBSS: 0.3236	DLBS: 0.5618 SBSS: 0.1338
CAS No.: 109-94-4	1	2	3	4	5
					
DLBS: 0.4256 Base_Truth	DLBS: 0.7731 SBSS: 0.3924	DLBS: 0.5698 SBSS: 0.3413	DLBS: 0.5696 SBSS: 0.4102	DLBS: 0.5667 SBSS: 0.3026	DLBS: 0.5611 SBSS: 0.3914
CAS No.: 109-89-7	1	2	3	4	5
					
DLBS: 0.5189 Base_Truth	DLBS: 0.6887 SBSS: 0.3197	DLBS: 0.6864 SBSS: 0.2643	DLBS: 0.6856 SBSS: 0.1537	DLBS: 0.5898 SBSS: 0.2238	DLBS: 0.5826 SBSS: 0.3589
CAS No.: 109-76-2	1	2	3	4	5
					
DLBS: 0.6673 Base_Truth	DLBS: 0.8028 SBSS: 0.5117	DLBS: 0.7889 SBSS: 0.656	DLBS: 0.7887 SBSS: 0.4607	DLBS: 0.7615 SBSS: 0.5278	DLBS: 0.7433 SBSS: 0.3738

CAS No.: 109-21-7



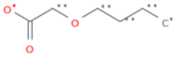
DLBS: 0.6207
Base_Truth

1



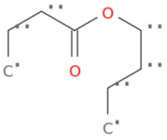
DLBS: 0.7402
SBSS: 0.7684

2



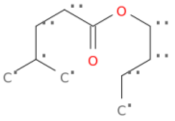
DLBS: 0.7307
SBSS: 0.5285

3



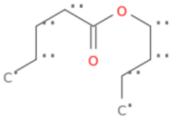
DLBS: 0.6993
SBSS: 1.0

4



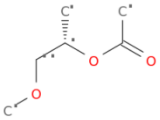
DLBS: 0.6874
SBSS: 0.7753

5



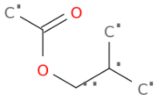
DLBS: 0.6864
SBSS: 0.8569

CAS No.: 108-65-6



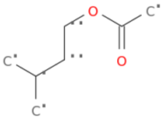
DLBS: 0.3848
Base_Truth

1



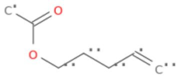
DLBS: 0.7662
SBSS: 0.5166

2



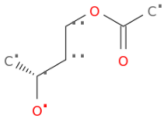
DLBS: 0.7057
SBSS: 0.4487

3



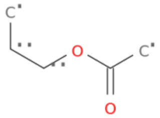
DLBS: 0.612
SBSS: 0.4002

4



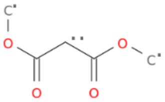
DLBS: 0.6111
SBSS: 0.4348

5



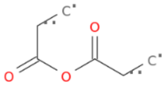
DLBS: 0.6099
SBSS: 0.4208

CAS No.: 108-59-8



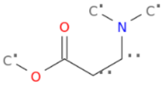
DLBS: 0.4542
Base_Truth

1



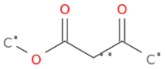
DLBS: 0.8041
SBSS: 0.4747

2



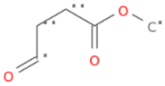
DLBS: 0.7277
SBSS: 0.5385

3



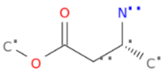
DLBS: 0.6325
SBSS: 0.7411

4



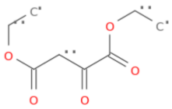
DLBS: 0.6151
SBSS: 0.598

5



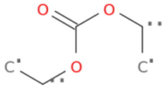
DLBS: 0.6018
SBSS: 0.5544

CAS No.: 108-56-5



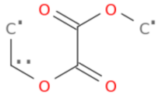
DLBS: 0.3278
Base_Truth

1



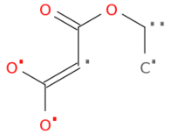
DLBS: 0.597
SBSS: 0.4054

2



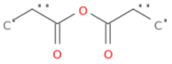
DLBS: 0.5831
SBSS: 0.4943

3



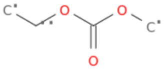
DLBS: 0.5578
SBSS: 0.3695

4



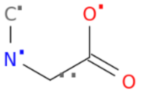
DLBS: 0.5486
SBSS: 0.4005

5



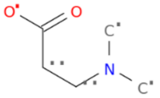
DLBS: 0.5396
SBSS: 0.3709

CAS No.: 107-97-1



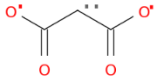
DLBS: 0.6214
Base_Truth

1



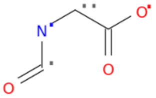
DLBS: 0.8102
SBSS: 0.4434

2



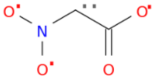
DLBS: 0.7053
SBSS: 0.4365

3



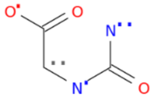
DLBS: 0.6841
SBSS: 0.622

4



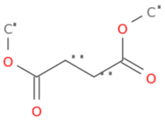
DLBS: 0.6728
SBSS: 0.4636

5



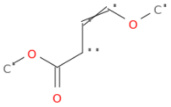
DLBS: 0.6716
SBSS: 0.5783

CAS No.: 106-65-0



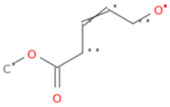
DLBS: 0.5833
Base_Truth

1



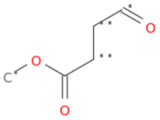
DLBS: 0.9
SBSS: 0.5532

2



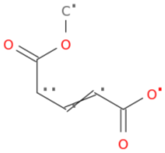
DLBS: 0.884
SBSS: 0.5703

3



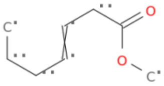
DLBS: 0.8806
SBSS: 0.7453

4



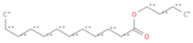
DLBS: 0.8734
SBSS: 0.5197

5



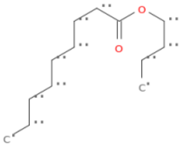
DLBS: 0.8423
SBSS: 0.5488

CAS No.: 106-18-3



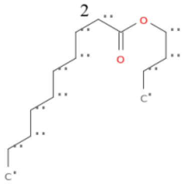
DLBS: 0.674
Base_Truth

1



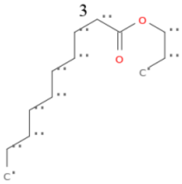
DLBS: 0.9181
SBSS: 0.9655

2



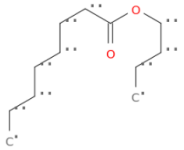
DLBS: 0.9075
SBSS: 0.9783

3



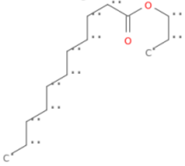
DLBS: 0.9041
SBSS: 0.8666

4

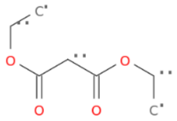
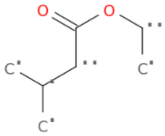
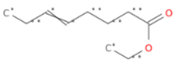
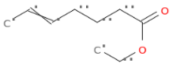
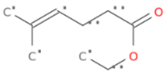
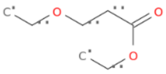
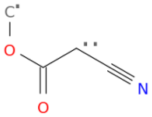
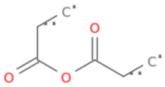
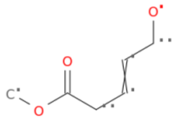
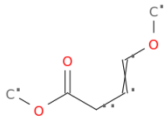
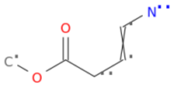
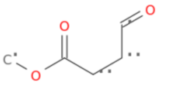
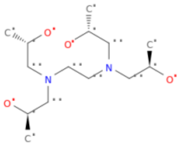
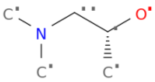
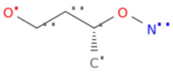
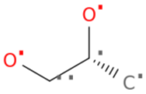
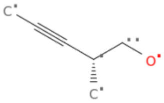
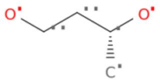
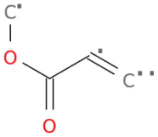
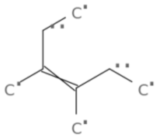
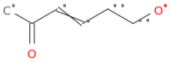
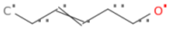
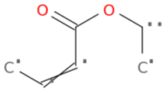
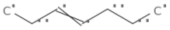
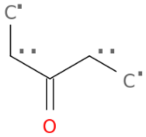
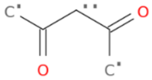
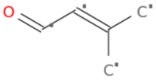
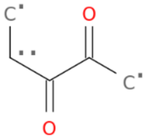
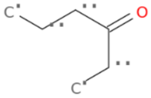
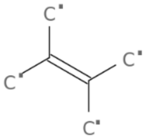
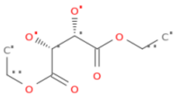
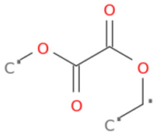
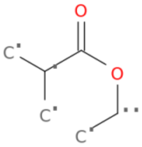
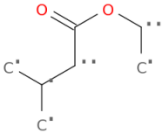
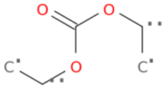
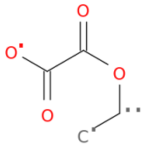
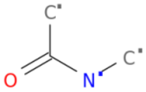
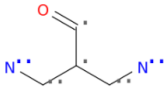
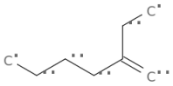
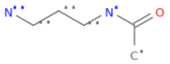
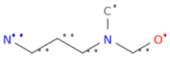
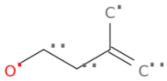


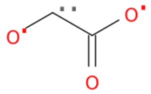
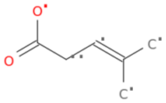
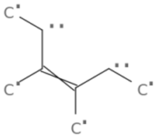
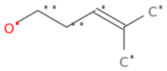
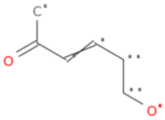
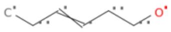
DLBS: 0.8987
SBSS: 0.9427

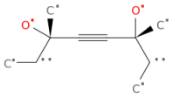
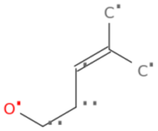
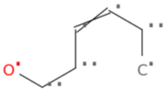
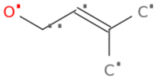
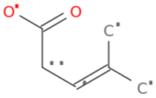
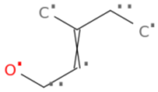
5

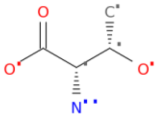
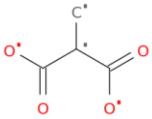
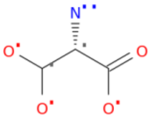
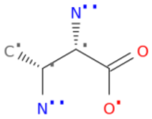
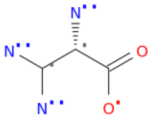
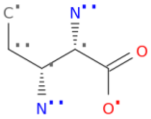


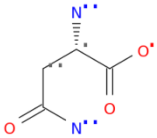
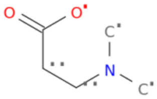
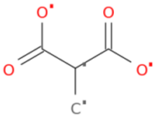
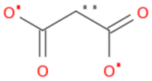
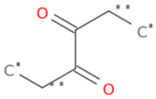
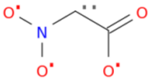
DLBS: 0.8943
SBSS: 0.8768

CAS No.: 105-53-3	1	2	3	4	5
					
DLBS: 0.5274 Base_Truth	DLBS: 0.7346 SBSS: 0.6249	DLBS: 0.6741 SBSS: 0.5066	DLBS: 0.6655 SBSS: 0.5245	DLBS: 0.6636 SBSS: 0.5701	DLBS: 0.6543 SBSS: 0.6412
CAS No.: 105-34-0	1	2	3	4	5
					
DLBS: 0.4947 Base_Truth	DLBS: 0.7386 SBSS: 0.393	DLBS: 0.6302 SBSS: 0.4867	DLBS: 0.6195 SBSS: 0.4708	DLBS: 0.6127 SBSS: 0.4845	DLBS: 0.6056 SBSS: 0.5069
CAS No.: 102-60-3	1	2	3	4	5
					
DLBS: 0.3223 Base_Truth	DLBS: 0.3867 SBSS: 0.548	DLBS: 0.3695 SBSS: 0.2193	DLBS: 0.3475 SBSS: 0.3376	DLBS: 0.3335 SBSS: 0.2043	DLBS: 0.3288 SBSS: 0.3336
CAS No.: 96-33-3	1	2	3	4	5
					
DLBS: 0.5438 Base_Truth	DLBS: 0.682 SBSS: 0.1605	DLBS: 0.6462 SBSS: 0.233	DLBS: 0.6052 SBSS: 0.111	DLBS: 0.5599 SBSS: 0.4093	DLBS: 0.5461 SBSS: 0.1069
CAS No.: 96-22-0	1	2	3	4	5
					
DLBS: 0.5251 Base_Truth	DLBS: 0.9017 SBSS: 0.3922	DLBS: 0.8645 SBSS: 0.2111	DLBS: 0.7612 SBSS: 0.4817	DLBS: 0.7509 SBSS: 0.6959	DLBS: 0.7468 SBSS: 0.1818
CAS No.: 87-91-2	1	2	3	4	5
					
DLBS: 0.2852 Base_Truth	DLBS: 0.6268 SBSS: 0.4641	DLBS: 0.6235 SBSS: 0.5957	DLBS: 0.5857 SBSS: 0.494	DLBS: 0.5593 SBSS: 0.4486	DLBS: 0.5545 SBSS: 0.4883
CAS No.: 79-16-3	1	2	3	4	5
					
DLBS: 0.4272 Base_Truth	DLBS: 0.5065 SBSS: 0.0745	DLBS: 0.474 SBSS: 0.0525	DLBS: 0.4664 SBSS: 0.3615	DLBS: 0.4507 SBSS: 0.0	DLBS: 0.4475 SBSS: 0.1089

CAS No.: 79-14-1	1	2	3	4	5
					
DLBS: 0.4363 Base_Truth	DLBS: 0.6667 SBSS: 0.3957	DLBS: 0.6194 SBSS: 0.1027	DLBS: 0.5909 SBSS: 0.214	DLBS: 0.5873 SBSS: 0.2765	DLBS: 0.5214 SBSS: 0.1679

CAS No.: 78-66-0	1	2	3	4	5
					
DLBS: 0.2315 Base_Truth	DLBS: 0.4939 SBSS: 0.16	DLBS: 0.4883 SBSS: 0.232	DLBS: 0.477 SBSS: 0.1981	DLBS: 0.4706 SBSS: 0.1415	DLBS: 0.4541 SBSS: 0.1879

CAS No.: 72-19-5	1	2	3	4	5
					
DLBS: 0.5728 Base_Truth	DLBS: 0.9423 SBSS: 0.4715	DLBS: 0.896 SBSS: 0.6497	DLBS: 0.8445 SBSS: 0.6563	DLBS: 0.7607 SBSS: 0.5993	DLBS: 0.588 SBSS: 0.5881

CAS No.: 70-47-3	1	2	3	4	5
					
DLBS: 0.2594 Base_Truth	DLBS: 0.6793 SBSS: 0.2673	DLBS: 0.6037 SBSS: 0.3852	DLBS: 0.5964 SBSS: 0.3029	DLBS: 0.5961 SBSS: 0.2068	DLBS: 0.5849 SBSS: 0.345