Multi-label Classification of Organic Compounds Based on Functional Groups

---

General Assembly DSI Capstone Project

By: Patrick L. Cavins, PhD

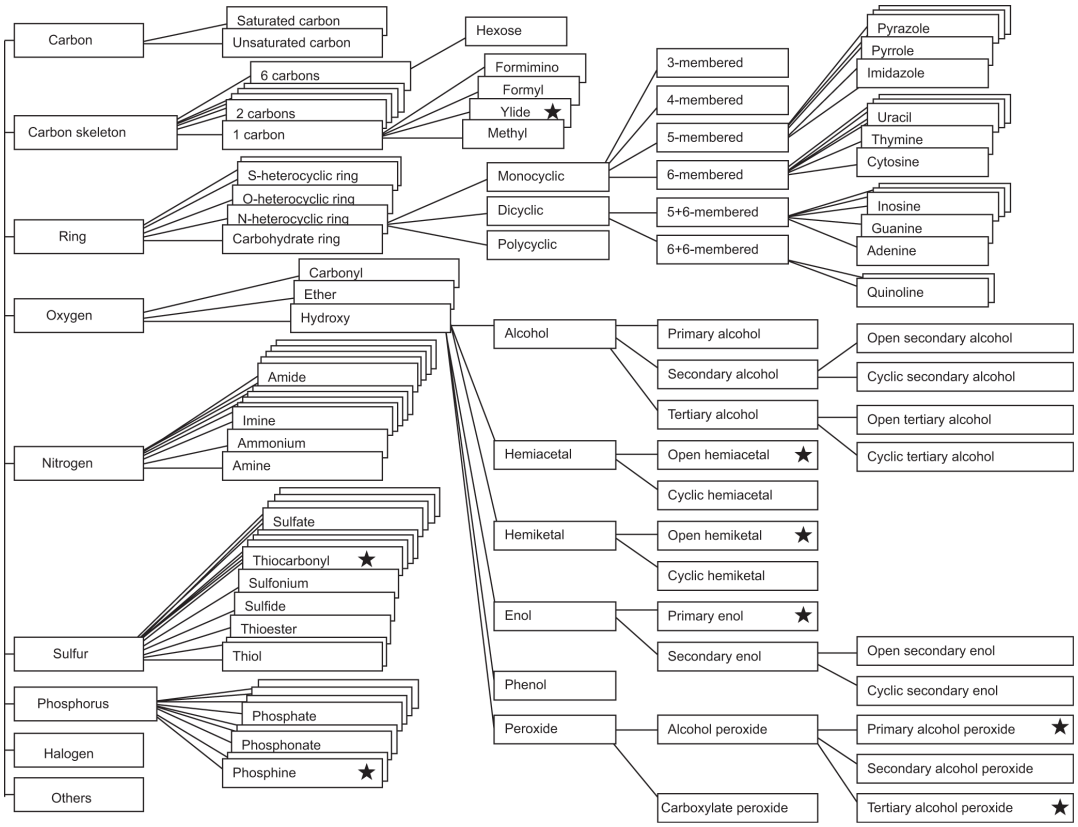


FIGURE X: A BREAK DOWN OF FUNCTIONAL GROUP ORGANIZATION IN ORGANIC CHEMISTRY. HERE DONE BY ATOM. FOUND IN TIPTON *ET AL*[X].

Executive Summary

Structure activity relationships (SAR) are the fundamental building block of organic chemistry and the cornerstone of biology. All biological activity is predicated on the structure of the molecules involved. In organic chemistry the key to those structure activity relationships are functional groups (**Figure X**). Information retrieval is a classic problem in computer science. In the context of chemistry, this topic is referred to as chemoinformatics. Numerous databases are currently available which help chemists understand SAR, but none contain high level functional group information. This is largely due to problems with chemical indexing and search. Such information would be largely beneficial to numerous community within chemistry. As a proof of concept, two convolutional neural networks (CNNs) have been developed which can detect binary differences, such as cyclic structures versus acyclic ones, and a more complex CNN that can detect 15 different functional groups.

Introduction

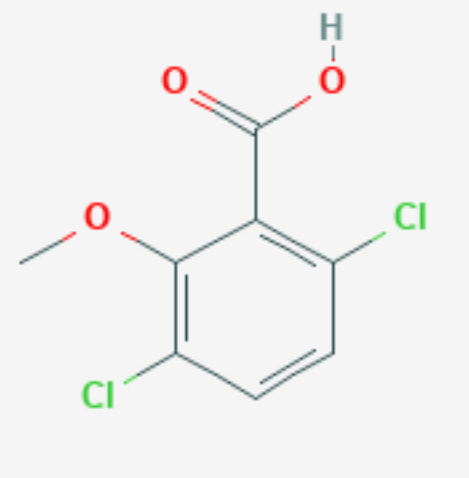


FIGURE 1: ORGANIC LINE STRUCTURE FOR DICAMBA.

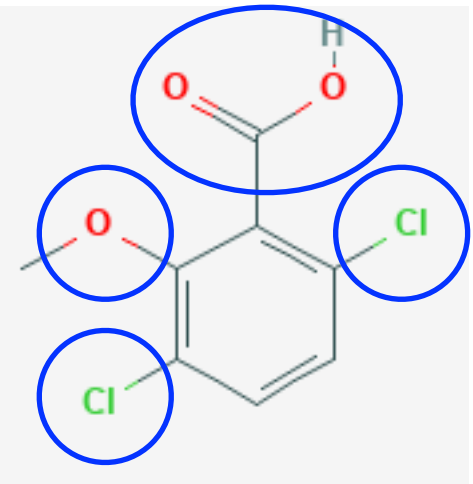


FIGURE 1: LINE STRUCTURE FOR DICAMBA WITH FUNCTIONAL GROUPS HIGHLIGHTED.

The concept of structure activity relationships (SAR) is introduced early in undergraduate organic chemistry classes. Often, though, we talk about it in terms of functional group transformations. If we zoom 10,000 feet, we already know what these words,SAR and functional group transformations, mean in practical terms. If we think about the classic experiment of mixing H2O and oil (n-octane). What is the result? They don't mix, this inability to mix is driven by the physical properties of these two molecules. These physical properties are in turn directly related the different functional groups present on the two molecules. As a quick aside, physical properties in chemistry are defined as properties which do not change the nature of the chemical substance. The fact that the physical properties are determined by the functional groups present in a given molecule means that identifying the the functional group present in programatic way is very valuable to the average chemist. For example, a drug design team in the pharmaceuticals industry team will look at modifying the functional groups present to improve the efficacy of a drug. A recent, in the news example, is the drifting of the herbicide, dicamba[1].

A common line drawing of the organic compound dicamba (**Figure 1**) contains many functional groups (**Figure 2**). A working chemist, would look at this image and immediately be able to infer some physical properties of molecule based on the functional groups present. On much larger scale, training machine learning algorithm to be able to detect these functional group could prove useful when it

comes building out models that on higher a level are used for quantitative structure activity relationship (QSAR) studies. Currently, there are numerous databases which house information relevant to QSAR studies, but none of them explicitly contain information about the functional groups[2].

Literature Survey

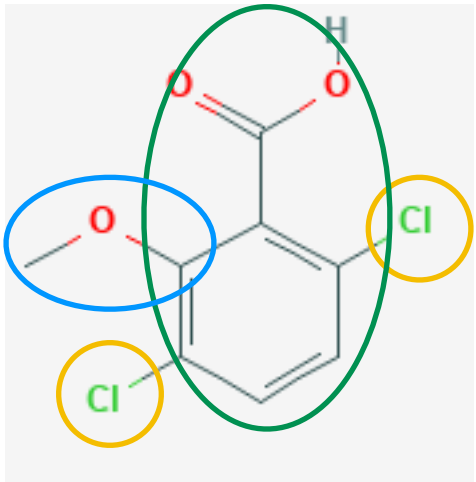
Chemoinformatics

Chemoinformatics is the broad study of storing chemical information as it relates to chemical compounds, D. Mendeleev is often credited as the being the first cheminformatic scientist. Mendeleev created the the modern periodic table by synthesizing many properties chemicals into table so that can it can be easily referenced[3]. Modern studies of chemoinformatics are focused on developing tools that can mine data from journal articles, building models to predict how a given class of compounds might behave under certain experimental conditions, and much work has been devoted to developing databases that can house this information.

From a cheminformatic and computer science perspective, collecting images and categorizing the functional broadly fall into two categories, informational retrieval and data mining. Currently there are numerous databased which are devoted to housing this information. Some of the most popular are KEGG**4**, PubChem**5**, ChemSpider**6**, and ChEMBL**7**. Each of these database house huge amounts of chemical information, but none, that are currently available, contain information specific to functional groups. In 2008, Tipton *et al* reported the creation of powerful database that housing both information on functional groups and substructures**2**. Sadly this resource appears to be no longer actively maintained as is offline at the time of writing this report.

Image to Text Translation

Proper nomenclature is an essential component in organic chemistry. There are numerous methods used to describe organic compounds in text. The three most common methods are CAS Numbers, SMILES and IUPAC Naming. All three popular naming conventions suffer from the same significant issue, information loss. Reexamining Dicamba (**Figure 3**), if we look at the IUPAC name we can see some useful descriptors of the image in the name. For example, *3,6-Dichloro* refers to the spatial relationship of chlorines (yellow circles) around the aromatic ring,



IUPAC Name: 3,6-Dichloro-2-methoxybenzoic acid
SMILES: Clc1ccc(Cl)c(c1OC)C(=O)O
CAS Number: 1918-00-9

FIGURE 3: ORGANIC LINE STRUCTURE FOR DICAMBA. WITH SEVERAL FUNCTIONAL GROUPS HIGHLIGHTED. YELLOW CIRCLES = CHLORINE FUNCTIONAL GROUPS; GREEN CIRCLE = BENZOIC ACID ;

benzoic. Furthermore, the phrase *benzoic acid* (green circle) refers to the aromatic ring and the attached carboxylic acid, and *2-methoxy* (blue circle) references the methyl alcohol group attached to the ring. For many simpler organic compounds like Dicamba one might image being able to use ReGex pattern or named entity recognition to derive the functional groups. This IUPAC naming convention, however, quickly intensifies take for example the blockbuster drug Lipitor (**Figure 4**). The full IUPAC name is as follows *(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid*. No chemist could look at the IUPAC string of text and accurately draw the organic line structure

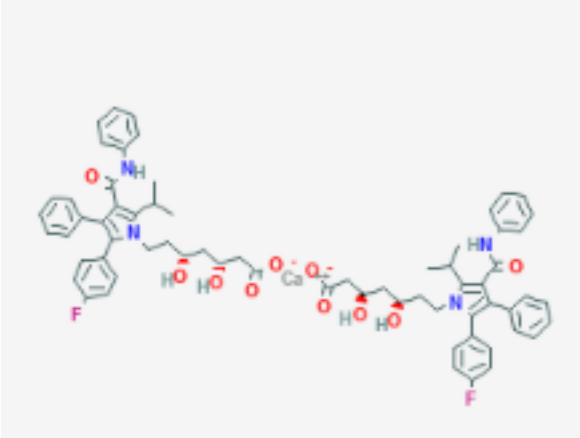


FIGURE 4: ORGANIC LINE DRAWING OF THE COMPOUND LIPITOR

of it. Next, the CAS Number contains no information relative to the structure of the organic compound. Finally, we have SMILES notation. SMILES notation was developed by the EPA in the mid 1980s as way to depict organic structures in text. SMILES is an acronym which stands for simplified molecular input line-entry system. SMILES notation has same pitfall as the IUPAC system. For simple organic compounds interpreting the SMILES notation is fairly straight forward, but as the molecules increase in complexity the usefulness of SMILES notation decreases.

Due to the complexity of the naming conventions/classifications used for organic compounds it unreasonable to pursue the identification of functional groups by using text based entries . As a side note, Tipton *et al* notes that work has been done to develop hash codes which can identify and classify elements of organic compounds. The hash codes are implemented in the CACTVS toolkit, free for academic use[9]. The key is that tool required the structure to be drawn in web-app, and thus the connectivity is explicitly known.

Chemical Indexing

The science of chemical similarity indexing is a complex and growing field in chemoinformatics. At the core of this problem is the science of information retrieval. Chemical indexing falls into three broad categories, similarity, substructure/fragment and superstructure[10]. For example, if we examined the molecule methyl isopropyl ketone (**Figure 5**) as input query. A similarity search owould return results that include deuterated versions, transposition of the methyl group, and ^{13}C contain versions (**Figure 5A)**. A substructure query would return all the molecules that contain the substructure methyl isopropyl ketone inside the larger structure (**Figure 5B**). Finally a superstructure query would return all the molecules which comprise or makeup the structure of methyl isopropyl ketone (**Figure 5C**). All three chemical indexing methods have advantages and disadvantages. Similarity searching will provide the most concise results, but can also return a lot of noise in the form of molecules which are very similar. Imagine versions deuterated methyl isopropyl ketone (**5A,1**) where the deuterium is moved a single atom away. Each of those results are included in a similarity screen, and therefore will decrease the scaffold diversity returned in any similarity search. A substructure query can return a large diversity of molecules which in many cases is desirable. One disadvantage is that this approach will yield results that may not contain the functional group(s) of interest or may contain additional functional not relevant to a given search. In this specific use case, results which contain additional functional groups would be false positives and highly detrimental to the model

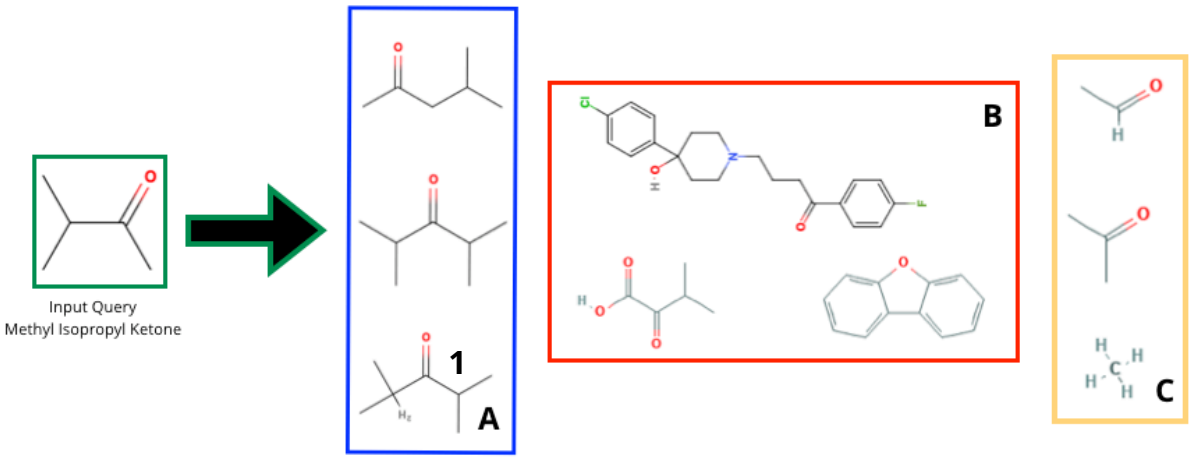


FIGURE 5: PARTIAL RESULTS FROM A PUBCHEM QUERY OF METHYL ISOPROPYL KETONE. (A) RESULTS FROM A TANIMOTO SIMILARITY SEARCH, THRESHOLD SET AT 90%. (B) RESULTS FROM SUBSTRUCTURE SEARCH. (C) RESULTS FROM A SUPERSTRUCTURE SEARCH

trying to be built. The final option is using a superstructure search which are highly dependent on the size of the input structure, and thus for a small molecules like methyl isopropyl ketone will yield a small number of results. However, in this specific use case using a superstructure search can generate an enormous amount of false positives because the specified functional groups will not be specified in results.

Results and Discussion

Experimental Design

As mentioned above, functional groups play an essential role in the fields of chemistry and biology. Collecting the right type of data, images, was essential to properly test the models. Due to the large number of possible functional groups data was collected into 3 broad buckets (**Figure X**). The first bucket used for the binary classification model comparing cyclic vs. acyclic organic molecules. The second bucket uses the four major functional groups found in the plurality of organic compounds including the binary classification. Finally the third bucket contains the four major functional groups plus some of the more common functional groups seen in organic chemistry and biology. Given the search capacity developed at this time, several key functional group where omitted such as ester, imides, and nitriles. To tackle these classification problems several CNNs where built, and their parameters are described below.

Data Collection and Processing

Binary Classification	Multi-Label Classification
1,167	2,313

TABLE 1 : SUMMARY OF IMAGES COLLECTED.

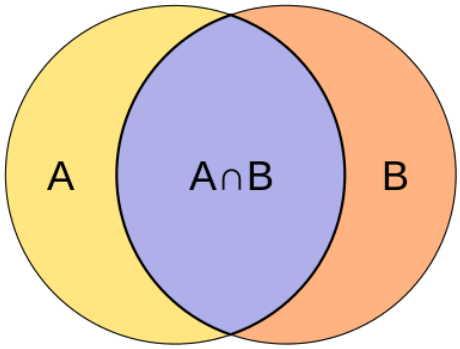


FIGURE X: GRAPHIC REPRESENTATION OF THE TANIMOTO SIMILARITY INDEX

Using the PubChem Power User Gateway (PUG) Rest API a total of 3,480 images where collected [5]. These results are summarized in **Table 1**. Currently a Tanimoto similarity search is being used with the threshold set between 80-95% depending on the class of molecule being queried. A full break down of the results for each classification model can be found in **Table 2** and **Table 3**. As briefly mentioned above, similarity indexing is the most reliable method to collect images that contain the exact functional groups specified by the query compound.

To collect images the PubChem API supported Tanimoto similarity indexing was used. [MORE TO COME]

Results

Binary Classification using CNNs

Using the 1,167 images collected from the PubChem databased a CNN was built largely based on the example network provided in the Keras documentation. Given the large amount of white space in the images only 3 convolutional layers (Conv2D) were used with a relatively small number of nodes (128) considering the size of the input image (300, 300, 1). This model was initially trained over ten epochs resulting in an accuracy score (0.62) which not improvement of the baseline accuracy (0.62).

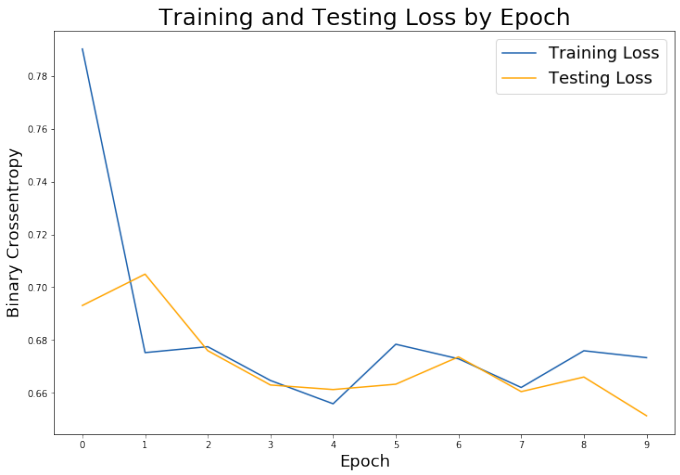


FIGURE X: INITIAL CNN TRAINED OF 10 EPOCHS

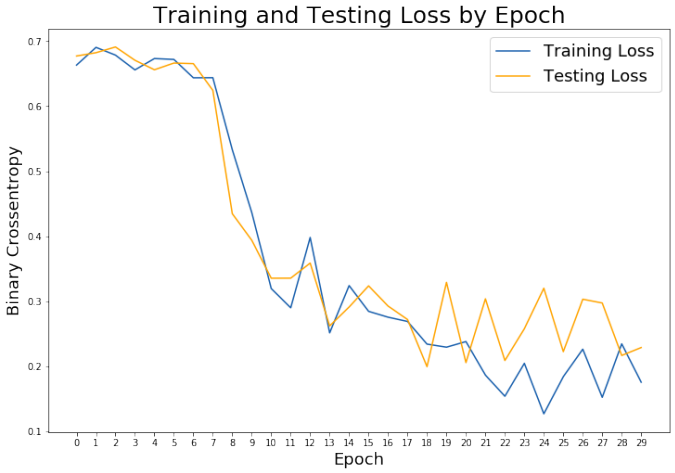


FIGURE X: CNN TRAINED OVER 30 EPOCHS

Instead of modifying the NN, the network was trained for 30 epochs. This NN attained at an accuracy score of 0.92, and given the validation data set the overall classification rate was 94%. Testing the NN on 15 random images collected from PubChem gave a classification rate of 81%. The training and test loss for the binary classification model can be seen in **Figure X** and **Figure X.**

[work in progress]

Functional Group	Count
Alkane (AKA)	1346
Alkene (AKE	249
Alkyl Halide (AKH)	189
Alcohol (ALC)	40
Amine (AMN)	325
Benzene (BNZ)	1231
Ether (COC)	433
Aldehyde (COH)	181
Carboxylic Acid (COO)	518
Amide (COONH2)	186
Ketone (KEY)	271
Sulfur (SHH)	283
Cyclic/Acylic (ring)	1709

TABLE 2: BREAK DOWN OF IMAGES COLLECTED FOR MULTI-LABEL CLASSIFICATION MODEL.

	Count
Cyclic	623
Acylic	371

TABLE 3: BREAK DOWN OF IMAGES COLLECTED FOR THE BINARY CLASSIFICATION MODEL

Directory Outline

These directory contains the jupyter notebooks and data used for this project.

```
|— technical_report.pdf
|— data
|— images_multi_label
|— images_binary
|— jupyter_notebooks
|— models
```

Jupyter Notebooks

These jupyter notebooks are for image collection, data munging, and some image preparation.

```
|— image_collection
|— data_cleaning_binary
|— data_cleaning_multilabel
|— image_prep_binary
|— data_frame_formation
```

Image Collection

SUMMARY: This notebook is a collection of query scripts which can access the Power User Gateway (PUG) rest API from PubChem with the help of the beautiful soup library. A Tanimoto similarity search and corresponding threshold are used in this version of the project. In the data frames created in this notebook, 1 always denotes the presence of the attribute. Also there is not currently distinction made between heterocycles and cyclic compounds. This extends to aromatic rings (i.e. benzene ring functional groups). Based on the model (binary or multi-label classification), images are saved into the appropriate directory.
****add_the_names_of_the_files****

FUTURE WORK: The majority of this notebook runs as script, my goal is to build out that functionality. I need to implement a better system for the handling different functional groups implemented in a given molecules. Furthermore, I hope to add a self-built substructure search based on tanimoto similarity.

Data Cleaning Binary

SUMMARY: Upon inspection of the images collected for the binary CNN some issues were discovered. Mainly some images contain two organic compounds. To programmatically removed them from the database a new PUG rest API query was used to bring in the IUPAC name, and then a ReGex pattern was developed to remove images (i.e. PubChem entries) that contain multiple compounds. The updated CSVs are called `cleaned_{base_image_name}` and `falsepositives_{base_image_name}`.

FUTURE WORK: This ReGex Pattern is not perfect, after many attempts using a the pattern `r';'` was the most effective. This pattern effectively removes any false positives which are images containing multiple structure, but also removes some true positives in the form of ionic compounds. Going forward I would like to improved search methods as described above.

Data Cleaning Multi-label

SUMMARY: This is an **in progress notebook**, used currently to better understand the different type of images being collected from the image_collection notebook for the multi-label classification CNN models.

Image Prep Binary

SUMMARY: This notebook creates a perfunctory train_test_split so that data processed using the .flow_from_directory method inside the Keras module. This also uses the shutil library to copy images into the appropriate directory based on the train test split. The baseline accuracy: Furthermore to use the `.flow_from_directory` method described in the Keras documentation a specific directory break down must be used. For the binary classification problem the directory structure used is described below.

FUTURE WORK: This directory structure should have been implemented during the image collection process. At the time, I was unaware of

```
|-- images
    |-- train
        |-- ring
        |-- not_ring
    |-- test
        |-- ring
        |-- not_ring
```

the `.flow_from_directory` method in Keras. Future versions maybe not require this notebook.

Dataframe Formation

SUMMARY: This notebook makes use of the os library to iterate through the .CSVs generated in either the `image_collection` or the `data_cleaning binary` notebook.

FUTURE WORK: Currently this notebook basically runs as a script. Need to build in some functionality to manage merge/concatenating data frames of different sizes.

Models

```
|— keras_model_binary
|— keras_model_multi_2
|- keras_model_multi_3
|— keras_model_multi_4
|— keras_model_multi_5
```

Keras Model Binary

SUMMARY: This was the first CNN built for the project. Much of the structure of the neural network (NN) was taken from keras documentation on image processing[11]. The images used for this model are very high quality, no noise and the image is center on the graph. To help increase the number of images being used in the model the Keras ImageDataGenerator was used for data augmentation. Only two hyper-parameters were implemented `rotation_range = 30` and `rescale = 1./ 255.`

RESULTS: The NN was initially trained over 10 epoch and results in an overall binary

Binary Classification	
Cylic	0.62
Acylic	0.37

cross entropy loss = 0.65 and train/validation accuracy score = 0.62. The baseline accuracy for this model was 0.62 . Interestingly though the predictions given the initial model were not accurate, 100% of image fed to the model were misclassified.

By examining the probability output, numerous instances where very close to the sigmoid threshold of 0.5. Instead of the changing the NN, the same NN was just trained over longer epochs. This decreased the binary cross entropy loss to 0.24 and accuracy increased to 0.92. The classification rate was 94%. Using the `image` class inside the Keras preprocessing module 15 random images were tested given a classfication rate equal to 0.81. The results from these modules are saved as pickled files inside the data subdirectory.

References:

1. https://thefern.org/2019/04/the-herbicide-dicamba-is-sparking-a-civil-war-in-farm-country/
2. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0001537
3. https://www.acs.org/content/acs/en/careers/college-to-career/chemistry-careers/cheminformatics.html
4. KEGG
5. PubChem
6. ChemSpider
7. ChEMBL
8. ACS, CAS
9. CACTVS
10. https://chem.libretexts.org/Courses/University_of_Arkansas_Little_Rock/ChemInformatics_(2017)%3A_Chem_4399%2F%2F5399/6%3A_How_to_Search_PubChem_for_Chemical_Information_(Part_2)