**Exploring Molecular Generation Using Variation Auto-Encoders**

Daniel Flockhart

Contents

[Project Overview 2](#_Toc142576768)

[Dataset 3](#_Toc142576769)

[Datasets Used In Initial Stages 3](#_Toc142576770)

[Preprocessing 4](#_Toc142576771)

[ChemBERTa 4](#_Toc142576772)

[Visualising Large Datasets 6](#_Toc142576773)

[Target/Ground Truth generation – Images of Chemical Skeletons 8](#_Toc142576774)

[Training - Variational Auto Encoder 12](#_Toc142576775)

[Inputs 12](#_Toc142576776)

[Encoder 12](#_Toc142576777)

[Latent Dimension 12](#_Toc142576778)

[Decoder 12](#_Toc142576779)

[Output 12](#_Toc142576780)

[Hyperparameters/Other information 12](#_Toc142576781)

[Generation 14](#_Toc142576782)

[User Input 14](#_Toc142576783)

[Conditions 14](#_Toc142576784)

[Experimentation 15](#_Toc142576785)

[Dataset 15](#_Toc142576786)

[Preprocessing 15](#_Toc142576787)

[Training 15](#_Toc142576788)

[Post Processing 15](#_Toc142576789)

[Visualisations/Analysis 15](#_Toc142576790)

[Future Work 16](#_Toc142576791)

[Input Synthesis 16](#_Toc142576792)

[Simulation 16](#_Toc142576793)

[Ranking 16](#_Toc142576794)

[Analysis Profiles 16](#_Toc142576795)

[Viable Synthesis Routes Prediction 16](#_Toc142576796)

[Reinforcement Learning 16](#_Toc142576797)

[Input Synthesis 18](#_Toc142576798)

[Target-Treatment Dataset 18](#_Toc142576799)

[Simulation of Generated Molecules 19](#_Toc142576800)

[QSAR – Predicting the effects of a molecule 19](#_Toc142576801)

[Docking – Predicting the interaction of molecules 19](#_Toc142576802)

[Ranking System for Generated Molecules 20](#_Toc142576803)

[Ranking Features 20](#_Toc142576804)

[Retrosynthesis 21](#_Toc142576805)

[Synthesis 21](#_Toc142576806)

[Review of Project 22](#_Toc142576807)

[Limitations of Program 22](#_Toc142576808)

[Project Summary 23](#_Toc142576809)

[Bibliography 24](#_Toc142576810)

[Academic Papers 24](#_Toc142576811)

[Videos 24](#_Toc142576812)

# Project Overview

This project is being completed as a part of my Summer Research Internship with Funding from the Birmingham Digital Chemistry Network and is supervised by Dr Jianbo Jiao and Dr Linjiang Chen.

This project is in the field of self-supervised learning for digital chemistry. Specifically, this project looks into developing algorithms to learn chemistry representations in an unsupervised manner. This goal of this project is to showcase a new way to learn chemistry knowledge without expert human guidance.

# Datasets

## Initial Testing Dataset

The first dataset I was provided with was a dataset of roughly 48,000 organic semiconductor molecules and their corresponding canonical SMILES representation selected from the Cambridge Structural Database. For the purpose of testing and development of the project, this dataset is suitable.

When the time comes to generate a fully trained model that allows the user to modify a molecules properties, we will require a larger dataset with more information about the molecular properties.

## Secondary Testing Dataset

Due to some limitations with the previous, initially provided dataset that I will get onto in the preprocessing section I required a smaller and highly related dataset for the purpose of testing the application of t-SNE for visualisation of the dataset. To solve this I found a dataset of ~550 molecules with a diverse range of related structures.

## Ideal Dataset

In

# Preprocessing

## ChemBERTa

Machine learning models typically receive discrete or continuous quantitative data as inputs. The dataset previously mentioned is formatted as the SMILES representation followed by the other information about its molecular properties. Unfortunately, we cannot directly use the SMILES representation as an input to our model as SMILES are a string of numbers and letters that represent its structural information. I had needed to find vector representations of these SMILES without losing critical information about their structures.

To solve this problem I decided to use a ChemBERTa natural language processing model to do this. Specifically, I decided to use a model trained on 100k SMILES strings from a benchmark dataset, ZINC. For the earlier stages of testing, this model will have suitable accuracy. In later stages of testing I will experiment with using different fine-tuned RoBERTa based models to assess performance and the effect they have on model performance.

To visualise the performance of the model for converting SMILES to Vectors I have used t-SNE. T-SNE is a statistical method for visualising high-dimensional data by reducing each datapoint to a location in two or three dimensions. The result of passing a SMILES representation through the ChemBERTa model is a Vector of size 768. Therefore, it is critical to reduce these dimensions for visualising. I first tested a smaller dataset of ~550 related molecules. To enable us to see patterns in the data clearer and before using t-SNE to visualise them, I used the k-means algorithm with the elbow method of cluster analysis to approximate the best number of clusters and provide every molecule with an appropriate cluster label. After doing this I then passed these vectors into our t-SNE algorithm and displayed the data using the previously generated labels to enable us to better understand the produced graph.

A colorful dots on a white background

Description automatically generated3 Clusters

A colorful dots on a white background

Description automatically generated with medium confidence

4 Clusters

A screen shot of a computer screen

Description automatically generated

A graph with a line

Description automatically generatedA graph with a blue line

Description automatically generated44 Clusters – Elbow Method was used for this to calculate the optimal number of clusters

The above graphs shows the smaller dataset clustered multiple times with varying cluster numbers and visualised using t-SNE and two example graphs of the convergence of the elbow method.

When you hover over individual data points, you are provided with the labels and other information about the molecule.

A yellow background with black lines

Description automatically generated

A yellow background with black lines

Description automatically generated

Respective Molecules Skeleton for Arachidonoyl Serotonin and AM-404

A chemical structure of a molecule

Description automatically generatedA structure of a chemical formula

Description automatically generated

Aswell as clear patterns in the locations and distributions of clusters, and noted by the molecules shown above, cluster items have similar structural properties. I am confident that the ChemBERTa model provides more than adequate conversions of SMILES to Vectors, without losing the most important structural information.

## Visualising Large Datasets

As described earlier, one of the test datasets I am using consists of ~48000 Molecules and their respective smiles. It is useful for us to be able to visualise this data in order to help us understand any pre-existing patterns in the data. I have had a few issues with visualising this dataset

A colorful dots on a white background

Description automatically generated

## Limitations

With larger datasets that require more clusters, without having the ability to hover over each data point and check their specific cluster numbers, it may be hard to see specific clusters because of similar shades of colours, I am working on a fix for this.

I have also experimented with the use of PCA for the visualisation of the data, but I have not seen any additional advantage over t-SNE.

A black and white arrow pointing to a black rectangle

Description automatically generated

The above image displays the simplified process of this component of the project.

Below display and example of the initial dataset provided.

A screenshot of a computer

Description automatically generated

After processing all the data from the original dataset above, an inputs csv is created for use with the Variational Auto-Encoder during training.

A screenshot of a computer

Description automatically generated

## Target/Ground Truth generation – Images of Chemical Skeletons

In order to train the Self-Supervised Machine learning model from an initially unstructured dataset for the generation of molecules, we must make the program automatically generate data labels which are to be further used in the training process as ground truth. To do this I have decided to use the Python RDKit Library for the generation of images of chemical skeletons from their SMILES representation.

Firstly, to generate the molecules we pass the SMILES representation into the RDKit program which will then generate unscaled and unnormalized images of the skeletons. This means that the images are of different sizes and have different scales for bonds/lengths. We have to then make the program automatically standardise the size of these bonds between all the molecules in the dataset and then ensure that a large percentage of the molecules still fit in the resulting images. The following image shows some of the unprocessed molecules generated by rdkit.

A screenshot of a computer screen

Description automatically generated

As seen above all the images have different scales and sizes. To standardize this, the program gets the range of sizes of the images generated by RDKit, it then finds a suitable scaling bound to fit the images. Standardizes the sizes of the bonds and centers the molecules. The program ensures that >98% of the molecules fit into the frame. Unfortunately, there is a small fraction of molecules in the dataset that are far too large to fit into the images without scaling the rest of the images substantially down and losing data to compression.

For the <2% of molecules that do not fit into the frames, I have a working heuristic that covers a large percentage of the remaining large molecules. For the images where the width or height of the molecule is larger than the rescaled frame, we rotate it through 45 Degrees. After this there is a tiny fraction (Approximately 0.2%\*) of the dataset that still do not fit, even after this process. In later stages of development I will work on a solution for these molecules that compresses their representation, however, for the purpose of this project, it is not necessary to include these molecules in the dataset.

Here is are some examples of an image that does not fit into the frames. Note – if I increase the scaling bounds too substantially, smaller molecules lose data and images would have to be larger, therefore increase the computation required.

A black and red lines on a white background

Description automatically generatedA diagram of a molecule

Description automatically generated

In later experimentation I will adjust the bounds for the preprocessing of these images when I have access to the actual dataset of molecules and smiles.

Additionally, there are some issues with specific molecules that RDKit Struggles to generate because of their form in the dataset.

A black and white drawing of a crystal

Description automatically generatedA drawing of a circle

Description automatically generated

At a later stage in development I will develop an automatic detection system for finding these images and removing them, however, as they make up such a small portion of the current dataset, I can just manually remove them from the initial dataset.

**A screenshot of a computer screen

Description automatically generated**

Displayed above is the results of the preprocessing for the molecules.

All the images consist of 400x400 pixel images with standardized and proportional bond lengths to limit the model having to learn its own scaling approximation which would not provide us with any additional value, it would only serve to increases the processing time of the model.

Note : This program is designed to be entirely automated and only requires the user to provide a dataset of SMILES

A close-up of a sign

Description automatically generated

The above image displays the simplified process of this component of the project.

Summary

* Generates vector representations of SMILES for input to training Model
* Generates processed ground truth labels/images for model

# Training - Variational Auto Encoder

For this project, we have decided to use Variational Auto Encoders for the generation of novel molecules.

## Inputs

The Inputs to the encoder network consists of the Vector representations of molecules smiles, generated by the ChemBERTa model, as described previously. The conditions that will eventually modify the molecular properties are used as inputs to the decoder after being concatenated with the latent dimension.

## Encoder

The encoder currently consists of dense layers followed by Batch Normalisation, with each dense layer decreasing in size to compress the vector input. The batch normalisation improves training stability and accelerates convergence. In later stages of experimentation, I will test different depths, types and lengths of hidden layers.

## Latent Dimension

The encoder produces a compressed representation of the input, this is called the latent dimension. This is also the stage where we input the conditions that effect the generated molecule. During the training process the conditions are just the current states of the molecule. When I have access to the full dataset I will be able to give a full description of the conditions used and their chemical significance. The conditions are concatenated with the latent dimension and passed into the decoder.

## Decoder

The decoder consists of 1 Dense layer followed by multiple Convolutional Transposing Layers upscaling the image to generate an output vector.

## Output

This vector can then be converted into an image and the loss can be calculated, allowing the model weights to be updated.

Below is a summary of the model.

A diagram of a diagram

Description automatically generated

## Hyperparameters/Other information

These values are subject to change for experimentation purposes.

Loss Functions : Adam Optimiser (Adaptive Moment Estimation) with Mean Squared error and KL Loss

Input Size : 768

Output Size : 400x400 pixels (160000)

Batch Size : 128

Condition Size : 10

Latent Dimension : 128

Learning Rate : 0.0001

After training the outputs are displayed to the user.

The next stage of the development process is to create a description profile of these molecules and their difference to the starting molecules. Then, following this, we need to simulate the molecules and rank them on their performance for there given targets. For this we may use QSAR – Quantitative Structure-Activity Relationships.

# Generation

## User Input

The goal of this project is to make a fully automated pipeline for generating new molecules with specific characteristics. To do this, all we should require from the user is a target characteristic and the program should generate molecules for that characteristic.

For our previously mentioned model, the input is a Vector representation of a starting molecule that is trained in a self-supervised manor, from the images generated by RDKit. This means that we need to user to provide the model with a possible starting molecule for the model and a condition with which they wish to enforce on the molecule. This however, is not ideal. Ideally, I would want the user to input a target molecule/virus etc, the model would identify multiple possible starting molecules and then generate a condition to improve upon the starting molecules and provide the user with a large amount of new novel molecules.

After this process, when the new molecules have been generated, we can then assess if they are valid, create a description profile and predict their effects. After this we can then simulate and rank the results, producing the user with a detailed analysis of potential target molecules.

This entire process is designed to be entirely automated, other than the initial user input.

## Conditions

Masks

A blue rectangular box with white text

Description automatically generated

A diagram of a molecule

Description automatically generated

# Experimentation

This section describes some of the further experiments I intent to do.

## Dataset

Use different datasets varying in:

* Size of dataset
* Content Type
* Conditions

## Preprocessing

* Experiment with different size images
* Coloured Images
* Different Scaling Bounds
* Automatic Molecule Cleaning

## Training

* Try with GANs, Diffusion Models or other model types
* Change encoder depth and architecture
* Change latent dimension size
* Change conditions
* Change decoder depth and architecture

## Post Processing

## Visualisations/Analysis

* Use different Dimensionality Reduction algorithms
* Different Perplexity
* Use different clustering algorithms
* Different Amount of Clusters

# Future Work

Despite this Internship Ending soon, I intend to continue work on this project in my own time, here is some of the future components I wish to integrate into the project.

## Input Synthesis

Having to find a potential starting molecule and specific chosen condition manually is not ideal for speed and efficiency, therefore I propose an input synthesis tool where the use is able to enter a specific target illness/disease/molecule etc, a model then selects a variation of potential starting molecules to begin from. This model will also generate a selection of conditions to improve upon these molecules based on their shortfalls such as low bioavailability, lack of potency, short effect times etc.

## Simulation

After the model has generated an amount of viable molecules that have been verified as valid molecules, and post processed. We will need to find a way of simulating the molecules to discover their effects, any potential toxicity/side effects, their efficiency and other characteristics that are useful for chemists.

## Ranking

After simulating the molecules, we need a system for ranking molecules by their performance in the simulation, comparing their side effects, potency, how simple the molecules are etc.

## Analysis Profiles

After generating, post-processing, simulating and ranking the molecules, we need to automatically generate profiles for the created molecules, showing information about their performance, effects and other important information.

## Viable Synthesis Routes Prediction

After discovering potential target molecules, we need to generate possible synthesis routes for the molecules to speed up the Drug Discovery process. This will most likely require a large dataset of synthesis routes which will most likely be difficult to source.

## Reinforcement Learning

I would like to experiment with, and add functionality for reinforcement learning for the generation of more effective molecules in combination with the input synthesis component.

A screenshot of a computer screen

Description automatically generated

# Input Synthesis

In future development of this project, I intend to work on an input synthesis component that increases the usability and efficiency of the drug discovery process. This component will allow the use to input either a target receptor, disease or virus and it will select a variety of possible starting molecules for the model. Additionally, it will optimise a set of conditions to modify the starting molecules to improve upon limitations of the starting molecules.

This component will require a large dataset of treatments for specific illnesses, molecules and their applications in medicine.

## Target-Treatment Dataset

# Simulation of Generated Molecules

In the drug discovery process virtual simulation and analysis of the possible effects of a drug is critical to conduct before following onto physical trails of a molecule. In later stages of development, I intend to make a simulator for testing and analyzing the potential effects of a molecule. There are a few

## QSAR – Predicting the effects of a molecule

## Docking – Predicting the interaction of molecules

# Ranking System for Generated Molecules

It is important that, after the generation of a selection of molecules, we rank the performance of each molecule to allow us to save time and sort any non-viable molecules.

## Ranking Features

There are a large number of features that we are able to rank a molecules performance by. As mentioned above in the simulation section, we can use methods like QSAR to predict the activity of a molecule. Some additional features we can use to rank molecules include:

* Binding Affinity
* Toxicity
* Structural Diversity – We may want to test a range of structures to discover new potential leads.
* Bioactivity
* ADME Properties: Absorption, Distribution, Metabolism and Excretion
* Synthetic Feasibility
* Solubility

# Retrosynthesis

Generating synthesis routes from a generated molecule using retrosynthesis.

## Synthesis

The goal of

# Review of Project

In this section, I review the project and any potential limitations or shortfalls.

## Limitations of Program

# Project Summary

**Dataset**

**Preprocessing**

**Model Generation**

**Input Synthesis**

**Molecule Generation**

**Molecule Analysis**

**Results Summary**

**Conclusions**

# Bibliography

## Academic Papers

<https://jcheminf.biomedcentral.com/articles/10.1186/s13321-018-0286-7>

<https://arxiv.org/pdf/2209.01712.pdf>

<https://www.nature.com/articles/s41597-022-01142-7>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5570547/>

<https://www.kaggle.com/datasets/jithinanievarghese/drugs-related-to-common-treatments>

<https://www.nature.com/articles/s42256-022-00557-6>

## Videos

<https://www.youtube.com/watch?v=4mygq7Brtu8>