**Molecular Generation Using Variational Auto-Encoders**

Daniel Flockhart

Artificial Intelligence and Computer Science BSc

Contents

[Abstract 2](#_Toc143964662)

[Project Overview - Introduction 3](#_Toc143964663)

[Related Work 4](#_Toc143964664)

[Datasets 5](#_Toc143964665)

[Zinc 250k 5](#_Toc143964666)

[Conditions 5](#_Toc143964667)

[Pre-Processing 6](#_Toc143964668)

[Vectors 6](#_Toc143964669)

[Conditions 6](#_Toc143964670)

[Target Images 6](#_Toc143964671)

[Methods 7](#_Toc143964672)

[ChemBERTa 7](#_Toc143964673)

[Variational Auto Encoder 7](#_Toc143964674)

[QSAR 7](#_Toc143964675)

[Training 8](#_Toc143964676)

[Computation/Hardware 8](#_Toc143964677)

[Hyper Parameters 8](#_Toc143964678)

[Analysis 8](#_Toc143964679)

[Genetic Information 8](#_Toc143964680)

[Generation 9](#_Toc143964681)

[Input Synthesis 10](#_Toc143964682)

[Inputs 10](#_Toc143964683)

[Outputs 10](#_Toc143964684)

[Analysis 11](#_Toc143964685)

[Simulation 12](#_Toc143964686)

[Ranking 13](#_Toc143964687)

[Personalised Molecule Generation 14](#_Toc143964688)

[Genetic Tailoring 14](#_Toc143964689)

[Synthesis Routes 15](#_Toc143964690)

[Experimentation 16](#_Toc143964691)

[Results 17](#_Toc143964692)

[Discussion 18](#_Toc143964693)

[Future Work 19](#_Toc143964694)

[Conclusion 20](#_Toc143964695)

[Acknowledgements 21](#_Toc143964696)

[References 22](#_Toc143964697)

[Academic Papers 22](#_Toc143964698)

[Videos 22](#_Toc143964699)

[Appendices 23](#_Toc143964700)

# Abstract

# Project Overview – Introduction

This project is a continuation of work I completed over Summer of 2023 as part of the a Summer Research Internship in association with the University of Birmingham and with Funding from the Birmingham Digital Chemistry Network. The internship was supervised by Dr. Jianbo Jiao and Dr. Linjiang.

This project is in the field of self-supervised learning for digital chemistry. Specifically, this project investigates the development of algorithms to learn chemistry representations in an unsupervised manner without expert human guidance.

The goal of this project is to create an automatic pipeline for the generation of new molecules based on a starting molecule and a set of conditions to modify the structure or chemical properties of the starting molecule. To do this, we will use Variational Auto-Encoders to take an input molecule in the form of a vectorised version of its SMILE, generate a latent vector, concatenate it with a condition and generate a molecule based on these provided conditions.

Below illustrates the general pipeline plan for generation.

A screenshot of a computer screen

Description automatically generated

# Related Work

# Datasets

## Initial Dataset

For my internship 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 testing and development of the project, this dataset is suitable.

## Psychoactive Substances Dataset

For the purpose of generating test models for the experimentation with different architectures for the training models, I decided I needed a smaller dataset of molecules that had similar structural properties and yet had enough diversity to be able to modify conditions. For this reason, I found a small dataset of 1200< items of known psychoactive substances grouped into their corresponding pharmacological classes.

Not only was this dataset better for experimenting with model architectures, it also allowed me to test and experiment with visualisation techniques like t-SNE and PCA for representing molecules vector representations on a 2D plane. As I will discuss later, I used a Natural Language Model based on BERT models for converting molecule smiles into vector representations that could then be used by machine learning models.

## Zinc 250k

Additionally, I required a larger dataset to test how well VAE model converge on generating detailed structures. For this reason, I found a substantially larger dataset of over 250,000 items to create a dataset of molecules and their corresponding conditions relating to their structure and properties.

During the processing of these molecules, when gathering the data for the molecules conditions, over half of the data could not be found due to a lack of information about the chemical properties of such molecules. In most of these cases, these molecules appeared on databases like ChemBL and PubChem but did not have the corresponding information required for the dataset. As a result my dataset consisted of the remaining ~110,000 molecules. In future iterations, I would ideally prefer an even larger dataset with a greater number of conditions due to the complexity of the task at hand.

## Conditions

The following list describes each of the available conditions in the ZINC dataset.

* Atom Stereo Count

Atom stereochemistry indicates the spatial arrangement of atoms around a stereocenter (chiral center) in a molecule. It is important in drug design and biochemistry because the 3D orientation of atoms can impact the molecule's biological activity.

* Bond Stereo Count

Bond stereochemistry refers to the arrangement of atoms or groups around a double bond or a ring junction. It's crucial for understanding molecular conformation and reactivity, especially in organic chemistry.

* Charge

The charge of a molecule reflects its overall electrical state, whether it is positively charged (cation), negatively charged (anion), or neutral. Charge influences a molecule's chemical reactivity and interactions with other molecules.

* Complexity

Molecular complexity quantifies how structurally intricate a molecule is. It can be used to assess the diversity of chemical compounds and their potential for drug discovery, as more complex molecules often have unique properties.

* Exact Mass

Exact mass is the precise mass of a molecule, accounting for the masses of individual atoms and their isotopes. It is crucial for identifying unknown compounds and verifying molecular formulas.

* H Bond Acceptor Count

This count represents the number of sites in a molecule capable of forming hydrogen bonds. Hydrogen bonding is critical in various chemical and biological processes, including drug-receptor interactions.

* H Bond Donator Count

This count represents the number of hydrogen atoms in a molecule that can act as hydrogen bond donors. It's important for predicting how a molecule can participate in hydrogen bonding interactions.

* Heavy Atom Count

The heavy atom count is the number of non-hydrogen atoms in a molecule. It's a simple descriptor of a molecule's size and can help estimate its molecular weight.

* Molecular Weight

Molecular weight is the sum of the atomic weights of all atoms in a molecule. It's a fundamental property used in chemistry and pharmacology to compare and characterize compounds.

* Rotatable Bond Count

This count represents the number of single bonds in a molecule that can rotate freely. It's important for assessing the molecule's flexibility, which can affect its biological activity and binding to target receptors.

* TPSA

TPSA quantifies the polar surface area of a molecule. It's a critical parameter for predicting the molecule's solubility and permeability, especially in drug design.

* XlogP

XlogP measures a molecule's lipophilicity, which indicates its tendency to partition between water and a lipid phase. This parameter is essential for predicting a molecule's bioavailability and distribution in the body.

Ideally, I would like to gather more data about the molecules such as clinical test data, this will likely be something I work on later into the project

# Pre-Processing

## Vectors and Conditions

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 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 tokenising 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 the later stages of testing, I will experiment with using different fine-tuned BERT-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 data point 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 – The elbow Method was used for this to calculate the optimal number of clusters

The above graphs show 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 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

As well as clear patterns in the locations and distributions of clusters, as 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 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

Above is an image of the dataset clustered with 10 clusters.

Currently, if I increase the number of clusters beyond this number, it becomes increasingly difficult to visualise the patterns in the data as there are often cluster labels with similar shades of the same colour.

At the same time, when I increase the number of clusters and analyse the molecules within them, they are structurally related. As an example, below I have listed the results of one specific cluster when the entire dataset is clustered with an arbitrary 2000 clusters. The SMILES are all related.

A screen shot of a computer code

Description automatically generated

A group of black and white molecules

Description automatically generated

Below is an image of the dataset clustered with 4 clusters where one cluster has been specified and visualised. There is a clear pattern to the data, I will also experiment with different models for tokenising in the future. It may also make the visualisation clearer if I have a more detailed dataset with information regarding chemical properties as the data for t-SNE was merely the SMILES encoded in vector form.

A pink dots in a circle

Description automatically generated

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. It is also known that PCA does not handle non-linear relationships between data as well as t-SNE.

Note: If you have any further questions about the visualisations and clustering process let me know; some nuances are not particularly clear.

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 displays an 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

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 must 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, standardises 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 does 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, it is not necessary to include these molecules in the dataset.

Here 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 increasing 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 are 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 scaling approximation which would not provide us with any additional value, it would only serve to increase 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.

The below image shows the full overview of the preprocessing stage of this project.

A diagram of a diagram

Description automatically generated

# Methods

## Variational Auto Encoder

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

## Inputs

The Inputs to the encoder network consist 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

A diagram of different types of layers

Description automatically generatedThe 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 affect the generated molecule. Currently, during the training process, the conditions are just the arbitrary current states of the molecule and hence I will not be able to experiment with modifying the conditions of the data. 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.

## Conditions

I trained some test models so far on arbitrary constants for the conditions as I do not have access to the full dataset. I will expand this section in later stages.

## Decoder

The decoder consists of 1 Dense layer followed by multiple Convolutional Transposing Layers upscaling the image to generate an output vector. Below is an image describing the architecture of the full variational auto-encoder. I will experiment with using different architectures and layers in the future.

A group of colorful rectangular shapes

Description automatically generated

## 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

## QSAR

The next stage of the development process after training a full model is to create a description profile of these molecules and their difference from the starting molecules. Then, following this, we need to simulate the molecules and rank them on their performance for their given targets. For this, we may use QSAR – Quantitative Structure-Activity Relationships. This is most likely something I will continue in my own time.

# Training

## Computation/Hardware

## 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 (Vector Representation of SMILES): 768

Output Size: 400x400x1 (400x400 pixels with 1 colour channel – Greyscale)

Batch Size: 128

Condition Size: 12 (Arbitrary)

Latent Dimension: 128 (I am experimenting with different values)

Learning Rate: 0.001

After training, the outputs are displayed to the user.

As mentioned before, when I have access to a full dataset of possible conditions and molecules, I will be able to train a full model.

## Analysis

# Genetic Information

Ranking with genetics

A colorful lines with text

Description automatically generated with medium confidence

Compress Genetic Information

A diagram of a different color

Description automatically generated with medium confidence

# Generation

Currently, I have trained a handful of smaller models on small subsets of the larger dataset. As I have not had access to a full dataset of conditions, I have not been able to experiment with modifying the conditions and states of molecules. I have been able to generate some clear molecules for ensuring the VAE architecture was capable of generating clear images. Here are some examples of images generated by my model.

A molecule structure on a white background

Description automatically generatedA black and white image of a molecule

Description automatically generatedA chemical structure on a white background

Description automatically generated

For testing purposes, I decided to only generate greyscale images. Generating coloured images will take 3x longer due to the extra 2 colour channels required. Switching to generate coloured images will be simple.

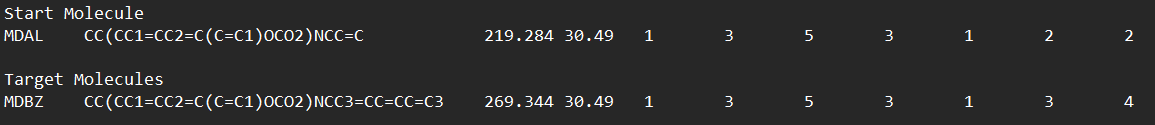
The images above were generated by a model trained on a test subset of 100 images for 10 minutes and were used to test whether VAE had sufficient layers in its decoder to generate high-resolution models.

When I have access to different datasets, I will be able to modify the latent dimensions to a greater extent and modify the conditional statements, to hopefully have a resultant effect on the molecules produced.

As another example, here is some generated images I tested with a tiny dataset.

A close-up of a molecule

Description automatically generated



## 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 manner, from the images generated by RDKit. This means that we need the 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 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. 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.

# Input Synthesis

## Inputs

The inputs to this section is the illness and any other required information

## Outputs

The goal of this component is to generate a chosen starting molecule for the VAE and generate a set of conditions.

# Analysis

The next experimental step in this process is to convert the images produced by the variational auto-encoder back into a smiles representation and caption the molecule – essentially creating a profile for the molecule so the user can then use/test/rank the molecule which I will discuss soon.

# Simulation

# Ranking

Heuristic

* Performance estimation based on simulation score + genetic ranking

# Personalised Molecule Generation

## Genetic Tailoring

I propose a new technique for personalized healthcare.

It consists of two separate (variational) Auto-Encoders and one decoder

Input Synthesis

* Generates a starting molecule and a condition

Encoders

* One compresses a target users genetic information into a latent space – We will not use the decoder of this
* One compresses a starting molecules representation into a latent space

Decoder

* Takes the Compressed genetic Information
* Take the starting molecule latent space
* Takes the user specified condition

# Synthesis Routes

# Experimentation

This section describes some of the further experiments I intend 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
* Experiment with inputting conditions into both the encoder and decoder

## Visualisations/Analysis

ChemBERTa Visualisations

* Use different Dimensionality Reduction algorithms
* Different Perplexity
* Use different clustering algorithms
* Different Amounts of Clusters
* Build Cluster Exploration tool

# Results

# Discussion

# Future Work

I intend to continue working on this project in my own time; here are 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 user can 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. I have further documentation of my ideas for this part, I have not included them here.

## Simulation

After the model has generated several viable molecules, we must verify the molecules are valid, and post-process them. 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. I intend to research and apply methods like QSAR and Docking.

## 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 may be difficult to source.

## Other Ideas

* Applying reinforcement learning to the generation, and testing of molecules

# Conclusion

**Preprocessing**

The preprocessing component of this project is complete to a standard I am content with, it is fully automated and other than providing it with a dataset, does not require any other interaction. I have identified a couple of areas for further improvement, and I would like to further experiment with this component in the future.

**Model and Molecule Generation**

I have used and experimented with the variational auto-encoder, it works as I had intended it to, now all I have to do is find a suitable dataset with a range of conditions and then I can train a full large model for the intended application of the mode.

**Conclusions**

There is much more I would like to continue working on for this project, as mentioned previously: input synthesis, simulation, ranking and retro-synthesis are some parts I am interested in designing. I am content with the progress I have made over the last 6 weeks and I have learned a great deal about the application of machine learning to Chemistry. I look forward to staying involved in this area of research and intend to conduct my own research projects related to this field in the future.

# Acknowledgements

# References

## Academic Papers

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

Journal of Informatics - Molecule Generative Model based on conditional variational autoencoder for de novo molecular design.

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

Arxiv - ChemBERTa-2: Towards Chemical Foundation Models

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

Nature - Organic Materials Repurposing, a dataset for theoretical predictions of new applications for existing compounds.

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

National Library of Medicine - Disease-Drug Database for Pharmacogenomic-Based Prescribing

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

Kaggle – Drugs Related to Medical Conditions

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

Nature - Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework

## Videos

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

Microsoft YouTube Channel - Research Talk: AI for drug discovery

# Appendices