Deep generative model for 2D structure of natural product

Michael Anastos (manastos) [1] **Nadim Farhat (nfarhat)** [2] **Liu Cao (liuc1)** [1]

Introduction

Natural products has become the center of pharmaceutical leads. Global Natural Products Social (GNPS) molecular networking infrastructure gathered unprecedented amount tandem mass spectra of natural products from thousands of laboratories over the world. However, most of the mass spectra do not have any annotations of the corresponding molecule structure. The direct reconstruction of molecule structure from mass spectra is very difficult due to the noisy nature of mass spectrometry data. Instead, searching mass spectra against the known molecule structure database is a common way of annotation. However, the search over the known molecule database will limit any structurally novel natural products discovery. One way to propose the putative structure of natural products is through microbiome genome mining, but it is currently limited to Ribosomally synthesized and Post-translationally modified Peptides (RiPPs), which is a only very small class of natural products. Therefore, generating putative molecule structure is of great importance for the discovery novel natural products and drug leads.

2D molecule structure can be represented as undirected graph, with nodes as atoms (e.g. carbon, hydrogen, oxygen, nitrogen, etc.) and edges as chemical bonds (single or double bonds). Thus, proposing putative natural products structure is essentially equivalent to generating graphs. Deep generative models have been successfully applied to the generation of images, music and prose. Recently, it has also been used to generate realistic graphs. Generating molecule graph is very challenging due to the discrete nature of the graph and structure constrains of realistic chemical compounds.

In this project , we first do a literature survey on the models of molecular generation/optimization and to find a suitable model to generate putative molecule structures. We will investigate any tailored model by training and validating it on the known organic compounds database like QM9.

[1] Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2] University of Pittsburgh, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

Related Work

Graph Generation methods: Finding probabilistic models for generating random graphs that describe real world networks is an intensively studied problem with a long history. For this goal a large number of models have been developed. We could split those models in two categories based on the input that they require, the data-tailored models and the non-data-tailored models.

The non-data tailored models are models that take as an input only a few parameters. For example the well known Erdős-Rényi model $G(n,p)$ (see (2)) assumes that each edge appears independently with probability p. Other models, like the preferential attachment model (1) try to capture concepts like 'the rich becomes richer', while the geometric graph model (4) assumes that the nodes are generated on some surface and two nodes are connected if they are close enough. The above models are easy to study due their simplicity. However they usually make strong independence assumptions and in most of the cases are unable of capturing complex dependencies that we may want to enforce. As a result they are unsuitable to model highly structured graphs.

In contrast, data tailored model take as input a set of graphs $\mathcal{G}$ and try to generate graph that 'looks like', hence having similar properties with, elements of $\mathcal{G}$. Many of the methods used are inspired by Image and video generation, an area where the task of finding generative models has great success. However trying to transfer such methods to discrete and highly structured data like graphs has a number of challenges. Some of the main challenges are (in contrast to image generation): (i) the output may vary in size, that is in the number of nodes and/or of edges (ii) if we represent a graph by a matrix then any permutation of the notes corresponds to a different matrix, hence a different representation of the same graph, and therefore a given graph on n nodes may has up to $n!$ representations, (iii) we want to capture complex underlying dependencies.

To overcome the above challenges recent models (see (5), (7)) like the DGMgraph and the graphRNN use a recurrent structure like RNN or GRU. The basic idea is to start from the empty graph and try to built it up by adding nodes/edges. The recurrent structure tries to compute the conditional probabilities of extending the current graph in some way, hence answering questions like should I add an edge/node. By adding additional constrains on the order that the nodes are generated (i.e. they correspond to a BFS ordering) we can even reduce the number of such questions.

Regularization and optimization of adversarial training over graphs: The discrete structure of graphs for adversarial optimizations poses challenges for the backpropagation optimization algorithm. Backpropagation assumes that samples are pulled from a continuous distribution. Therefore several groups developed methods to overcome this challenge. The Gumbel-Softmax distribution suggested by (8) substitutes non-differentiable samples from a discrete distribution with continuous samples. The gumble-softmax continuous relaxation was used in the recurrent neural networks for text style generation (9) Shen et all used the gumble-softmax with the Professor-Forcing algorithm to help in the cross alignment of non-parallel text (10) .

Molecular Generation: The advances in the generation of images, music, translation, text transfer over graph have inspired researchers to use graph generation in the field of molecular studies.

In terms of model categories, there are mainly two types of generative models for small molecule graphs: variational autoencoder (VAE) and generative adversarial networks (GAN). VAEs are likelihood models that allow for easier and more stable optimization than implicit generative models like GANs. In terms of graph generation strategies, people use sequential generation, adjacency matrix generation or SMILES generation. For sequential generation, graphs generation is formalized as a sequence of decisions of generating nodes, nodes types, edges and edges types one by the other. For adjacency matrix generation, there is a upper bound of number of possible nodes in the graph, and node types and edges weights are usually generated independently, which usually leads to invalid graphs. SMILES format is a linear string representation of a chemical structure. Below, we review different frameworks of small molecule graph generation strategies.

ChemVAE (17) is a a variational autoencoder that uses convolutional neural network to encode a SMILES string. The latent representation is then feed into a decoder to produce SMILES output and a chemical property prediction neural network, which enables the learning of latent representations that are expected to have high values of desired properties.

GrammarVAE (16) is a variational autoencoder that instead of encoding a SMILES string, it first forms parse trees according to SMILES grammar and string, then extract rules and maps these rules to the latent space.

GraphVAE (18) encodes a graph by graph convolution network and decode the latent layer into a probabilistic fully-connected graph containing a adjacency matrix, edge attributes probability tensor and a node attributes probabilities tensor.

JT-VAE (19) or junction tree VAE, instead of directly encode a graph, first decompose a graph into a junction tree and then develops a encoder and a decoder on the junction tree representation of the graph. The recursive decoding process guided by topological prediction and masking techniques guarantees that generated molecule is valid.

We now focus on 3 distinct models used for the generation of molecular graphs. Our proposed method draws ideas from all 3 of them in an attempt to combine them.

MolGAN: In (3) De Cao and Kirp combined a GAN model with reinforcement learning techniques in order to built a DNN that aims to generate new small molecular graphs with desired properties. Their model consists of 3 components. A generator and a discriminator, found in GANs, as well as a reward network . The reward network R_r takes as input a graph and outputs a number.

The generator generates new data under some prior. Then the discriminator tries to distinguish generated from real data. These two parts compose a GAN. The additional reward network tries to encourage the generator to adapt a policy that takes actions with high rewards.

In reinforcement learning, a policy $\pi_\theta(\cdot)$, parameterized by θ, takes as an input an environmental state s and outputs a distribution $p_\theta(a|s)$ over all the possible actions. Then an action a is chosen according to $p_\theta(a|s)$ and a step-reward $w(a)$ is assigned to the agent. The objective is to maximize the expected total reward gained by the agent over θ. In (3) they consider the generator to be the policy, action to be the outputted graph G and reward to be $R_r(G)$. Thus theoretically they can train their generator (policy) using any reinforcement learning technique. They choose to use an off-policy actor-critic algorithm. The reward network R_r is trained separately with both real and synthetic data.

For objective they use a linear combination of the GAN and the RL objectives.

SCAT: Graph generative scattering network (20) has two major components a encoder and a decoder, but we only need to train the decoder. With fixed parameters, the encoder first use scattering transformation to provides multiscale signal representation. The signals are then normalized to be Gaussian latent variables. The decoder are two networks with fully connected layers. One is edge weights tensor, and the other is vertex feature tensor. The two tensors together will leads to a graph representing a possible molecule structure.

CGVAE: Constrained Graph Variational Autoencoders (CGVAE) (13) is a model for graph generation that combines the following methods: gated graph neural networks (GGNNs) (14) and a variational autoencoder (VAE) (15). The GGNN is a gated sequential neural network that is used in the encoder-decoder of the VAE . The model incorporate domain-specific constraints, these constrains are useful for molecules generation. The benefits of this hybrid model, it allows the optimization of numerical properties of molecules, is able to generate molecules matching the statistics of the training distribution and it generates semantically meaningful molecules. The downsides of using the sequential generation of GGNN , is the loss of permutation symmetry. The loss of the symmetry is replaced with random graph linearizations, however random graph linearization needs to be constrained. Therefore they forced the generative model component to be conditioned only on the current state of generation.

Proposed Method

Noticeably there is a clear comparison between models with a VAE structure and models with a GAN structure used to generate molecular graphs in a non sequential way. The GAN models have higher fidelity meaning that they succeed in generating graphs that respect the properties that a molecular graph should satisfy with higher frequency (i.e. every vertex that corresponds to an atom of carbon should be incident to 4 edges). However they suffer from the low variation in the graphs that they generate. In contract VAE models have lower fidelity, but higher variability in the output. In this project we propose to combine VAE and GAN substructures in a unified framework that may enjoys the advantages of both VAE and GAN based models.

VAE-GAN

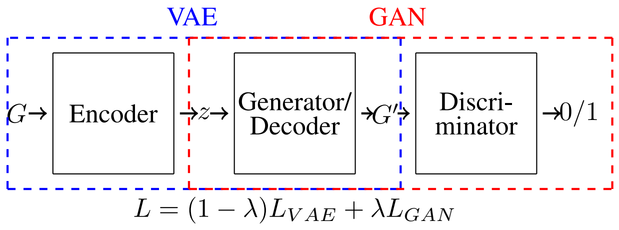

Figure 1. VAE-GAN

Our first model has a VAE-GAN architecture. It is composed by 3 components: the encoder E_ρ, the decoder/generator G_θ and the discriminator D_ϕ. The Encoder takes as an input a graph $G = (N, A, F)$. $N \in \{0,1\}^{n\times T}$ defines the atom's type of the vertices. $A \in \{0,1\}^{n\times n\times y}$ is the incident matrix. The edges correspond to bonds and the third coordinate of A corresponds to the bond type. Finally the i^{th} column of $F \in \{0,1\}^{n\times f}$ defines the feature vector of the i^{th} atom. G is mapped by the encoder to a d-dimensional $N(0, I)$ distribution. The generator/decoder takes an an input a d-dimensional vector and outputs a graph $G = (N, A, F)$. Finally the discriminator takes samples from both the dataset and the generator and learns to distinguish them.

We train VAE-GAN by minimizing, with respect to ρ, θ and ϕ, the following objective:

$$L_{VAE-GAN} = (1-\lambda)L_{VAE} + \lambda L_{GAN} \qquad (1)$$

where $0 \le \lambda \le 1$ is a parameter. In the the case $\lambda = 0$ or $\lambda = 1$ we drop the Encoder and the Discriminator respectively from our network. L_{VAE} is given by

$$L_{VAE} = E_{q_\rho(z|G)}[\log p_\theta(G|z)] - KL[q_\rho(z|G)||p(z)].$$

For L_{GAN} we used the improved WGAN loss, introduced in Gulrajani et al. (2017) and given by

$$L_{GAN} = -D_\phi(x) + D_\phi(G_\theta(z)) + \alpha(||\bigtriangledown_{x'} D_\phi(x')|| - 1)^2.$$

While training, if $\lambda > 0$, then for z we use the output of the encoder. Otherwise we sample z from $N(0, I)$. The first two terms of the above expression correspond to the original $WGAN$ loss, while the third parameter is a gradient penalty. x is sample from the data and $x' = \beta x + (1-\beta)G_\theta(z)$ with β being drawn from a Uniform[0,1] distribution. α is a hyperparameter. We set $\alpha = 10$.

SCAT with Validity loss

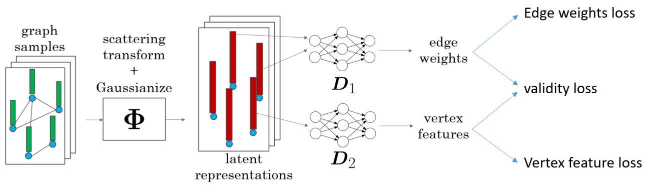

Figure 2. SCAT with validity loss

One of the main problem of SCAT is that many of the molecule structure it generates are, although novel and unique, not valid. With the encoding method of diffusion graph scattering transformation and Gaussian whitening, the molecule structure generated by SCAT has 98.1% uniqueness and 94.2% uniqueness. However, only 38% of the structures are valid chemical structure.

The original loss function of SCAT is the distance between reconstructed graph and the original graph:

$$L(D_1, D_2) = T^{-1}\sum_{i=1}^{T}[||W^{(t)} - D_1(\Phi(X^{(t)}))|| + ||X^{(t)} - D_2(\Phi(X^{(t)}))||]$$

where D_1 and D_2 are the decoding networks of edges weights and vertex features, $\{(X^{(t)}, W^{(t)})\}_{t=1}^{T}$ are corresponding vertex features and edges weights of training data points. Obviously, edges and vertexes are generated independently without any constraints. However, in realistic molecule structure, the the node degree is determined by the node label. For example, carbon can have up to 4 single bonds, while oxygen can have only 2. To guide the learning of a valid graph, we propose to add a validity loss to the original loss function (Figure 2) as follows:

$$\begin{aligned} L'(D_1, D_2) &= T^{-1}\sum_{i=1}^{T}[||W^{(t)} - D_1(\Phi(G^{(t)}))|| \\ &+ ||X^{(t)} - D_2(\Phi(G^{(t)}))||] \\ &+ \lambda * relu(softmax(D_1(\Phi(G^{(t)})))[0,1,2,3] \\ &- softmax(D_2(\Phi(G^{(t)})))[4,2,3,1,0]) \end{aligned}$$

where $softmax(D_1(\Phi(X^{(t)})))[0,1,2,3]$ is the expected number of bonds of the edge weights matrix, and $softmax(D_2(\Phi(X^{(t)})))[4,2,3,1,0]$ is the expected number of maximum bonds according to vertex feature matrix. The additional loss encourages the total number of bonds of each node to be less than the maximum bonds of its node type. λ is the hyper-parameter that adjusts the validity loss contribution. The greater λ is, the higher validity loss will contribute to the whole loss function.

SCAT decoder with different hidden layers

The authors of the SCAT (20) noted that they found different results when they tested out 1 hidden layers and 3 hidden layers for the decoder ; they noticed that a single hidden layer generates more valid molecules however the uniqueness of these molecules are very low. On the other hand, a SCAT decoder with 3 hidden layers can generate unique molecules while the validity of the molecules drop. We decided to investigate further the relation between the number of layers in the decoder and the validity, novelty and uniqueness. The authors correlated the difference in the results to the complexity of the decoder; a more complex decoder can generate new molecules however a simpler decoder is not able to create novel molecules, yet it is able to generate valid molecules. The authors did not provide proof for their claim therefore we decided to investigate SCAT with 1,2,3 and 4 fully connected hidden layers.

Experiments

1. Dataset

We mainly focus on the organic compounds structure database QM9 (22). QM9 contains 133,885 stable organic molecules made up of atoms C (carbon), H (hydrogen), O(oxygen), N (nitrogen), F (florine) with up to 9 heavy atoms. The structure information of each compound in QM9 database is in sdf format. We use the python package 'chainer_chemistry' to download and extract the molecule structures, and convert them into structure graphs.

2. Implementation

2.1. VAE-GAN

The code of the VAE- GAN model is adapted from the code in MolGAN github repository. It is mainly based on Tensorflow 1.7.0 and chain_chemistry and RDKit.

For the generator and the discriminator we closely follow the architecture given in (3). The encoder has almost identical architecture to the discriminator. The first part of both is a Relational-GCN with two layers and [64,32] hidden units. At the $(\ell+1)^{th}$ layer, the feature representation at node i, $h_i^{\ell+1}$ is calculated via

$$(h_i^{\ell+1})' = f_s^{\ell}(h_i^{\ell}, x_i) + \sum_{j=1}^{|N|}\sum_{y=1}^{|F|}\frac{A_{ijy}}{N_i}f_y^{\ell}(h_j^{\ell}, x_i),$$

$$h_i^{\ell+1} = \tanh((h_i^{\ell+1})').$$

f_s^{ℓ} and f_y^{ℓ} are linear transformation that are shared across the nodes of layer ℓ. Then in both encoder and discriminator the second layer of the Relational-GCN is aggregated into a 128 dimensional vector via

$$h'_G = \sum_{v\in V}\sigma(r(h_v^2, x_v)) \odot \tanh(t(h_v^2, x_v)),$$

$$h_G = \tanh h'_G.$$

Here V represents the set of nodes in the second layer and x_v its features. $\sigma(x) = 1/(1+\exp(-x))$, r and t are MLPs with a linear layer output. Their outputs, are passed ro σ and $\tanh$ respectively and then combined by an element-wise multiplication that is denoted in the above expression by $\odot$.

In the generator the last 128 dimensional layer is connected to a 2-layer MLP of dimensions 128 and 1 while in the encoder it is connected to a 2-layer MLP both of dimensions 128.

The decoder is a 3-layer MLP of 128, 256 and 512 hidden units respectively. The last layer is linearly projected to match N, A and F.

2.2. ”SCAT with validity loss”

The code of the model ”SCAT with validity loss” is adapted from the code in SCAT github repository. The encoder is the same with SCAT method with 15 long embedding feature vector for each atom. The decoder for both edge weights and vertex features are fully connected three layer neural networks with 128, 256 and 512 nodes in each hidden layer. The validity loss contribution hyper-parameter λ is set as 1, 2, 4, 8, 16 and 32. Total number of epoches is 300. The program is ran on Precision Tower 5810 with GPU NVIDIA GeForce GTX 1080 (8G RAM). Each epoch takes about 6 seconds. The code is mainly based on Tensorflow 1.13.1 and chain_chemistry and RDKit.

2.3. ”SCAT with different hidden layers”

We cloned the github SCAT directory. The code had 3 hidden layers by default, we have rewritten three additional version of the codes to run for 1(128 units), 2(128, 256 units) and 4 layers(128,256 and 512 units). The epoch number was set to 300 and it was ran on a Nvidia Geforce GTX 980 with estimated 40 seconds per epochs for the 3 Hidden layers(with 128,256 and 512 fully connected units).

Benchmarking			
Model	Valid	Unique	Novelty
CGVAE	100	98.57	94.35
GraphVAE	55.7	76.0	61.6
VAE-GAN (λ=1)	25.6	95.7	91.9
MolGan	99.2	37.1	64.5
SCAT-SN-Valid loss	82.2	85	83.9
SCAT-SN	61.35	91.29	86.23
SCAT-DW-Valid loss	51.6	96.5	91.9
SCAT-DW	38.0	98.1	94.2
SCAT-SN-1 layer	67	87.93	86.65
SCAT-SN-2 layers	96.94	12.33	95.67
SCAT-SN-3 layers	65.37	91.24	85.75
SCAT-SN-4 layers	47.12	93.44	90.79

Table 1. Performance comparison of different molecular structure generative model. Bold rows are the models proposed in this paper. SCAT-SN represents SCAT with spectral graph scattering transformation and Gaussian spherization. SCAT-DW represents SCAT with diffusion graph scattering transformation and Gaussian whitening.

3. Results

We will compare the models based on the following statistics: Validity, Uniqueness and Novelty. A molecule (molecular graph) is said to be valid if each atom is incident to the right number of bonds. The latest depends on the type of the molecule. Validity measures the ratio of valid and all generated molecules. Uniqueness measures the ratio of unique and valid molecules. Finally Novelty measures the proportion of valid molecules that are not in the train dataset.

3.1. Result of VAE-GAN

We train the VAE-GAN model with respect to (1) for $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. We graph the uniqueness, validity and novelty scores of the corresponding models in Figure 3. For $\lambda = 0$ the $VAE - GAN$ is equivalent to a VAE model while for $\lambda = 1$ it is equivalent to a GAN model. As we vary λ from 0 to 1 the uniqueness score drops from almost 100 per cent to 0. In opposition the validity increases from a low 25 to a high 99.

3.2. Result of SCAT with validity loss

Under SCAT setting of spectral graph scattering transformation and Gaussian whitening (SCAT-SN in Table 1), with the increase of different validity loss contribution parameter λ, the validity keeps increasing, while the novelty and uniqueness keep decreasing. What is interesting about the the trend is that when the validity loss contribution parameter is small, the validity rate increases very fast, while the uniqueness and novelty only decreases very slowly (Figure 4), which makes it possible to greatly enhance the validity rate by sacrificing only a little portion of unique and novel structures. Overall, the number of valid novel and unique structure will be hugely increased.

As is shown in Table 1, SCAT-SN with validity loss is the most balanced model in terms of the three metrics. It even outperforms some of the previous methods like GraphVAE in all the three metrics. Under SCAT setting of diffusion graph scattering transformation and Gaussian whitening (SCAT-DW), although validity loss help increase the validity rate, but the uniqueness and novelty rate drops also very quickly.

3.3. Results of SCAT with decoder of different hidden layers

As per the results in Table 1, we ran the SCAT decoder with fully 1,2,3 and 4 connected hidden layers. except for the decoder with one hidden layer, the number of layers is found to be correlated with the increase in uniqueness e.g number of layers 2,3,4 have uniqueness of 12, 91 and 93 respectively. The novelty is not associated with the number of layers and the validity is anti-correlated with the number of layers;layers 2,3,4 have 97, 65 and 47 validity respectively.

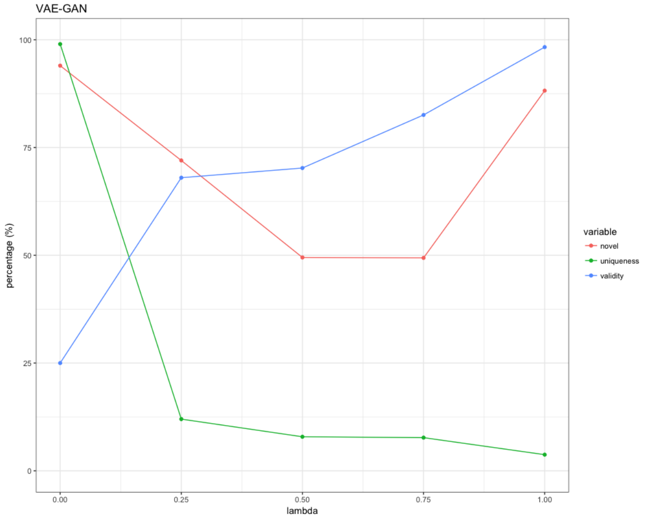

Figure 3. The performance of VAE-GAN. We trained our model with respect to the objective given in (1). The blue, red and green graphs represent the validity, novelty and uniqueness respectively of our model as we vary $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$.

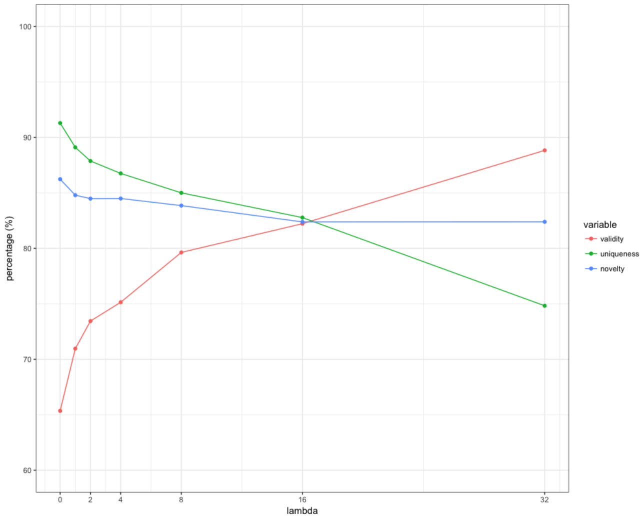

Figure 4. The performance of SCAT with validity loss over different validity loss contribution. The x-axis represents validity loss contribution hyper-parameter λ. The blue, red and green graph tracks the validity, novelty and uniqueness respectively of our model as we vary $\lambda \in \{0, 1, 2, 4, 8, 16, 32\}$.

3.4. Molecule structure

Figure 6 shows samples of generated molecules via SCAT + validity loss. The number below each molecule is the Quantitative Estimate of Drug-likeness (QED) score outputted by the RDKIT. First, it can be seen that although all of them are valid in terms of the number of bonds for each type of atoms, they might still be unstable and can not exist at all. For example, some of them has triangle substructures, which is rarely seen in the nature. On the other hand, the QED scores of the generated structures are not high enough to be drug lead. Thus, a generative model that can generate a structure with special property might be more useful in real application.

Conclusion

Our first model, VAE-GAN, combines both VAE and GAN architectures. By considering only the validity and the uniqueness metrics we see that all the in-between models are incomparable, meaning there are no two models such that the first one dominates in both metrics the second one. However if in addition we consider the novelty, Figure 3 suggest that it may be optimal to consider a purely GAN or a purely VAE model than the set of models .

SCAT with validity loss, as expected can help increase the validity rate of the generated molecule structures. Since the validity rate increase very fast, with only a slight sacrifice of the uniqueness and novelty rate, the SCAT with validity loss can finally gives a very balanced model (all the three rates are high enough).

SCAT decoder with different hidden layers.

As seen in the results as the number of layers increase in the decoder. The SCAT is able to generate more unique molecules at the price of the molecules validity. The validity of a molecules seem to be anti-correlated with the uniqueness. The novelty of the molecules generated doesn't seem to be associated with the number of layers. The number of layers can be a hyperparamter that can be searched to generate molecules with high uniqueness and high validity.

For real world application, in order to make the generated structure as realistic as possible, we should not only consider the number of bonds that can be linked to an atom, other properties like stableness of a novel structure should also be taken into account. Moreover, in some scenarios, it will be more useful to build a generative model that can produce molecule structures with special properties such as the potential to be a drug lead. Therefore, an addition reward network might be added to any of the proposed models. In addition, in order to generate some special types of small molecules like antibiotics, the specialized database needs to be developed and adversarial samples should be choosen carefully.

References

[1] Barabási, Albert-László and Albert, Réka, Emergence of scaling in random networks, Science (286),p 509–512, 1999.

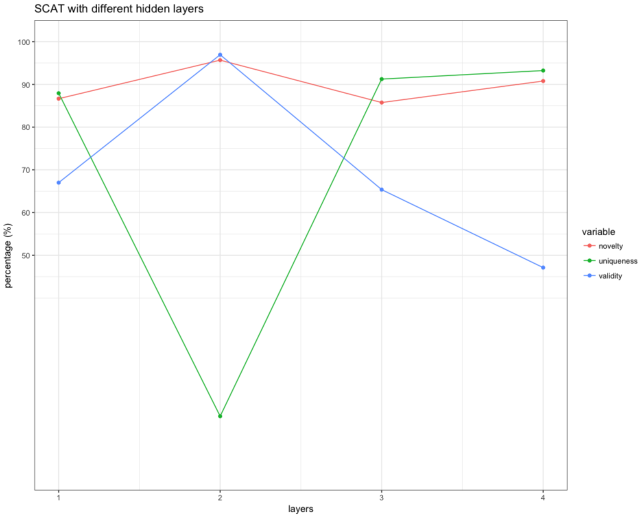

Figure 5. SCAT with different number of hidden layers

[2] Erdős, Paul and Rényi, Alfréd, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci (5), p 17–60, 1960.

[3] De Cao, Nicola and Kipf, Thomas MolGAN: An implicit generative model for small molecular graphs, arXiv preprint arXiv:1805.11973, 2018.

[4] Penrose, Mathew and others, Random geometric graphs (5), 2003.

[5] You, Jiaxuan and Ying, Rex and Ren, Xiang and Hamilton, William L and Leskovec, Jure, GraphRNN: a deep generative model for graphs, arXiv preprint arXiv:1802.08773, 2018.

[6] You, Jiaxuan and Liu, Bowen and Ying, Zhitao and Pande, Vijay and Leskovec, Jure, Graph convolutional policy network for goal-directed molecular graph generation, Advances in Neural Information Processing Systems, p. 6412–6422, 2018.

[7] Li, Yujia and Vinyals, Oriol and Dyer, Chris and Pascanu, Razvan and Battaglia, Peter, Learning deep generative models of graphs, arXiv preprint arXiv:1803.03324, 2018.

[8] Jang, Eric and Gu, Shixiang and Poole, Ben, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144, 2016

[9] Shen, Tianxiao and Lei, Tao and Barzilay, Regina and Jaakkola, Tommi, Style transfer from non-parallel text by cross-alignment, Advances in neural information processing systems, p 6830–6841, 2017

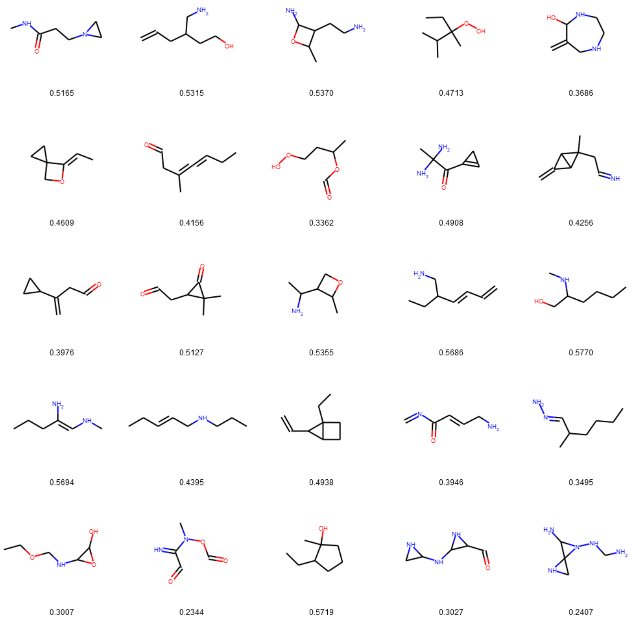

Figure 6. Molecule structures generated by SCAT with validity loss

[10] Lamb, Alex M and Goyal, Anirudh Goyal Alias Parth and Zhang, Ying and Zhang, Saizheng and Courville, Aaron C and Bengio, Yoshua, Professor forcing: A new algorithm for training recurrent networks, Advances In Neural Information Processing Systems, p. 4601–4609, 2016.

[11] Dalke, Andrew, DeepSMILES: An Adaptation of SMILES for Use in, 2018.

[12] Jin, Wengong and Yang, Kevin and Barzilay, Regina and Jaakkola, Tommi, Learning Multimodal Graph-to-Graph Translation for Molecular Optimization, arXiv preprint arXiv:1812.01070, 2018.

[13] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt, Constrained Graph Variational Autoencoders for Molecule Design, Advances in Neural Information Processing Systems, 7806-7815

[14] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. ICLR, 2016.

[15] D. P. Kingman and M. Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[16] Matt J. Kusner, Brooks Paige, Jos Miguel Hernndez-Lobato Grammar Variational Autoencoder. arXiv preprint arXiv:1703.01925, 2017

[17] Gomez-Bombarelli, R. and Wei, J. N. and Duvenaud, D. and Hernandez-Lobato, J. M. and Sanchez-

Lengeling, B. and Sheberla, D. and Aguilera-Iparraguirre, J. and Hirzel, T. D. and Adams, R. P. and Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules ACS Cent Sci, 2018

[18] Martin Simonovsky, Nikos Komodakis. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. arXiv:1802.03480, 2018

[19] Junction Tree Variational Autoencoder for Molecular Graph Generation. Wengong Jin, Regina Barzilay, Tommi Jaakkola. arXiv:1802.04364, 2018

[20] Dongmian Zou, Gilad Lerman. Encoding Robust Representation for Graph Generation. arXiv:1809.10851

[21] Dictionary of Natural Products. http://dnp.chemnetbase.com/faces/chemical/ChemicalSearch.xhtml

[22] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. Scitific data, 2014

[23] AntiBase. https://application.wiley-vch.de/stmdata/antibase.php

[24] MarinLit. http://pubs.rsc.org/marinlit/

[25] Landrum, Greg. RDKit: Open-source cheminformatics. 2006

[26] , Gulrajani, Ishaan and Ahmed, Faruk and Arjovsky, Martin and Dumoulin, Vincent and Courville, Aaron C, Improved training of wasserstein gans, Advances in Neural Information Processing Systems, p 5767-5777, 2017.

Appendix A

Old division of work

In phase I of the project, the work is going to be split between the three authors; Liu will prepossess the databases to generate the adjacency matrices for the graphs of the putative structure of natural products. Michael will identify the state of art in molecular graph generation and summarize his findings to discuss with his colleagues. Nadim will learn about the available open source framework for graph generation and start coding the first pipeline. The three authors will further discuss the prepossessing, literature reviews and the coding steps with Hao Zhang. The discussion will be important to tackle any significant designs issues early on. The three authors will meet regularly the continuous development of the methods, to evaluate designs and to decided on the needed experiments. The coding and development of the deep generative model will be distributed between the authors.

By the end of phase I, the deliverables will be : the finalized versions of introduction and background sections , a method section that contains the initial progress, the development of initial version of the deep generative model for 2D structure of natural product with preliminary results and a revised plan of activities for the final report. Therefore by the end of Phase 1, we expect to be able to cross all the checklist of the Midway report.

After the Midway report and during Phase II (March 29-April 20) the authors will further discuss the preliminary results with Hao Zhang and decide on the final design of the algorithm to adopt. The authors will write and run sets of experiments to verify further and validate their model. The methods and experiments sections will be written during phase II.

At the final Phase III, the authors will wrap the project, finish the write-up and add their conclusion and future work. The main effort during Phase III is to iterate on the final report until it is well polished and revised.

Revised plans of activity

We were able to comply with most of the activities suggested in the old division of work. However, we expected, to start experimenting with our models and to have prelimnary results by the midway report. During our group meeting, we acknowledged this delay and we decided to put more time in the project for the two weeks after the midway report. We will focus on implementing and experimenting on models 0, 1 2 and 3 . Models 0-3 have faster training time than model 4 , therefore we can iterate faster on the design of 4. Model 4 is going to be slow to train on a single GPU, we will try to distribute the training on multiple GPUs by modifying the CGVAE code.

We expect to compensate the lost time of Phase I and in Phase II, therefore we will discuss our prelimnary results with Hao Zhang, and pick the best model out of four that is worth validating and optimizing. Therefore, Phase II will be: Two weeks after, the midway report, we expect to generate preliminary results for Models 0, 1, 2 ,3 and 4. We will select the best model depending on the validity and uniqueness of the molecules generated. Then,The authors will write and run sets of experiments to verify further and validate their model. The methods and experiments sections will be written during phase II.

Final Phase III will be as above: ”At the final Phase III, the authors will wrap the project, finish the write-up and add their conclusion and future work. The main effort during Phase III is to iterate on the final report until it is well polished and revised.”