Combining variational autoencoder representations with structural descriptors improves prediction of docking scores

Miguel Garcia-Ortegon
DPMMS
University of Cambridge
Wilberforce Rd, Cambridge, UK
mg770@cam.ac.uk

Andreas Bender
Dept. of Chemistry
University of Cambridge
Lensfield Rd, Cambridge, UK
ab454@cam.ac.uk

Carl E. Rasmussen
Dept. of Engineering
University of Cambridge
Trumpington St, Cambridge, UK
cer54@cam.ac.uk

Hiroshi Kajino
IBM Research - Tokyo
19-21 Hakozaki, Chuo-ku, Tokyo, Japan
kajino@jp.ibm.com

Sergio Bacallado
DPMMS
University of Cambridge
Wilberforce Rd, Cambridge, UK
sb2116@cam.ac.uk

Abstract

Molecular hypergraph grammar variational autoencoders are a generative model with great potential for *de novo* design in drug discovery. A numerical experiment with the aim of predicting docking scores for the D2 dopamine receptor shows that combining representations from this model with structural descriptors of the docking pose attains state-of-the-art performance. The results suggest that incorporating structural information in the training of variational autoencoders could lead to better representations and accelerate guided virtual screening.

1 Introduction

High-throughput and *in silico* screening technologies in drug discovery are typically based on chemical libraries. It is thought that this can severely limit the diversity of compounds available at the lead optimisation phase relative to the combinatorially large space of small molecules. Because of this, there has been great interest for several years on automating *de novo* design of drug-like molecules using generative models. Since their introduction in chemoinformatics [1], variational autoencoders (VAEs) have shown promise in active or guided virtual screens using Bayesian optimisation policies. The key properties of the representation produced by a VAE are (i) the capacity to map bijectively from molecule to latent representation and vice versa, and (ii) the smoothness of chemical properties on the latent space, which allows regression of the optimisation objective onto the representation. There has been much progress in recent years on improving the reconstruction or self-encoding rates

Machine Learning for Structural Biology Workshop, NeurIPS 2020, Vancouver, Canada.

of VAEs through the use of formal grammars for chemicals [2, 3, 4], as well as developing semi-supervised methods to improve the predictive power of latent vectors by augmenting the variational objective with a regression loss, in an approach known as train-with-predict [1, 4].

However, learning VAE representations that are highly predictive of a physical property, such as binding affinity to a protein target, is a difficult task. This is because the procedure is largely unsupervised and ignores any physical description of the molecules' 3D structure — it would amount to learning the laws of physics from a library of molecules and a single quantitative property. This would likely require datasets several times bigger than the largest existing libraries. Indeed, to date, most experiments with molecular VAEs have been restricted to fairly simple properties [1, 2, 3, 4, 5], and it isn't clear that the representations produced by VAEs are superior in library-based virtual screens to carefully engineered features such as molecular fingerprints. In this sense, the ability of VAEs to harness higher-order features of 3D structure, as opposed to the type of graph convolutional features obtained from molecular fingerprints, has yet to be definitively proven.

This paper provides evidence that combining VAE representations and structural descriptors of a small molecule can improve predictive power. Docking simulations were used to generate descriptors, as they provide a rich characterisation of a molecule's 3D structure and electrostatics while being computationally feasible for libraries of 10^5 or 10^6 compounds. We demonstrate that state-of-the-art predictions for docking scores are attained by combining structural descriptors with latent vectors from a molecular hypergraph grammar VAE. In particular, we observe enhanced operating characteristics for classification which are essential for guided screening. The implications of these results for *de novo* design and guided virtual screening are discussed in the final section.

2 Molecular hypergraph grammar VAE

The molecular hypergraph grammar variational autoencoder (MHG-VAE) encodes molecules in the form of a sequence of rules from a molecular hypergraph grammar (MHG) [4]. This follows the example by [2], who designed a similar model using a SMILES grammar. During decoding, later rules in the sequence are conditioned on previous rules, such that those that are incompatible with previous ones are masked out. In contrast to previous grammars, the MHG is remarkable in that it achieves 100% valid molecules when sampling from the latent space [4].

A MHG is a hyperedge replacement grammar (HRG) [6] whose hypergraphs represent molecules. Perhaps surprisingly, hyperedges correspond to atoms (labeled with atomic element, formal charge and a tetrahedral chirality tag) and vertices correspond to bonds (labeled with the bond type and a E-Z configuration tag for double bonds) [4]. In addition to achieving perfect validity, the rules of the MHG are learnt automatically, so they are less susceptible to human biases.

More specifically, a HRG is a tuple $G = (N, T, S, \mathcal{P})$, where:

- N is a finite set of non-terminal symbols. Each non-terminal symbol will be expanded into other symbols by applying a production rule.
- T is a finite set of terminal symbols which end the expansion process.
- $S \in N$ is a non-terminal symbol that acts as a starting symbol.
- $\mathcal{P}$ is a finite set of production rules $L \to R$, where L is a non-terminal symbol and R could be terminal or non-terminal.

Production rules are obtained from a corpus of hypergraphs that are represented as clique trees. Each node of each clique tree will give rise to a single production rule. The left-hand side L contains the connections to the parent node, while the right-hand side R is a copy of the node. Thus, by applying a sequence of production rules one can reconstruct, node by node, any clique tree in the corpus. In addition, novel sequences of rules can give rise to a great diversity of novel hypergraphs outside of the corpus [6].

3 Experiments and results

3.1 Model system

As a target receptor protein for our proof of concept, we chose dopamine receptor D2. D2 was appealing for the following reasons: first, it is well studied, and a high-quality crystal structure bound to risperidone has been published [7]; second, it is medically relevant, since it is used widely as a target to treat mental health diseases and there exist several approved drugs against them (one of which is risperidone); and third, it has been used as a benchmark for *de novo* molecule generation in the literature [8, 9]. Our ligand set is a subset of the ExCAPE database whose activity against D2 has been measured experimentally [10]. We took 50,000 compounds, of which 4,613 were active and the rest were inactive.

3.2 Receptor preparation and docking

A PDB file with the structure of D2 was obtained from a published crystal [7], and was minimized using the GAFF force field [11]. Protonation was done on the PDB2PQR server [12]. PDB files for the ligands were obtained with RDKit [13]. PDB files were converted to PDBQT using Autodock Tools [14]. Docking of each ligand against D2 was carried out with Autodock Vina [15], and the highest-ranking score and pose of each ligand were selected for downstream experiments. Docking proceeded successfully for 49,983 ligands.

3.3 Voxelization

In order to make them amenable to convolutional filters, docking poses were voxelized using the `moleculekit` package [16] with a grid resolution of 1Å^3. `moleculekit` represents molecules in 8 channels corresponding to different atom categories: hydrophobic, aromatic, H-bond acceptor, H-bond donor, positive ionizable, negative ionizable, metal, and occupancy (Figure 1). Each channel roughly captures whether a point on the grid is within a Van der Waals radius of an atom of the corresponding category. Voxelization was successful for 49,975 ligands.

3.4 Timeline-based data split

Given that chemical databases typically contain close structural analogs that exhibit very similar properties, a random split of the dataset is not recommended because it may result in data leakage. To avoid this, several strategies exist, such as cluster-based or timeline-based splits. We adopt the latter approach, where molecules are ordered according to their date of discovery, so that the training set contains the oldest molecules and the validation and test sets contain newer molecules. Thus, this split simulates a prospective validation. As a proxy for the date of discovery, we used the date of deposit of each ligand to PubChem [17], since that is a good representation of when a molecule became widely available to the scientific community. A date of deposit could be assigned to 44641 ligands. Finally, we chose an 80% training, 10% validation and 10% test split. For more information on the data split, see Appendix B.

3.5 Unsupervised latent vectors display low predictive performance

As a baseline, we trained a MHG-VAE on the unlabeled dataset, without using docking score or pose information. This is the usual unsupervised training regime followed by most VAEs. Details on the architecture and training parameters are given in the appendices. Then, we computed a latent vector for each molecule and trained a simple linear regressor of the docking score on the latent vectors. Finally, the regressor was evaluated on the unseen test set. For comparison, we trained a similar linear model based on molecular fingerprints (Morgan, radius 3) [13].

Our results, shown in Table 1, suggest that unsupervised latent vectors are not highly predictive of the docking score, although their performance is still higher than that of a dummy regressor that predicts the mean (which would attain an R^2 of 0). In comparison, molecular fingerprints show moderate predictive power.

Molecular representation	R^2	Average Precision	Reconstruction rate
Fingerprints (FP)	0.562	0.539	-
Unsupervised latent vectors (ULV)	0.248	0.317	**88.1**%
Train-with-predict latent vectors (TLV)	0.508	0.440	27.8%
CNN-processed voxels (VX)	0.550	0.527	-
ULV + FP	0.595	0.531	-
ULV + VX	0.656	0.597	-
TLV + FP	0.645	0.550	-
TLV + VX	**0.697**	**0.638**	-

Table 1: Predictive performance attained with various molecular representations, and reconstruction rates of VAEs.

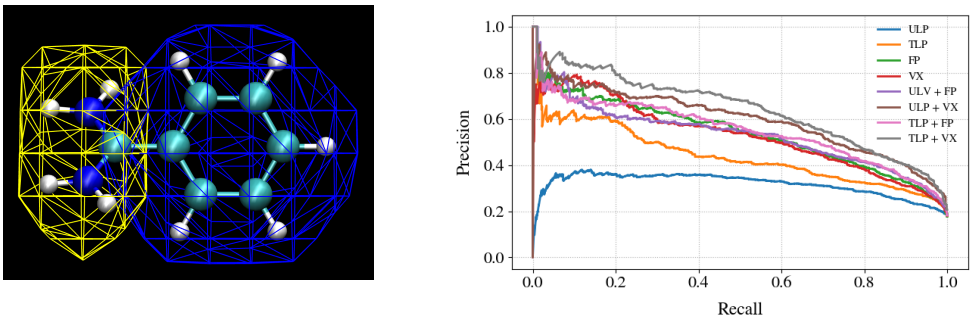
3.6 Train-with-predict induces latent vectors with competitive performance

In order to make latent vectors more amenable to docking score regression, we followed the train-with-predict approach of [1]. This regime adds a regression module to the architecture and optimises the usual variational objective together with a regression loss for the docking score. For details of the regression module see the appendices. Once the model was trained, we again computed latent vectors for all molecules, trained a linear regression model, and evaluated it on the test set. The results in Table 1 show that a train-with-predict regime produces latent vectors with performance comparable to that of fingerprints.

3.7 State-of-the-art performance can be achieved by adding structural information

A key aspect of our proposed strategy for guided virtual screening is improving latent vectors with structural information about the ligands. As a way to evaluate the potential of this approach, we concatenated each train-with-predict latent vector to another vector representing the ligand's voxelized pose. The latter vectors were formed by the activations of the last layer of a convolutional neural network (CNN) trained to predict docking score from the voxels, so they encapsulated the information about docking score contained in the voxels. For details of the CNN architecture, see Appendix A.3. Again, we trained a linear regressor and evaluated it on the test set. For comparison, we also evaluated the performance of the CNN-derived vectors alone, and the combination of train-with-predict latent vectors with molecular fingerprints.

The results in Table 1 suggest that a train-with-predict regime for the docking scores can be combined with structural information about the pose to achieve state-of-the-art predictions.



(a) (b)

Figure 1: (a) An example voxelized molecule (benzamidine) displaying the channels H-bond donor (yellow) and aromatic (blue). (b) Precision-recall curves of a linear SVC trained on the different representations. As it can be seen, a combination of train-with-predict latent vectors and voxels achieves the best performance.

3.8 Our representation allows identification of positives with high precision and recall

In practice, rather than performing numeric regression of the docking score, it is more useful to determine whether a compound is active or inactive, given that hit molecules will be validated and optimised in subsequent steps and the exact numeric score may change. To evaluate whether our representation was capable of distinguishing actives, we binarised our dataset, choosing a threshold such that 10% of molecules received active labels. Then, we plotted precision-recall curves for a linear support vector classifier (SVC) on each of the representations (Figure 1). Our results suggest that a combination of train-with-predict latent vectors and voxelized poses can be used to effectively identify active molecules in the virtual screen.

4 Discussion and future work

De novo design in virtual screening requires generative models in which the latent representation is chemically informative. Here, we provide evidence that docking poses improve the predictive power of VAE representations, even when they are trained in a semi-supervised fashion and combined with molecular fingerprints. The results presented here encourage the use of docking poses within the training of a VAE, for example, by explicitly including a train-with-predict objective linking the latent representation with a voxelization of the docking pose. We believe this approach could significantly accelerate virtual screens, by shaping the latent space more directly with features of the ligand's 3D structure. There is reason to believe that representations trained on a specific docking target could be transferrable and prove advantageous in guided optimisation of other properties — most immediately, expensive free energy perturbations based on molecular dynamics, but also affinities for different targets, and even phenotypic objectives. These questions are the subject of ongoing work.

References

[1] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[2] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. volume 70 of *Proceedings of Machine Learning Research*, pages 1945–1954, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[3] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. In *International Conference on Learning Representations*, 2018.

[4] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. volume 97 of *Proceedings of Machine Learning Research*, pages 3183–3191, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[5] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[6] Salvador Aguiñaga, David Chiang, and Tim Weninger. Learning hyperedge replacement grammars for graph generation. *CoRR*, abs/1802.08068, 2018.

[7] Sheng Wang, Tao Che, Anat Levit, Brian K Shoichet, Daniel Wacker, and Bryan L Roth. Structure of the d2 dopamine receptor bound to the atypical antipsychotic drug risperidone. *Nature*, 555(7695):269–273, 2018.

[8] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

[9] Josep Arús-Pous, Atanas Patronov, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Smiles-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics*, 12(1):1–18, 2020.

[10] Jiangming Sun, Nina Jeliazkova, Vladimir Chupakhin, Jose-Felipe Golib-Dzib, Ola Engkvist, Lars Carlsson, Jörg Wegner, Hugo Ceulemans, Ivan Georgiev, Vedrin Jeliazkov, et al. Excape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of cheminformatics*, 9(1):17, 2017.

[11] Junmei Wang, Romain M Wolf, James W Caldwell, Peter A Kollman, and David A Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004.

[12] Todd J Dolinsky, Paul Czodrowski, Hui Li, Jens E Nielsen, Jan H Jensen, Gerhard Klebe, and Nathan A Baker. Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic acids research*, 35(suppl_2):W522–W525, 2007.

[13] Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[14] Garrett M Morris, Ruth Huey, William Lindstrom, Michel F Sanner, Richard K Belew, David S Goodsell, and Arthur J Olson. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[15] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[16] S Doerr, MJ Harvey, Frank Noé, and G De Fabritiis. Htmd: high-throughput molecular dynamics for molecular discovery. *Journal of chemical theory and computation*, 12(4):1845–1852, 2016.

[17] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019.

[18] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Appendices

A Neural architectures

A.1 MHG-VAE

The encoder and decoder modules of the MHG-VAE consisted of 3 bidirectional GRU layers, each of which was 256 units wide and had a 0.1 dropout rate. The latent space was 56-dimensional. Rules were encoded with a 256-dimensional embedding. Implementation for this and other neural models was done in Pytorch.

A.2 Train-with-predict MHG-VAE

The configuration of the train-with-predict MHG-VAE was the same as for the MHG-VAE, except for the introduction of an additional regression module which predicted docking scores from the latent space $\hat{f} : \mathbb{R}^{56} \to \mathbb{R}$. This module was a single layer of neurons without activations, i.e. linear regression. The regression loss function of choice was mean squared error, and was minimized together with the ELBO.

A.3 CNN

Voxelized poses were passed through a CNN that was trained to predict docking scores. This CNN consisted of three 3D convolutional layers, with 8, 6 and 4 input channels respectively; 6, 4 and 4 output channels, and a kernel size of $3 \times 3 \times 3$. Convolutional layers were followed by 3 fully-connected layers with leaky relu activation, of width 500, 100 and 20 respectively. Activation values of the 20-dimensional fully-connected were saved to be used later as CNN-processed voxels (see section 3.7). The final layer of the model was a single neuron with no activation, and the regression loss was mean squared error (equivalent to linear regression on the processed voxels).

B Model training

B.1 MHG-VAE

The MHG-VAE was trained for 30 epochs using the optimizer Adam, a learning rate of $5 \cdot 10^{-4}$ and a batch size of 128. The training, validation and test sets described in section 3.4 were used, and hyperparameters were tuned based on the validation set. Since labels were not necessary in this model (as opposed to the train-with-predict MHG-VAE), the training set was augmented with 500000 molecules from ZINC [18].

B.2 Train-with-predict MHG-VAE

The training of the train-with-predict MHG-VAE was the same as for the MHG-VAE but ZINC molecules were not added to the training set because a docking score was needed for every ligand.

B.3 CNN

The CNN was trained for 100 epochs with the optimizer Adadelta, a learning rate of 0.1 and a batch size of 256.

B.4 Linear regression

The linear regression models in sections 3.5, 3.6, 3.7 were trained with the combined training and validation sets, which amounted to 90% of the dataset. The implementation used `scikit-learn`.

B.5 Support vector classifier (SVC)

The SVC in section 3.8 was trained with 5-k cross validation on the combined training and validation sets. The implementation used `scikit-learn`.