Supplementary Materials

Exploring Deep Learning for Metalloporphyrins: Databases, Molecular Representations, and Model Architectures

An Su [a]†, Chengwei Zhang [a]†, Yuan-Bin She [a], Yun-Fang Yang [a]**

a. College of Chemical Engineering, Zhejiang University of Technology, Hangzhou 310014, China;

* Correspondence: ansu@zjut.edu.cn (A.S.); yangyf@zjut.edu.cn (Y.-F.Y.)

† These authors contributed equally to this work.

Detailed explanation of PBDD parameters

The porphyrin molecules stored in the database are stored in multiple parts. These include three side groups (**R1, R2, R3**), an anchoring group(**A**), the main structure of the porphyrin and the central metal with the axial ligand(**M**).

In the key **R1，R2，R3**, the code names of the side groups are stored. The corresponding relationship between the structure and the code name of the side group is shown in

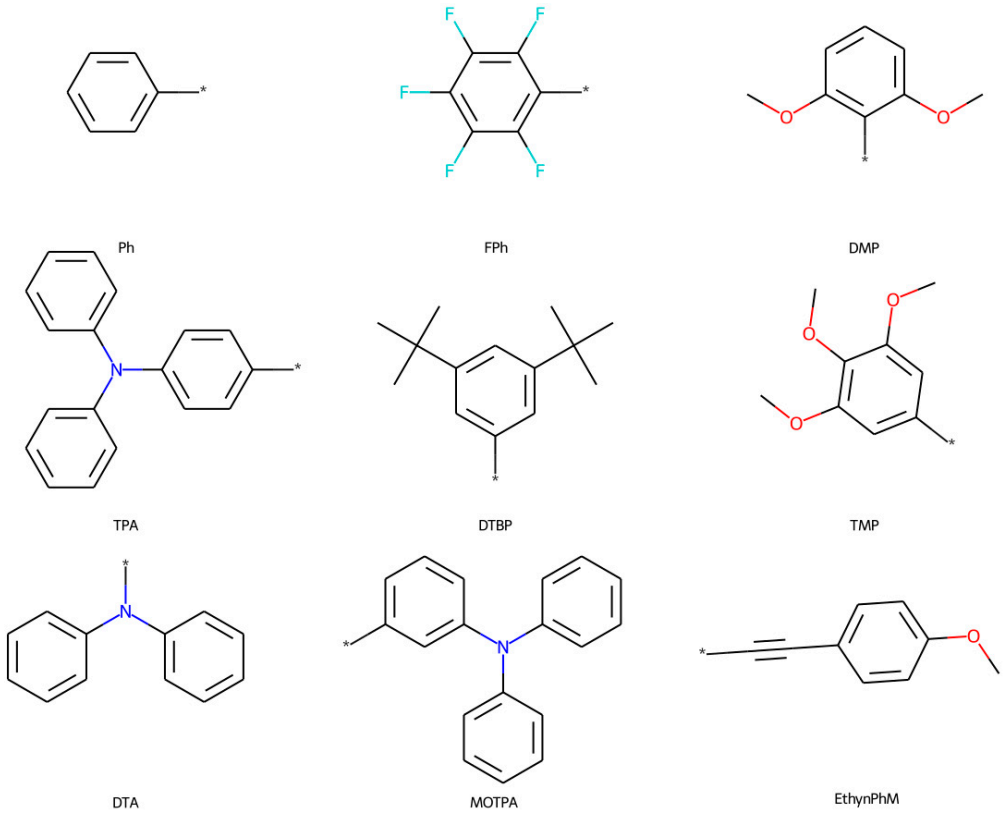

Figure S1. Each side group in each porphyrin molecule will independently select a structure from it and connect with the main part of the porphyrin molecule at "*".

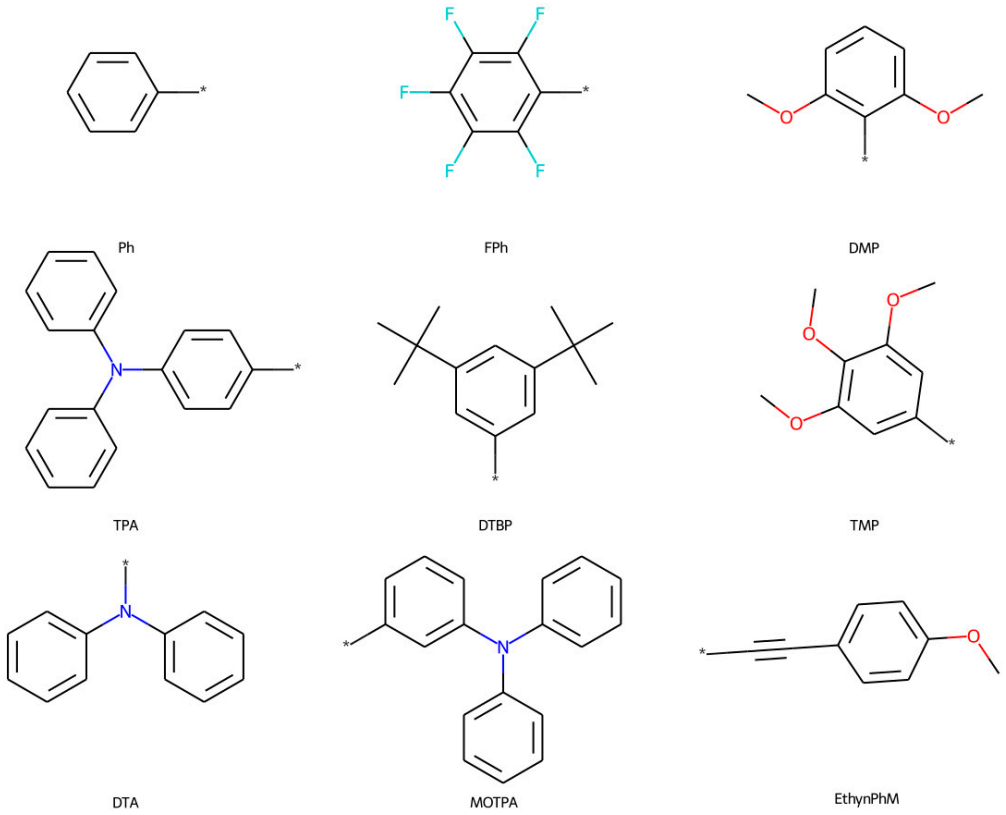

Figure S1 Display of side groups and their code names

Similar to the side group, key **A** records the code name of the anchor group of the porphyrin molecule. The code and structure of the anchor group are shown in Figure S2, and "*" indicates the connection with the main body of the porphyrin molecule.

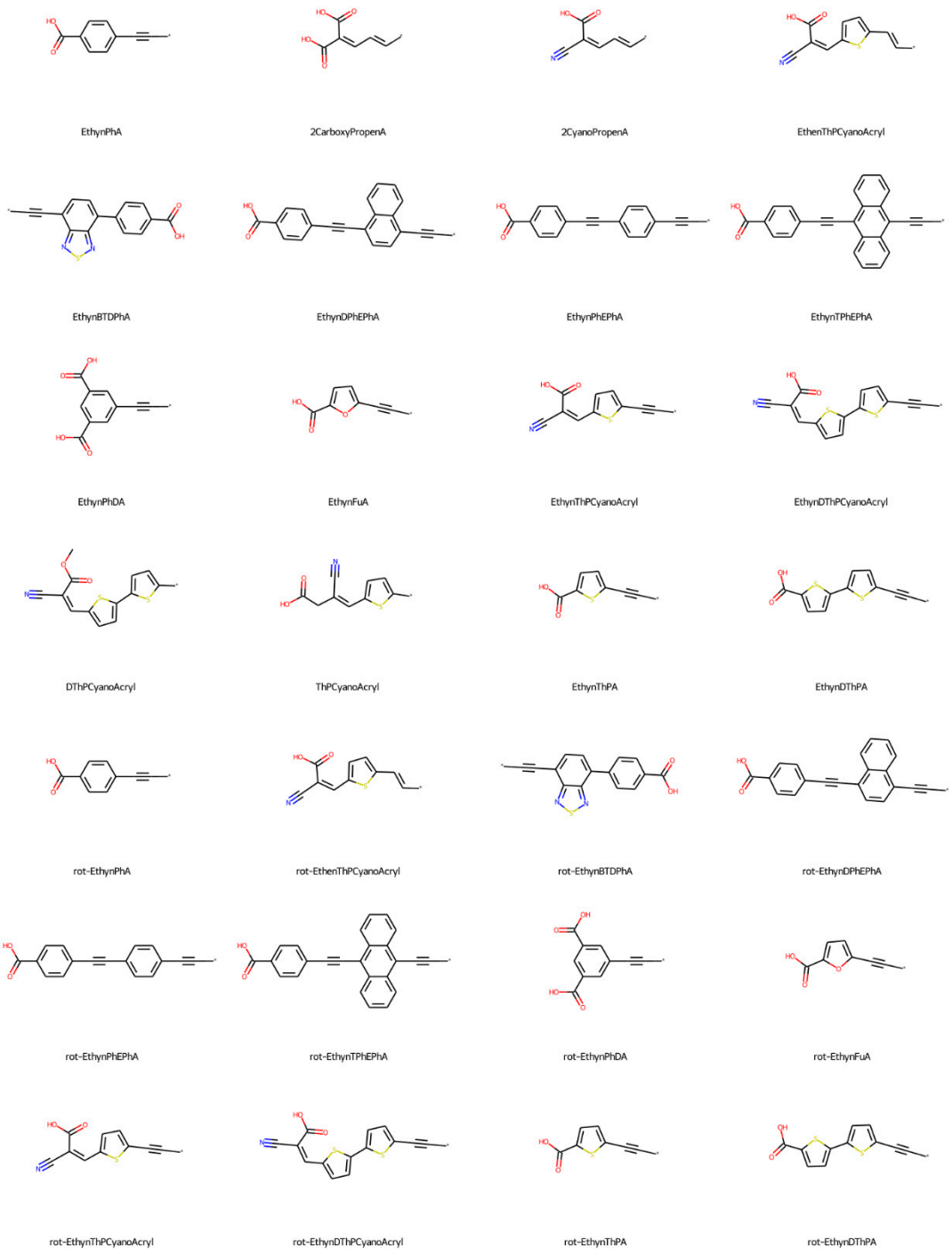

Figure S2 Anchor group and its code display

The codes stored in the key **M** correspond to the main structure of the porphyrin molecule and the central metal and the axial ligands on the central metal. There are a total of 6 kinds of codes, which are “H2P” “ZnP” “FZnP” “TiOP” “FTiOP” “TiO2RP”.

When M is "FZnP" or "FTiOP", all the hydrogen atoms on the 8 β positions of the porphyrin molecule are replaced by fluorine atoms. When M is "H2P", the porphyrin molecule has no central metal. When M is "ZnP" or "FZnP", the central metal of the porphyrin molecule is a zinc atom, and there is no axial ligand. When M is "TiOOP" or "FTiOP", the central metal is titanium, and the axial ligand is hydroxyl, as shown in the middle of the Figure S3. When M is "TiO2RP", its central metal and axial ligands are the structures shown on the right side of the Figure S3.

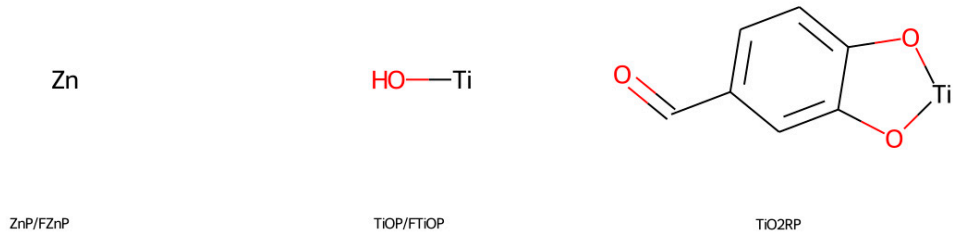

Figure S3 The code name of the key M and its corresponding structure

Method – Model Architecture

There are many different types of models used in the field of deep learning, and there are two main types of models that have excelled in the field of chemistry. One is the graph structure-based neural network, which treats the molecular structure formula as a non-Euclidean graph, with the atoms that make up the molecule as points and the bonds between atoms as edges. In this type of deep learning model, we have chosen three models: graph convolutional neural networks, message passing neural networks

and directed message passing neural networks.

Graph Convolutional Neural Networks(GCN) A graph neural network is a process that propagates information about nodes and edges in a non-Euclidean graph, and then compares the results of multiple propagations with existing results to update the parameters in the model for training purposes. A simplification of the steps in a graph neural network can be as follows.

Denote the eigenvector of a node V as X_v, and the eigenvector of the edge associated with the node V as $X_{CO[v]}, X_{ne[v]}$ denoting the eigenvector of the neighboring nodes of the node V.

h_v denotes the state embedding with node V. Obtaining the state embedding of each node is the learning goal of the graph neural network. $h_{ne[v]}$ means the state embedding of the neighboring nodes of node V.

Let the function f be used to update the state embedding of a node based on the information of the node and neighboring nodes, called the local transfer function, i.e.

$$h_v = f\left(X_v, X_{CO[v]}, h_{ne[v]}, X_{ne[v]}\right)$$

Let function g be a local output function that maps the resulting node state embedding to the data labels of downstream tasks, i.e.

$$o_v = g\left(h_v, X_v\right)$$

The global transfer function F and the global output function G can be obtained by iterating the above two functions over all the nodes and superimposing them several times, i.e.

$$H^{t+1} = F\left(H^t, X\right)$$
$$O = G\left(G, X_N\right)$$

The model is a GCN model when the transfer function F and G in the above equation are functions associated with convolution.

In this paper, we used GCN from DeepChem[1] to train the PDBB, using the method mentioned in this literature to characterize the molecules[2].

Message Passing Neural Network (MPNN) The MPNN is developed from the spatial domain convolution in the traditional graph convolutional neural network, which abstracts GCN into a deep learning framework consisting of two phases, the message passing phase U and the readout phase R. The formula is shown below:

$$h_v^t = U\left(h_v^{t-1}, \sum_{w \in N(v)} M_t\left(h_v^{t-1}, h_w^{t-1}, e_{vw}\right)\right)$$

$$O = R\left(h_v^T \mid v \in G\right)$$

In this paper, we used an MPNN model built from the *Keras* library[3] and trained the data. In this model, we used the *rdkit* library[4,5] to extract features from the molecules, including atomic features, bond-forming atoms and chemical bond features.

Directed Message Passing Neural Network(D-MPNN) DMPNN is a further development of MPNN. Rather than using messages associated with vertices (atoms), D-MPNN uses messages associated with directed edges (bonds). Compared to the atom based message passing approach, this message passing procedure is more similar to belief propagation in probabilistic graphical models.[6] We used the model mentioned in this literature,[7] which can read both atomic and chemical bond information and molecular descriptor information of a molecule.

String-based Model

Another class of models is the Transformer and BERT models for NLP, which treat molecular representation as a language and molecular property prediction as a neural network-based translation problem. Simply put, it is the translation of SMILES into molecular properties. Next a brief introduction to Transformer and BERT.

Transformer Transformer[8] is a relatively new class of NLP models based entirely on *Attention* mechanisms[9]. The formula for *Attention* is shown as follows:

$$Attention = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q, K, V are input embedding matrices and d_k is the embedding dimension. Its unique model architecture allows it to have better performance and better interpretability than

traditional RNN, seq2seq and other models when dealing with chemical molecular formulae. We used the Transformer provided in *SimpleTransformer.ai*[10] to train the data.

BERT BERT[11], known as Bidirectional Encoder Representation from Transformers, is a pre-trained language representation model. It emphasizes the use of the new masked language model (MLM) instead of the traditional unidirectional language model or the shallow splicing of two unidirectional language models for pre-training as in the past. This architecture makes it somewhat more powerful than the Transformer model in terms of performance.

In this work we use the BERT model built by Philippe Schwaller[12], which called rxnfp. This model was originally used for chemical reaction yield prediction[13], but with some simple adjustments we used it for molecular property prediction.

Transfer Learning

Our chosen database stores data on more than 12,000 porphyrins, which is still slightly inadequate when compared to other open-source chemical databases such as QM7, QM9 and tox21. Therefore, we also investigated the impact of transfer learning on the generalization ability of the model. Transfer learning is divided into two steps, pre-training and fine-tuning, where a larger dataset is first used for multiple rounds of training and the parameters of the completed model are saved, called the pre-trained

model. The pre-trained model is then trained with a smaller dataset (in this paper,PBDD), and the parameters of the pre-trained model are adjusted slightly to obtain a better performing model.

We used several models provided in ChemBERTa[14], which are pre-trained models trained using open source datasets such as those provided in PubChem, which can be invoked for training via *Hugging face*[15] and *SimpleTransformer.ai.*

In addition, we randomly selected one million SMILES of organic molecules from the open source database ZINC15[16] and did unsupervised learning on the BERT model, and the resulting model was used as a pre-trained model for the BERT model.

In summary, we have chosen GCN, MPNN and D-MPNN models in the direction of graph neural networks, and transformer and BERT models in the direction of NLP, and compared the effectiveness of these two models for transfer learning.

TreeMAP

TMAP is a very fast library for visualizing large, high-dimensional datasets, allowing us to very easily downscale and visualize high-dimensional molecular features to two dimensions and presents this two-dimensional data in the form of a tree diagram. [17]This tree-based layout helps chemists to better find relationships between molecules in chemical space by clearly showing the closest distances between clusters

and by showing the detailed structure of clusters through branches and sub-branches

Model evaluation The data used in this paper are all labelled data, so the most widely used regression coefficients (R^2), root mean square error (RMSE) and mean absolute error (MAE) in supervised learning are used to evaluate the model results. Its calculation formula is as follows：

Assuming that there are a total of n known observations of $y_1, y_2, y_3 \cdots$ in a data set, the corresponding predicted value $f_1, f_2, f_3 \cdots$ is obtained after prediction by the model.

$$R^2 \equiv 1 - \frac{\sum_{i=1}^{n} (y_i - f_i)^2}{\sum_{i=1}^{n} \left(y_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right)^2}$$

$$RMSE \equiv \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - f_i)^2}$$

$$MAE \equiv \frac{1}{n} \sum_{i=1}^{m} |y_i - f_i|$$

Results

Model performance comparison summary To make model performance data more convincing, we performed 10 parallel training sessions using each of the screened deep learning models, and after averaging the data from the 10 training sessions, we obtained the results shown in Figure S4

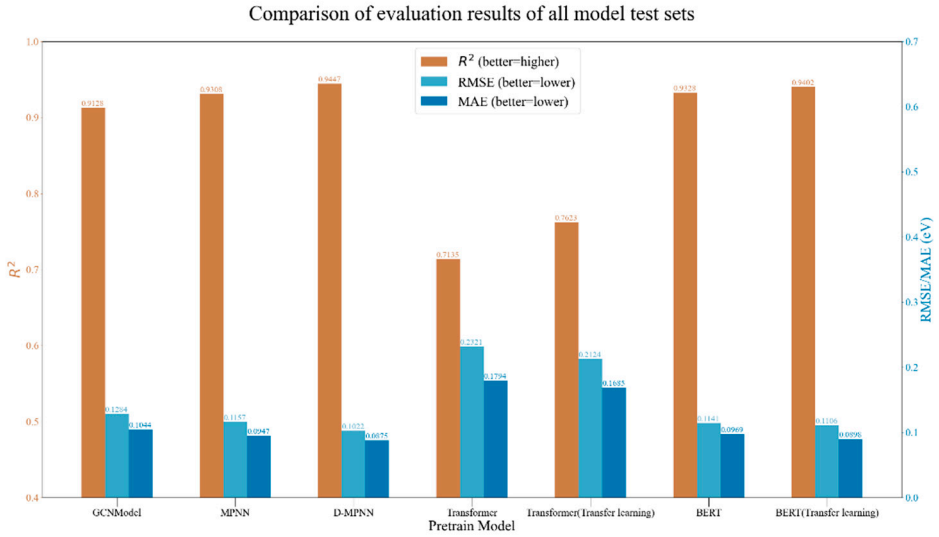

Figure S4 Comparison of evaluation results of all model test sets

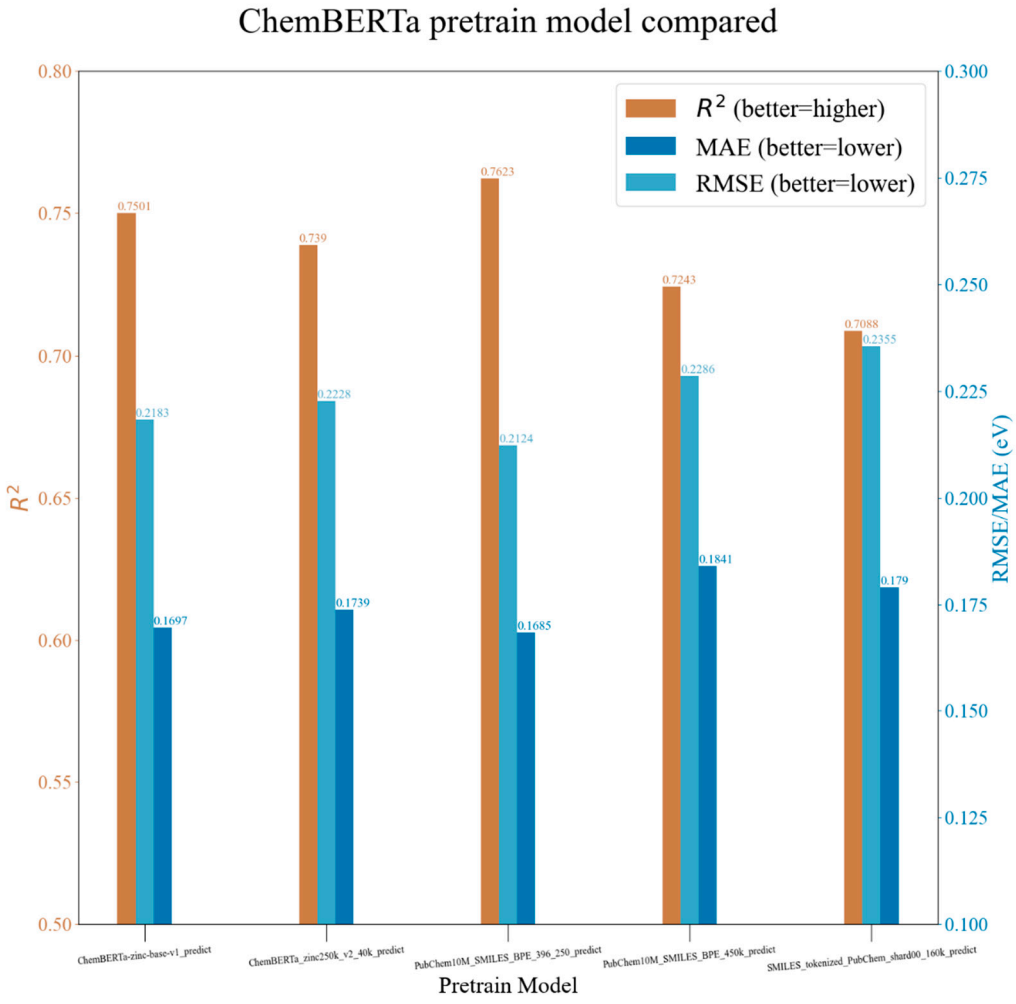

Figure S5 Comparison of results of ChemBERTa pre-trained models.

Reference

1. Eastman;, B.R.P. *Deep Learning for the Life Sciences*, 1st edition ed.; O'Reilly Media, Inc.: Sebastopol, California, United States, 2019.

2. Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph

convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* **2016**, *30*, 595-608.

3. Chollet, F.; others. Keras. Available online: https://keras.io (accessed on 18 November 2022).

4. Landrum, G.; Tosco, P.; Kelley, B.; Ric; sriniker; gedeck; Vianello, R.; NadineSchneider; Kawashima, E.; Dalke, A.; et al. *rdkit/rdkit: 2022_03_4 (Q1 2022) Release*; Zenodo.

5. Landrum, G. RDKit: Open-source Cheminformatics. Available online: http://www.rdkit.org. (accessed on 18 November 2022).

6. Koller, D.; Friedman, N. *Probabilistic graphical models: principles and techniques*; MIT press: 2009.

7. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling* **2019**, *59*, 3370-3388, doi:10.1021/acs.jcim.9b00237.

8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.

9. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning

to align and translate. *arXiv preprint arXiv:1409.0473* **2014**.

10. Rajapakse, T.C. Simple Transformers. Available online: https://github.com/ThilinaRajapakse/simpletransformers (accessed on 18 November 2022).

11. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* **2019**, *abs/1810.04805*.

12. Schwaller, P.; Vaucher, A.C.; Laino, T.; Reymond, J.-L. Prediction of chemical reaction yields using deep learning. *Machine learning: science and technology* **2021**, *2*, 015016.

13. Schwaller, P.; Vaucher, A.C.; Laino, T.; Reymond, J.-L. Data augmentation strategies to improve reaction yield predictions and estimate uncertainty. **2020**, ChemRxiv preprint at chemrxiv.13286741.

14. Chithrananda, S.; Grand, G.; Ramsundar, B. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885* **2020**.

15. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* **2019**.

16. Sterling, T.; Irwin, J.J. ZINC 15 – Ligand Discovery for Everyone. *Journal of*

Chemical Information and Modeling **2015**, *55*, 2324-2337, doi:10.1021/acs.jcim.5b00559.

17. Probst, D.; Reymond, J.-L. Visualization of very large high-dimensional data sets as minimum spanning trees. *Journal of Cheminformatics* **2020**, *12*, 12, doi:10.1186/s13321-020-0416-x.