Supporting Information

MolMiner: You only look once for chemical structure recognition

Youjun Xu,∗,†,§ Jinchuan Xiao,†,§ Chia-Han Chou,†,§ Jianhang Zhang,† Jintao Zhu,† Qiwan Hu,† Hemin Li,† Ningsheng Han,† Bingyu Liu,† Shuaipeng Zhang,† Jinyu Han,† Zhen Zhang,† Shuhao Zhang,† Weilin Zhang,† Luhua Lai,∗,‡,¶ and Jianfeng Pei∗,†

∗Infinite Intelligence Pharma, Beijing, China, 100083
‡Center for Quantitative Biology, Peking University, Beijing, China, 100871
¶BNLMS, Peking-Tsinghua Center for Life Sciences at the College of Chemistry and Molecular Engineering, Peking University, Beijing, China, 100871
§These authors contributed equally

E-mail: xuyj@iipharma.cn; lhlai@pku.edu.cn; jfpei@pku.edu.cn
Fax: (+86)10-62751725; (+86)10-62759595

Supplementary Datasets and Methods

Dataset Generation and Annotation

Dataset for MolMiner-ImgDet. The MolMiner-ImgDet dataset was automatically generated by Reportlab (an open source Python package of PDF generation). The parameters, about font face and size, column number (1 or 2), and vertical space of a line or paragraph,
are used to control the layout of styled images (from MolMiner-ImgRec) and English corpus
(https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2) on one PDF-format page. A total of 3 thousand PDF-format pages with a large size of 3306 × 4674 were generated and used to develop the MolMiner-ImgDet model.

**Dataset for MolMiner-ImgRec.** We randomly selected from the ChEMBL29 dataset \(^1\) (with removing the molecules in the four benchmark datasets) to construct the training/validation dataset (a ratio of 8:2) for the MolMiner-ImgRec model with a modified drawing code based on the RDKit v2021.09.1 toolkit. This code is used to control the parameters of bond line width, dash line count, line width, gap between bond and char, char font face and size, supergroup of right- and left-align, text with subscripts. With a random parameter to control the proportion of molecular images with different styles, we generated 170 thousand SVG-format images, and then converted them into PNG-format images. An SVG-format image is a vector-based image that can be parsed to char (atom and supergroup) and bond objects. These parsed objects of chars and bonds can be used as precise annotations of chars and bonds that can be easily trained for the recognition model.

**Dataset for MolMiner-TextOCR.** We used about 250 thousand annotated small images cropped from the SVG images, that are generated by MolMiner-ImgRec, to develop the MolMiner-TextOCR model. Some data augmentation tricks have been added in the recognition process.

**Model Construction**

**MolMiner-ImgDet Construction.** Our MolMiner-ImgDet model is an image segmentation model based on Deeplab v3+\(^2\) framework which uses MobileNet V2\(^3\) as network backbone. During the model development process, we referred to this implementation (https://github.com/bonlime/keras-deeplab-v3-plus). We first trained the model on our training datasets for 200 epochs by Adam optimizer with learning rate of 5e-3, and then fine-tuned it for 160 epochs by SGD optimizer with learning rate of 1e-4.\(^4\) More-
over, the size of input images is 1536x1536. To accelerate the model convergence, we used the pretrained weights of MobileNet V2 from this code repository (https://github.com/fchollet/deep-learning-models/releases/tag/v0.6).

**MolMiner-ImgRec Construction.** Our MolMiner-ImgRec model was developed using default YOLOv5m architecture. More specifically, we preformed the training process with default parameters (https://github.com/ultralytics/yolov5/blob/master/models/yolov5m.yaml) for 250 epochs on our training datasets (about 170 thousand images) divided by a ratio of 8:2 (training/validation). The pretrained weights (https://github.com/ultralytics/yolov5/releases) were used in our model.

**MolMiner-TextOCR Construction.** Our MolMiner-TextOCR model was developed based on the EasyOCR v1.4 framework. This framework mainly contains character text detection module and scene text recognition module. We used the mentioned dataset cropped from the MolMiner-ImgRec model to fine-tune an English-based pretrained model (https://github.com/JaidedAI/EasyOCR/releases/download/v1.3/english_g2.zip) for 200 epochs by Adam optimizer with a learning rate of 1e-4. This pretrained model was tested to reach a higher score and be suitable for noisy images.

**Postprocessing**

We used the algorithm of connected components (provided in OpenCV v4.5.5) to group the positive categorical outputs predicted by MolMiner-ImgDet. One component represents one chemical image with relative positions of the PDF page. These relative positions are used to crop the wanted regions from a whole PDF page. Subsequently, the cropped images are directly fed into the recognition model to produce the predicted relative positions of char and bond objects. The predicted positions and labels are then used to construct molecular graphs according to relative distances between objects.
Supplementary Figures

Figure S1: Case about chirality check. The original figure (left) has some errors on chiral atoms, shown in red circles. RDKit toolbox would check these atoms and then produce the molecule (right) where wedge bonds are masked.

![Chirality Check](image)

Figure S2: Case about double bonds with $trans(E)$, $cis(Z)$, either $E/Z$. In the Current MolMiner, we assign this type of bond to a simple double bond without stereo.

![Double Bonds](image)
Figure S3: Representative failure cases from MolVec github issue (https://github.com/ncats/molvec/issues/18). The headers of columns represent corresponding problems from MolVec recognition results. The recognized errors are marked by red circles.

References

(1) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M. P.; Mosquera, J. F.; Mutowka, P.; Nowotka, M., et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* 2019, 47, D930–D940.

(2) Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. Proceedings of the European conference on computer vision (ECCV). 2018; pp 801–818.

(3) Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 4510–4520.

(4) Keskar, N. S.; Socher, R. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628* 2017.

(5) Jocher, G. et al. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations. 2021,

(6) EasyOCR. https://github.com/JaidedAI/EasyOCR, Accessed: 2022-05-18.
(7) Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character region awareness for text detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; pp 9365–9374.

(8) Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence* 2016, 39, 2298–2304.

(9) OpenCV. https://github.com/opencv/opencv/releases/tag/4.5.5, Accessed: 2022-05-18.