# User Guide of Fluor-predictor

The initial interactive interface of Fluor-predictor is shown below, which mainly includes two functional modules: "Fluorescent Dye Retrieval" and "Fluorescent Dye Prediction." Users can simply select the corresponding module based on their needs. Fluor-predictor does not require registration or login, and there are no limitations on usage. It is a simple and efficient standalone software. Users can also directly use the provided GUI.py to add and modify software features. The interface upon entering is shown in Figure 1.

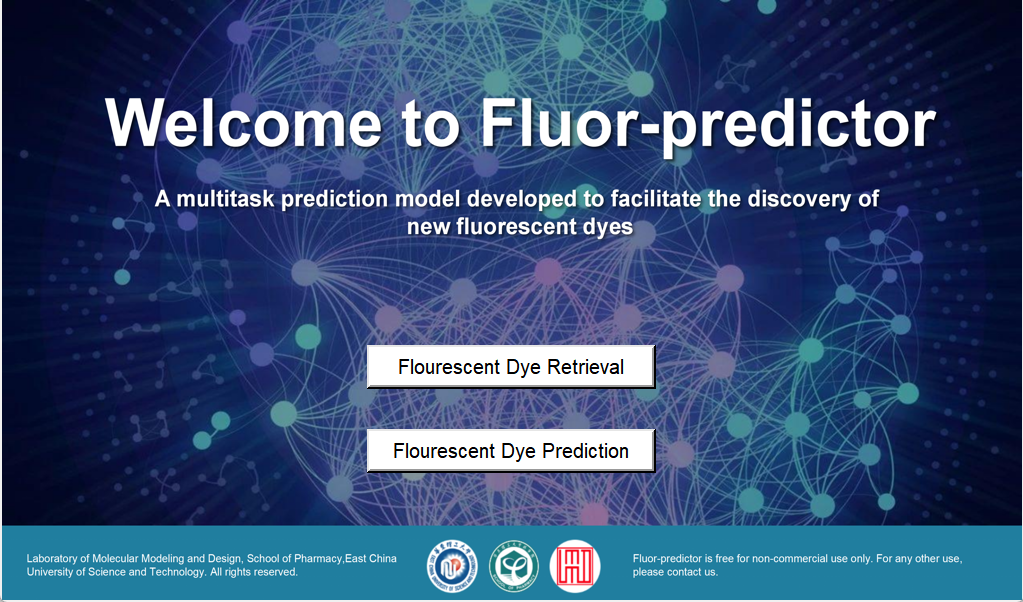


Figure 1. The initial interface of Fluor-predictor

# Function 1: Fluorescent Dye Retrieval

We utilize the data from the model construction as well as manually collected data on xanthene and cyanine dyes as our built-in database, providing direct retrieval, similarity search, and substructure search. Users simply need to input the SMILES of the dye molecule to be searched, select the search mode, and all search results will be saved in the Results folder. The differences between the three search methods are described in the following steps:

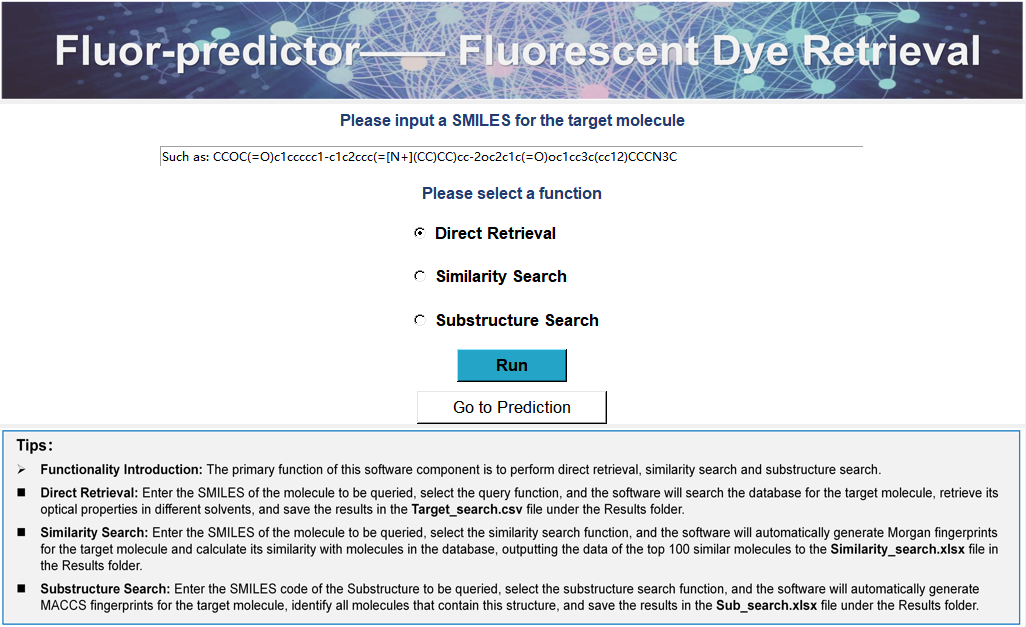


Figure 2. The interface of Function 1

* **Direct Retrieval:** Enter the SMILES of the molecule to be queried, select the query function, and the software will search the database for the target molecule, retrieve its optical properties in different solvents, and save the results in the **Target\_search.csv** file under the Results folder. Direct retrieval is suitable for querying known dye molecules, allowing for a quick search of the optical properties of the target dye in different solvents.
* **Similarity Search:** Enter the SMILES of the molecule to be queried, select the similarity search function, and the software will automatically generate Morgan fingerprints for the target molecule and calculate its similarity with molecules in the database, outputting the data of the top 100 similar molecules to the **Similarity\_search.xlsx** file in the Results folder. The number of recommendations given by the model can be defined in GUI.py. Similarity search is suitable for querying new dye molecules, allowing for a rough understanding of the properties of the target dye by knowing similar structures. For newer structures with potentially large differences in similarity search results compared to the target molecule, we recommend using the prediction module to directly predict the target dye.
* **Substructure Search:** Substructure Search: Enter the SMILES code of the substructure to be queried, select the substructure search function, and the software will automatically generate MACCS fingerprints for the target molecule, identify all molecules in the database that contain this structure, and save the results in the **Sub\_search.xlsx** file under the Results folder. Substructure Search is suitable for data searches of specific categories of dyes, allowing for the rapid collection of all dyes with related structures.

# Function 2: Fluorescent Dye Prediction

The prediction module integrates three forecasting models, respectively used for predicting all types of dyes, xanthene dyes, and cyanine dyes. To enhance the models' predictive ability for novel structures, all three models are derived from molecular-based partitioning. Users first need to select the corresponding prediction model based on the molecule to be predicted; we do not recommend using the xanthene and cyanine models to predict other categories of dyes.

* **Usage Steps:** First, please confirm the category of your dye molecule and make the corresponding selection in the category options. Then, enter the SMILES codes for the dye molecule and the solvent. Common solvent SMILES codes are provided below. Finally, click the prediction button. The prediction results will be saved in the **pred-results.csv** file under the Results directory.
* **Support for Multiple Inputs:** The model also supports the prediction of multiple dye molecules at once, simply separate the dye molecules with commas, and please ensure that the number of dyes corresponds to the number of solvents.

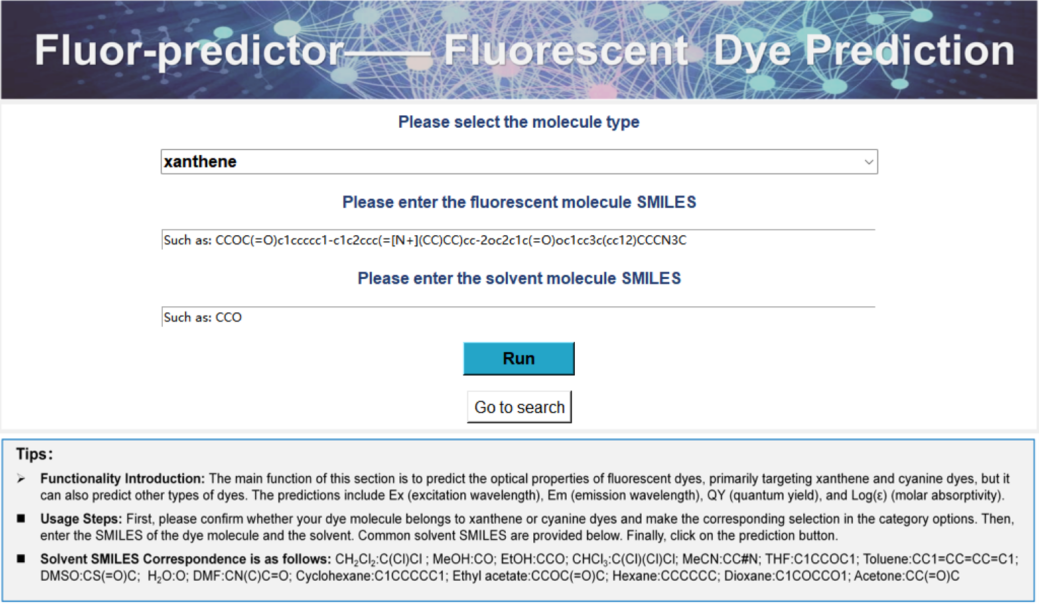


Figure 3. The interface of Function 2

# Creation of the Software Usage Environment：

To address the issues of software lag and large size after direct packaging, we recommend that users run the software in their own virtual environments. Below are the steps to create the environment for using Fluor-predictor; please follow the steps for installation:

1. **To Create the Environment:**

conda create -n dye37 python=3.7

1. **Library Installation:**

* conda install pytorch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1 pytorch-cuda=11.7 -c pytorch -c nvidia
* pip install pandas==1.3.0
* pip install dgllife==0.2.8
* pip install rdkit-pypi
* pip install dgl==1.1.2+cu117 -f https://data.dgl.ai/wheels/cu117/repo.html
* pip install packaging
* conda install xlsxwriter
* conda install -c anaconda scikit-learn

1. **Correction of Source Code Errors:**

There is an error in the **AttentiveFP** library that needs to be corrected. Here are the steps for the modification: In GUI.py, jump to the code in attentivefp.py through AttentiveFPGNN, rename copy\_edge to copy\_e; rename src\_mul\_edge to u\_mul\_e. After the installation is complete, you can run it directly.