DeepMol: a python-based machine and deep learning framework for drug discovery

João Correia[1,2], João Capela[1,2], Vítor Pereira[1,2], Miguel Rocha[1,2]

[1] BIOSYSTEMS, Centre of Biological Engineering University of Minho, Campus de Gualtar 4710 057 Braga, Portugal

[2] LABBELS –Associate Laboratory, Braga, Guimarães, Portugal

Introduction

Drug discovery is a complex and challenging process that involves the identification of small molecules with therapeutic potential. To expedite this process, computational frameworks like DeepMol leverage machine learning (ML) and deep learning (DL) algorithms to develop accurate predictive models for molecular properties. In this poster, we introduce DeepMol, a comprehensive and user-friendly framework that enables researchers to efficiently analyze large volumes of molecular data and generate predictive models with high performance. DeepMol provides a wide range of features, including preprocessing of molecular data, generation of features, model construction, and hyperparameter optimization. It also includes other techniques such as dimensionality reduction and feature explainability to help researchers gain insights into the underlying molecular mechanisms. With its user-friendly interface and powerful capabilities, DeepMol has the potential to significantly accelerate drug discovery efforts. In this poster, we will provide an overview of DeepMol's features and showcase some of its potential applications.

Methods

DeepMol employs a variety of preprocessing and ML techniques to develop predictive models for molecular properties (Fig. 1). The framework includes the following key steps in the ML pipeline:

- **Data loading**: The package can read data from CSV and SDF files or directly from numpy arrays.
- **Molecule standardization**: DeepMol offers a customizable set of steps to prepare molecules for analysis, such as sanitization, removal of isotope information, salt and fragment removal, and neutralization.

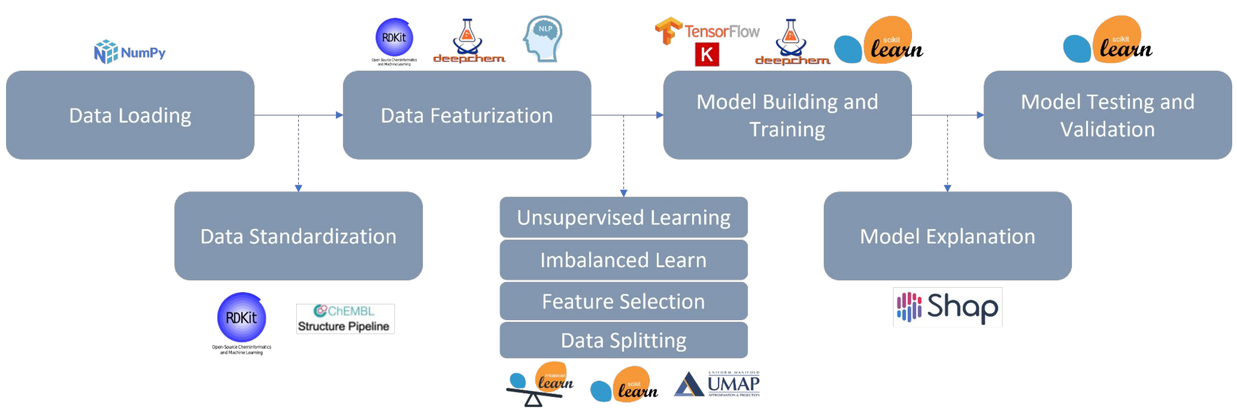

Fig. 1 - Overall DeepMol architecture. Each step includes the main packages it uses.

- **Feature generation**: The framework can compute a wide range of molecular features, including fingerprints, NLP-based embeddings, graph-based features, and 2D/3D descriptors.
- **Data scaling**: DeepMol can scale the features to improve model performance.
- **Unsupervised learning**: The framework supports techniques such as PCA, K-Means, t-SNE, and UMAP to extract insights from the data.
- **Imbalanced learning**: DeepMol can perform over and undersampling to balance classes in the data.
- **Feature selection**: The package offers methods for selecting the most relevant features for model building.
- **Data splitting**: DeepMol can split datasets based on molecular similarities and scaffolds to reduce overfitting.
- **Model building and training**: The framework includes various ML models, such as deep neural networks, and can perform hyperparameter optimization to improve model performance.
- **Model explanation**: DeepMol uses SHAP values to explain the contribution of each feature to the model predictions, providing valuable insights to understand the relationship between molecular structure and activity.
- **Model testing and validation**: Finally, the framework can evaluate the performance of the models using various metrics and techniques, such as cross-validation and ROC analysis.

Usage

How to use DeepMol?

Supervised:

```python
from sklearn.metrics import roc_auc_score
from deepmol.metrics import Metric
from deepmol.splitters import RandomSplitter
from sklearn.ensemble import RandomForestClassifier
from deepmol.models import SklearnModel
from deepmol.feature_selection import LowVarianceFS
from deepmol.standardizer import ChEMBLStandardizer
from deepmol.compound_featurization import MorganFingerprint
from deepmol.loaders import CSVLoader

#Load data
data = CSVLoader(dataset_path='data_path...',
                 smiles_field='Smiles',
                 labels_fields=['Class']).create_dataset()
#Standardize molecules
ChEMBLStandardizer().standardize(data)
#Compute Morgan fingerprints
MorganFingerprint(radius=2, size=1024).featurize(data)
#Remove features with low variance
LowVarianceFS(threshold=0.15).select_features(data)
# Split data into train and test (80/20)
train, test = RandomSplitter().train_test_split(data, frac_train=0.8)
#Random Forest from ScikitLearn
model = SklearnModel(model=RandomForestClassifier())
#Train the random forest model
model.fit(train)
#Evaluate model on test data using ROC AUC from scikit-learn
roc_auc = Metric(roc_auc_score)
model.evaluate(test, metrics=[roc_auc])
```

Fig. 2 - Data loading, molecule standardization, feature extraction, feature selection, data splitting and model training and evaluation code example.

Unsupervised Exploration (UMAP):

```python
from deepmol.unsupervised import UMAP
from deepmol.compound_featurization import LayeredFingerprint
from deepmol.standardizer import BasicStandardizer
from deepmol.loaders import CSVLoader

#Load data
data = CSVLoader(dataset_path='data_path...',
                 smiles_field='Smiles',
                 labels_fields=['Class']).create_dataset()
#Standardize molecules
BasicStandardizer().standardize(data)
#Compute Morgan fingerprints
LayeredFingerprint(fpSize=1024).featurize(data)
#Perform unsupervised dimensionality reduction with UMAP
umap = UMAP()
umap_df = umap.run_unsupervised(data)
umap.plot(umap_df.X, path='umap_output.png')
```

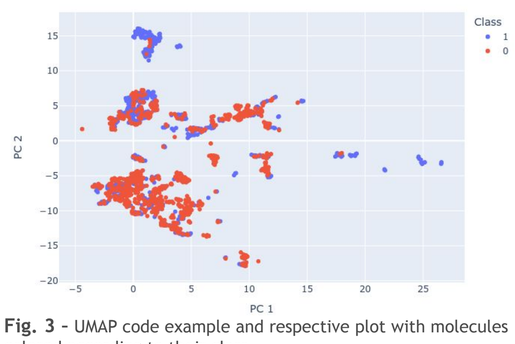

Fig. 3 - UMAP code example and respective plot with molecules colored according to their class.

Model Explainability:

```python
from deepmol.compound_featurization import MACCSkeysFingerprint
from deepmol.models import SklearnModel
from sklearn.ensemble import RandomForestClassifier
from deepmol.loaders import CSVLoader
from deepmol.feature_importance import ShapValues

#Load data
data = CSVLoader(dataset_path='data_path...',
                 smiles_field='Smiles',
                 labels_fields=['Class']).create_dataset()
#Featurize data
MACCSkeysFingerprint().featurize(data)
#Scikit-Learn Random Forest
model = SklearnModel(RandomForestClassifier())
model.fit(data)
#Compute SHAP values
shap_calc = ShapValues(data, model)
shap_calc.computePermutationShap()
```

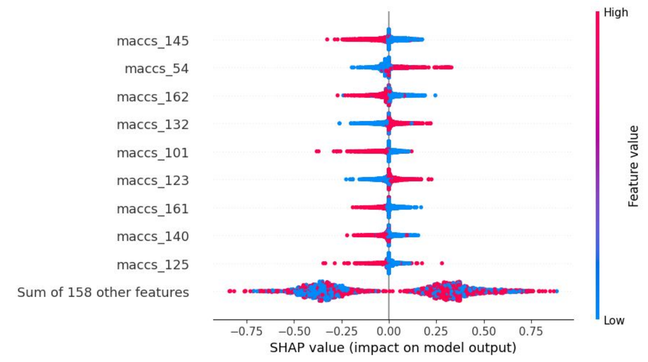

Fig. 4 - Code for feature importance on the left. The plot on the right shows the direction of the relationship between a feature and the model prediction. Positive SHAP-values are indicative of positive predictions (1), while negative SHAP-values are indicative of negative predictions (0).

Case Studies:

Identification of new sweeteners: In [1], DeepMol was used to train ML/DL models to predict if a molecule can be a potential sweetener. The results of the ML/DL pipelines are shown in Table 1. A SHAP values analysis was also conducted to infer the impact of the features (ECFP4 bits) on the model's decisions. As highlighted in Figure 3, bits associated with sweetness have a positive impact (Sweet prediction) on the model's prediction.

Table 1 - ML/DL models' performance

Descriptor-FS Method -Algorithm	Test ROC AUC	Test Precision	Test Recall
2D-SelectFromModel-RF	**0.929**	0.925	**0.933**
RDK-DNN	0.928	0.947	0.906
2D-Kbest-DNN	0.928	0.941	0.912
GCN	0.925	0.946	0.901
ECFP4-SVM	0.925	0.937	0.911
AtomPairFP-SelectFromModel-DNN	0.925	0.945	0.902
ECFP8-SVM	0.920	0.930	0.908
GraphConv	0.920	0.931	0.906
TextCNN	0.920	0.915	0.925
GAT	0.914	**0.954**	0.870
BiLSTM	0.912	0.944	0.884
LSTM	0.729	0.698	0.918

TextCNN: textual Convolutional Network; **GCN**: Graph Convolutional Network; **GAT**: Graph Attention Network; **GraphConv**: Duvenaud GCN; **RF**: Random Forest; **SVM**: Support Vector Machine; **DNN**: Deep Neural Network; **LSTM**: long short-term memory.

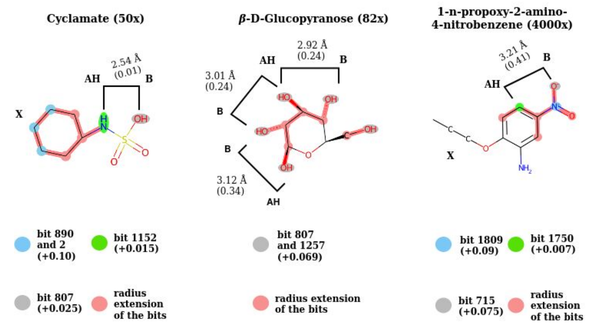

Fig. 5 - Bits associated with sweetness. The SHAP values of these bits for the ECFP4-SVM are presented between parenthesis.

Drug response prediction: In [2] and [3], DeepMol was used to predict drug sensitivity in cancer cell lines using 12 different molecular representations benchmarked on 5 compound screening datasets. The results indicate that end-to-end DL models perform similarly to, and sometimes better than, models trained on molecular fingerprints, even when less training data is available. Additionally, combining multiple compound representation methods in an ensemble can improve the model's performance (Tables 2 and 3).

Table 2 - Ensemble results for classification tasks.

Task	Model	ROC-AUC	Accuracy	PRC-AUC	Precision	Recall
NCI 1	ECFP4	0.831	0.831	0.870	0.826	0.831
	11-Model ensemble	0.862	0.862	0.898	0.881	0.831
	5-Model ensemble	0.851	0.852	0.888	0.866	0.825
NCI 109	GCN	0.824	0.824	0.864	0.809	0.842
	11-Model ensemble	0.856	0.856	0.894	0.878	0.822
	5-Model ensemble	0.845	0.845	0.882	0.847	0.838

Table 3 - Ensemble results for regression tasks.

Task	Model	RMSE	Pearson	R^2	Spearman
PC-3	TextCNN	0.607	0.806	0.641	0.773
	11-Model ensemble	0.581	0.831	0.671	0.797
	5-Model ensemble	0.555	0.837	0.700	0.811
CCRF-CEM	AtomPair	0.742	0.798	0.629	0.761
	11-Model ensemble	0.706	0.840	0.664	0.817
	5-Model ensemble	0.645	0.849	0.720	0.825
A549/ATCC	TextCNN	0.787	0.667	0.421	0.593
	11-Model ensemble	0.741	0.715	0.487	0.641
	5-Model ensemble	0.736	0.711	0.494	0.637

Conclusions

- DeepMol is a comprehensive and user-friendly framework that provides powerful ML and DL techniques for drug discovery research;
- The framework's ability to preprocess, analyze, and model large volumes of molecular data enables researchers to generate accurate predictive models with high performance;
- DeepMol has been successfully applied in three peer-reviewed studies, demonstrating its value in generating accurate predictive models and accelerating drug discovery efforts.

Code Availability

DeepMol

References

[1] Capela, J. *et al.*, "Development of Deep Learning approaches to predict relationships between chemical structures and sweetness," IJCNN, Padua, Italy, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.2022.9891992.

[2] Baptista, D. *et al.*, "Evaluating molecular representations in machine learning models for drug response prediction and interpretability" Journal of Integrative Bioinformatics, vol. 19, no. 3, 2022, pp. 20220006. https://doi.org/10.1515/jib-2022-0006

[3] Baptista, D. *et al.*, "A Comparison of Different Compound Representations for Drug Sensitivity Prediction". PACBB 2021. Lecture Notes in Networks and Systems, vol 325. Springer, Cham. https://doi.org/10.1007/978-3-030-86258-9_15

Acknowledgements: This research has been supported by the Portuguese Foundation for Science and Technology (FCT) through the DeepBio project - ref. NORTE-01-0247-FEDER- 039831, funded by Lisboa 2020, Norte 2020, Portugal 2020 and FEDER - Fundo Europeu de Desenvolvimento Regional. We also thank FCT for the PhD fellowships to J. Capela (DFA/BD/08789/2021) and J. Correia (SFRH/BD/144314/2019).