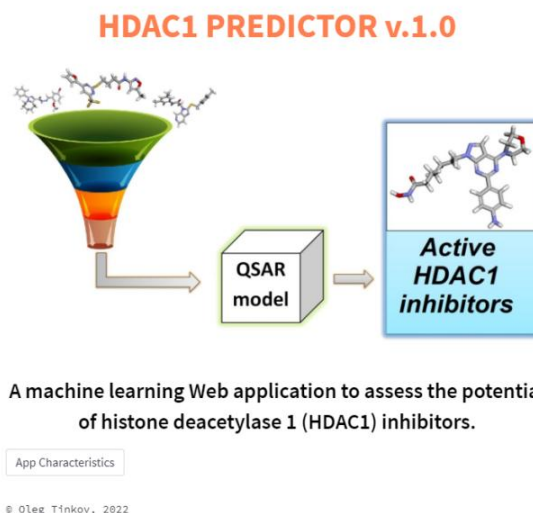


## Overview

The HDAC1 Predictor application provides an alternative method for assessing the potential of chemicals to be Histone deacetylase 1 (HDAC1) inhibitors. Compound is classified as active if the predicted  $IC_{50}$  value is lower than mean  $IC_{50}$  value of the reference drug Vorinostat (11.08 nM) otherwise compound is labeled as inactive. This application makes predictions based on Quantitative Structure-Activity Relationship (QSAR) models build on curated datasets generated from scientific articles. The consensus models were developed using open-source chemical descriptors based on ECFP4-like Morgan fingerprints and 2D RDKit descriptors, along with the random forest (RF), gradient boosting (GBM), support vector machines (SVM) algorithms, using Python 3.7. The models were generated applying the best practices for QSAR model development and validation widely accepted by the community. Batch processing is available through <https://github.com/ovttiras/HDAC1-inhibitors>. For more information, please refer to our paper:


**Step 1.** Select descriptors for consensus QSAR model. You can choose one of two types of molecular descriptors that were used in the development of the QSAR model. If you choose ECFP4 descriptors, an additional analysis of the contributions of molecular fragments of the studied compound to the inhibition of HDAC1 will be carried out.



**Step 2.** Select input molecular files.

If you choose smiles, please, directly paste the SMILES representation of the desired chemical structure and press Ctrl+Enter. If the entered chemical structure is correct, the application will generate a 2D image of the studied compound. Click on the “PREDICT COMPOUND FROM SMILES” button.

## HDAC1 PREDICTOR v.1.0



The diagram illustrates the workflow of the HDAC1 Predictor v.1.0. It starts with a funnel representing the selection or filtering process, where various chemical structures are input. The output of the funnel is a single chemical structure, which is then processed by a QSAR model. The final output is a box labeled "Active HDAC1 inhibitors", indicating the predicted active compounds.

App Characteristics

© Oleg Tinkov. 2022

### App Characteristics

rdkit.Chem.rdchem.AtomValenceException: This app has encountered an error. The original error message is redacted to prevent data leaks. Full error details have been recorded in the logs (if you're on Streamlit Cloud, click on 'Manage app' in the lower right of your app).

Traceback

```
File ~/home/appuser/venv/lib/python3.7/site-packages/streamlit/scrptrunner/
exec(code, module, _dict...)
File ~/app/hdcl-inhibitors/HDCL_predictor_app.py, line 14, in module:
smiles=standardize_smiles(compound_smiles)
File ~/home/appuser/venv/lib/python3.7/site-packages/molvs/standardize.py, l
mol = Standardizer().standardize(smiles)
File ~/home/appuser/venv/lib/python3.7/site-packages/molvs/standardize.py, l
Chem.SanitizeMol(mol)
```

Share 

[← Manage app](#)



A machine learning Web application to assess the potential of histone deacetylase 1 (HDAC1) inhibitors.

#### Key Characteristics

rdkit.Chem.rdchem.AtomValenceException: This app has encountered an error. The original error message is redacted to prevent data leaks. Full error details have been recorded in the logs (if you're on Streamlit Cloud, click on "Manage app" in the lower right of your app).

### Tracebacks

```
File ~/home/appuser/venv/lib/python3.7/site-packages/streamlit/scriptrunner.py:
exec(code, module.__dict__)

File ~/app/hack2-inhibitors/HG41_predictor/app.py, line 74, in module:
smiles=standardize_smiles(compound_smiles)

File ~/home/appuser/venv/lib/python3.7/site-packages/molvs/standardize.py, l:
mol = Standardizer().standardize(mol)

File ~/home/appuser/venv/lib/python3.7/site-packages/molvs/standardize.py, l:
chem.SanitizeMol(mol)
```

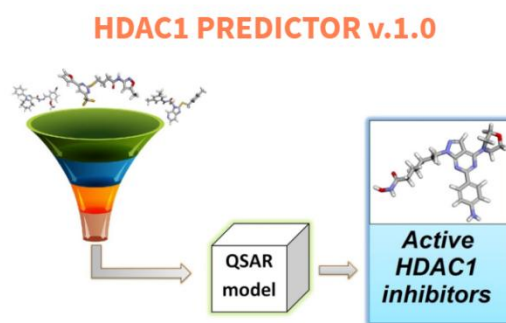
2000

```

[WARNING] apt dependencies were installed from /app/ibmc-inhibitors/packages.txt using apt-get.
[WARNING] Processed dependencies:
[WARNING] Updated apt
[WARNING] Pulling new changes from Github...
[WARNING] Processing dependencies...
[WARNING] apt dependencies were installed from /app/ibmc-inhibitors/packages.txt using apt-get.
[WARNING] Processed dependencies:
[WARNING] Updated apt
[WARNING] Pulling new changes from Github...
[WARNING] Processing dependencies...
[WARNING] apt dependencies were installed from /app/ibmc-inhibitors/packages.txt using apt-get.
[WARNING] Processed dependencies:
[WARNING] Updated apt
[WARNING] Pulling new changes from Github...
2022-06-06 09:45:03.043 ubuntu apt exception
Traceback (most recent call last):
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/atsmcli/scripts/prower/script_runner.py", line 443, in run_
    exec(code, module_dict, {})
  File "/app/ibmc-inhibitors/ibmc_inhibitor_genetic_app.py", line 67, in <module>
    smcli=atsmcli_wrapper(smcli)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 306, in standardize_smcli
    ml = Standardizer().standardize(ml)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 95, in standardize
    Chew(SmcliToken)
  File "/home/chen.chen/atsmcli/exception.py: Cannot tokenize ml. Uninitialized atoms: 0 1 2 4
2022-06-06 09:48:05.792 ubuntu apt exception
Traceback (most recent call last):
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/atsmcli/scripts/prower/script_runner.py", line 443, in run_
    exec(code, module_dict, {})
  File "/app/ibmc-inhibitors/ibmc_inhibitor_genetic_app.py", line 396, in <module>
    smcli=atsmcli_wrapper(smcli)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 306, in standardize_smcli
    ml = Standardizer().standardize(ml)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 95, in standardize
    Chew(SmcliToken)
  File "/home/chen.chen/atsmcli/exception.py: Cannot tokenize ml. No ring atom 9 marked aromatic
2022-06-06 09:48:05.792 ubuntu apt exception
Traceback (most recent call last):
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/atsmcli/scripts/prower/script_runner.py", line 443, in run_
    exec(code, module_dict, {})
  File "/app/ibmc-inhibitors/ibmc_inhibitor_genetic_app.py", line 74, in <module>
    smcli=atsmcli_wrapper(smcli)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 306, in standardize_smcli
    ml = Standardizer().standardize(ml)
  File "/home/appuser/.vscode/ibm/ibmcloud/7.9.1/packages/mlvs/standardize.py", line 95, in standardize
    Chew(SmcliToken)
  File "/home/chen.chen/atsmcli/exception.py: Explicit valence for atom # 5 C 5, is greater than permitted

```

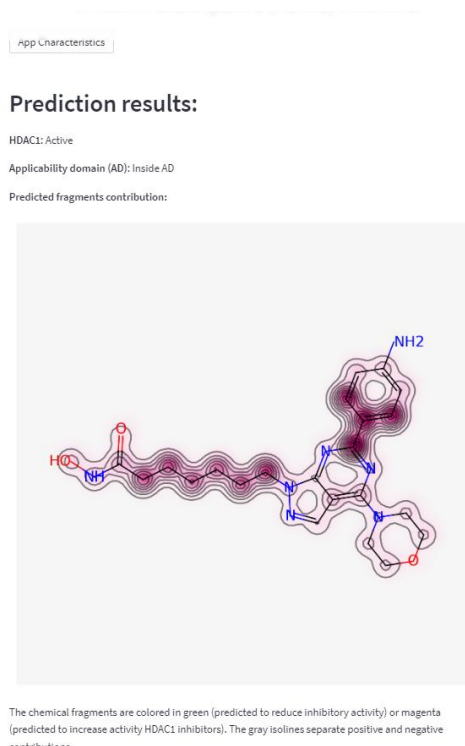
If you choose a file \*.sdf, that may contain a different number of chemical structures, please specify the path to this file on your computer's hard drive. In this case, you need to click the " Browse files" button.



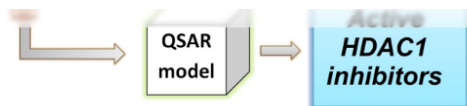
A machine learning Web application to assess the potential of histone deacetylase 1 (HDAC1) inhibitors.

[App Characteristics](#)

**Step 3.** Prediction results. The form of presentation of the results depends on the type of descriptors selected, as well as the format of the input chemical data. For example, when selecting SMILES, the results will be displayed for a single molecule.



If you select a file \*.sdf, the results will be presented in tabular form. If incorrect structures are detected in the file \*.sdf, the corresponding information will appear in the section "CHEMICAL STRUCTURE VALIDATION AND STANDARDIZATION"



A machine learning Web application to assess the potential of histone deacetylase 1 (HDAC1) inhibitors.

App Characteristics

## 1. CHEMICAL STRUCTURE VALIDATION AND STANDARDIZATION:

Original data: 211 molecules

Failed data: 9 molecules

|   | No. failed molecule in ori... | SMILES of wrong structure:             |
|---|-------------------------------|--|
| 1 | 80                            | <chem>COC1=C/C2=C(C(OC3=CC=C...</chem> |
| 2 | 88                            | <chem>O=C/C=C/C1=CC=C(C(CNC...</chem>  |
| 3 | 90                            | <chem>COC1=C(C(OC)=C2(C(=O)...</chem>  |
| 4 | 92                            | <chem>COC1=C/C2=C(C(OC3=CC=C...</chem> |
| 5 | 93                            | <chem>CCC1=CN(C2=CC=CC=C2)...</chem>   |
| 6 | 153                           | <chem>O=C(NO)C1=CC=C(CN2CC...</chem>   |
| 7 | 158                           | <chem>O=C(CC1=CC=C(CN2CCN(...</chem>   |
| 8 | 193                           | <chem>O=C(/C=C/C1=CC=C(CN2C...</chem>  |
| 9 | 203                           | <chem>COC1=C(C(OC)=C2(C(=O)...</chem>  |

Kept data: 202 molecules

## 2. RESULTS OF PREDICTION:

Show results as table

Show results and map of fragments contribution for each molecule separately

The prediction results for correct chemical structures are displayed in a table that can be downloaded, or separately for each molecule if you click the " Show results and map of fragments contribution for each molecule separately" button.

Select descriptors for consensus QSAR model

ECFP4

Select input molecular files

☐ SMILES notations (\*.smi)
 ☒ MDL multiple SD file (\*.sdf)

Choose a file

Drag and drop file here

Limit 200MB per file

Browse files

211.sdf

0.5KB

|   |     |                                       |
|---|-----|---------------------------------------|
| 5 | 93  | <chem>CCC1=CN(C2=CC=CC=C2)...</chem>  |
| 6 | 153 | <chem>O=C(NO)C1=CC=C(CN2CC...</chem>  |
| 7 | 158 | <chem>O=C(CC1=CC=C(CN2CCN(...</chem>  |
| 8 | 193 | <chem>O=C(/C=C/C1=CC=C(CN2C...</chem> |
| 9 | 203 | <chem>COC1=C(C(OC)=C2(C(=O)...</chem> |

Kept data: 202 molecules

## 2. RESULTS OF PREDICTION:

Show results as table

|    | SMILES                                       | HDAC1 activity | Applicability domain (AD) |
|----|--|----------------|---------------------------|
| 1  | <chem>C[C@H]1CN(C(=O)Nc2ccc(-c...</chem>     | Active         | Inside AD                 |
| 2  | <chem>Nc1ccc(-c2nc(N3CCOCC3)c3cn...</chem>   | Active         | Inside AD                 |
| 3  | <chem>COc1ccc(-c2nc(N3CCOCC3)c3cn...</chem>  | Active         | Inside AD                 |
| 4  | <chem>CN(Cc1nc2c(N3CCOCC3)nc(-c...</chem>    | Active         | Inside AD                 |
| 5  | <chem>CN(Cc1nc2c(N3CCOCC3)nc(-c...</chem>    | Active         | Inside AD                 |
| 6  | <chem>COc1ccc(Br)cc1C(=N)N(C)=O)CC...</chem> | Active         | Inside AD                 |
| 7  | <chem>CN(Cc1nc2c(N3CCOCC3)nc(-c...</chem>    | Active         | Inside AD                 |
| 8  | <chem>CCC(=O)CCCCC[C@H]1N[C]=...</chem>      | Active         | Inside AD                 |
| 9  | <chem>O=C(CCCCCC1ccc(-c2ccc3ncn...</chem>    | Active         | Inside AD                 |
| 10 | <chem>CN(C1C1cccc(-c2nc(N3CCOCC3)...</chem>  | Active         | Inside AD                 |

Download results of prediction as CSV

Show results and map of fragments contribution for each molecule separately

© Oleg Tinkov, 2022

Share

Manage app